

BridgeVIEW™ und LabVIEW™

Referenzhandbuch zur Programmierung in G

Deutschsprechende Zweigstellen

National Instruments Germany GmbH	National Instruments Ges.m.b.H.	National Instruments Switzerland
Konrad-Celtis-Straße 79 D-81369 München	Plainbachstraße 12 A-5101 Salzburg-Bergheim	Sonnenbergstraße 53 CH-5408 Ennetbaden

Internet-Unterstützung

E-Mail: ni.germany@natinst.com
FTP Site: <ftp.natinst.com>
Web Adresse: www.natinst.com

Mailbox Unterstützung–Bulletin Board Support (BBS)

Deutschland: 089/71 25 61
USA: 512 794 5422

Telefon–Unterstützung

Deutschland:	Tel: 089/741 31 30	Fax: 089/714 60 35	
Österreich:	Tel: 0662/45 79 90 0	Fax: 0662/45 79 90 19	
Schweiz:	Tel: 056/200 51 51	Fax: 056/200 51 55	Tel: 022/980 05 11 (Genf)

Internationale Zweigstellen

Australien 03 9879 5166, Belgien 02 757 00 20, Brasilien 011 288 3336, Dänemark 45 76 26 00,
Finnland 09 725 725 11, Frankreich 01 48 14 24 24, Großbritannien 01635 523545, Holland 0348 433466,
Hongkong 2645 3186, Israel 03 6120092, Italien 02 413091, Japan 03 5472 2970,
Kanada (Ontario) 905 785 0085, Kanada (Québec) 514 694 8521, Korea 02 596 7456, Mexiko 5 520 2635,
Norwegen 32 84 84 00, Singapur 2265886, Spanien 91 640 0085, Schweden 08 730 49 70, Taiwan 02 377 1200

National Instruments - Firmenhauptsitz

11500 North Mopac Expressway Austin, Texas 78759 USA Tel: (+1) 512 794 0100

Wichtige Informationen

Garantie

Für die Datenträger, auf denen Sie die Software von National Instruments erhalten, wird für den Zeitraum von 90 Tagen nach dem Versand (nachweisbar durch Quittungen oder andere Dokumente) garantiert, daß sie keine Defekte in Material und Verarbeitung aufweisen, durch die die Ausführung der Programmanweisungen behindert wird. Sofern National Instruments während der Garantiezeit über bestehende Schäden informiert wird, entscheidet National Instruments nach eigenem Ermessen, ob Software-Datenträger, auf denen die Ausführung der Programmanweisungen nicht möglich ist, entweder repariert oder ersetzt werden. National Instruments leistet keine Gewähr dafür, daß die Ausführung der Software unbehindert und fehlerfrei erfolgen kann.

Es können keine Einsendungen zur Garantiebearbeitung entgegengenommen werden, die nicht deutlich auf der Außenseite durch eine Autorisierungsnummer für die Rücksendung, eine sogenannte RMA-Nummer (Return Material Authorization), gekennzeichnet sind. Lassen Sie sich vom Werk eine solche Autorisierungsnummer zuweisen. National Instruments übernimmt die Versandkosten für Teile, die im Rahmen einer Garantieleistung an den Kunden zurückgesendet werden.

National Instruments geht davon aus, daß die Informationen in diesem Handbuch korrekt sind. Die technischen Angaben in diesem Dokument wurden sorgfältig überprüft. Falls trotzdem technische oder typographische Fehler vorhanden sind, behält sich National Instruments das Recht vor, in nachfolgenden Auflagen dieses Dokuments Änderungen ohne weitere Mitteilung an die Benutzer dieser Auflage durchzuführen. Leser, die der Meinung sind, daß ein Fehler vorliegt, sollten sich direkt an National Instruments wenden. Unter keinen Umständen übernimmt National Instruments eine Haftung für Schäden, die eventuell aufgrund dieses Dokuments bzw. der darin enthaltenen Informationen oder im Zusammenhang damit entstehen.

NEBEN DER HIER BESCHRIEBENEN GARANTIE ÜBERNIMMT NATIONAL INSTRUMENTS WEDER AUSDRÜCKLICHE NOCH STILLSCHWEIGENDE GEWÄHRLEISTUNGEN. INSBESONDERE WIRD KEINE GARANTIE FÜR MARKTGÄNGIGKEIT ODER DIE EIGNUNG FÜR EINEN BESTIMMTEN ZWECK ÜBERNOMMEN. DIE SCHADENSERSATZANSPRÜCHE FÜR SCHÄDEN, DIE DURCH VERSCHULDEN ODER FAHRLÄSSIGKEIT VON NATIONAL INSTRUMENTS VERURSACHT WERDEN, SIND AUF DIE HÖHE DES KAUFPREISES BESCHRÄNKT, DEN DER KUNDE FÜR DAS PRODUKT BEZAHLT HAT. NATIONAL INSTRUMENTS IST NICHT HAFTBAR FÜR SCHÄDEN, DIE DURCH DEN VERLUST VON DATEN, GEWINNEINBUSSEN, DURCH DIE EINSCHRÄNKUNG DER VERWENDBARKEIT DER PRODUKTE ODER DURCH NEBEN- ODER FOLGESCHÄDEN ENTSTEHEN. DIES GILT AUCH DANN, WENN NATIONAL INSTRUMENTS ÜBER DIE MÖGLICHKEIT SOLCHER SCHÄDEN UNTERRICHTET WURDE. DIESE EINSCHRÄNKUNG DER HAFTUNG VON NATIONAL INSTRUMENTS GILT UNABHÄNGIG VON DER ART DER JEWEILIGEN HANDLUNG, SEI ES IM RAHMEN EINES VERTRAGS ODER ALS UNERLAUBTE HANDLUNG, UND GILT AUCH IN FÄLLEN VON FAHRLÄSSIGKEIT. GERICHTLICHE SCHRITTE GEGEN NATIONAL INSTRUMENTS MÜSSEN INNERHALB VON EINEM JAHR NACH EINTRETEN DES GRUNDES FÜR DIESE SCHRITTE EINGELEITET WERDEN. NATIONAL INSTRUMENTS IST NICHT HAFTBAR FÜR DIE VERZÖGERUNG BESTIMMTER LEISTUNGEN, DIE DURCH VORGÄNGE VERURSACHT WIRD, ÜBER DIE NATIONAL INSTRUMENT BEI EINER VERNÜNFITIGEN ABWÄGUNG KEINE KONTROLLE AUSÜBEN KANN. DIE HIERMIT GEBEBENE GARANTIE ERSTRECKT SICH NICHT AUF SCHÄDEN, DEFEKTE, FEHLFUNKTIONEN ODER FUNKTIONSAUSFÄLLE, DIE DADURCH VERURSACHT WERDEN, DAß DER BENUTZER DIE ANLEITUNGEN VON NATIONAL INSTRUMENTS FÜR DIE INSTALLATION, DEN BETRIEB UND DIE WARTUNG NICHT EINHÄLT. DIESER GARANTIEAUSSCHLUß GILT EBENSO FÜR SCHÄDEN, DIE DURCH MODIFIKATIONEN, DURCH MIßBRAUCH ODER FAHRLÄSSIGES VERHALTEN AUF SEITEN DES BENUTZERS, DURCH STROMAUSFÄLLE ODER SPANNUNGSSTÖßE, DURCH BRAND, ÜBERSCHWEMMUNGEN, UNFÄLLE, HANDLUNGEN EINER DRITTEN PARTEI ODER ANDERE VORFÄLLE VERURSACHT WERDEN, DIE NICHT IN EINEM VERNÜNFITIGEN RAHMEN KONTROLLIERT WERDEN KÖNNEN.

Copyright

Laut Urheberrechtsgesetz darf diese Veröffentlichung weder teilweise noch insgesamt in irgendeiner Form, sei es auf elektronischer oder mechanischer Weise einschließlich Fotokopieren, Aufzeichnen und Speichern in einem Informationsabrufsystem oder Übersetzen ohne die vorherige schriftliche Genehmigung von National Instruments Corporation reproduziert oder übertragen werden.

Warenzeichen

BridgeVIEW™, CVI™, LabVIEW™, natinst.com™, National Instruments™, und NI-DAQ™ sind Warenzeichen der National Instruments Corporation.

Die aufgeführten Produkt- und Firmennamen sind Warenzeichen oder Handelsnamen der jeweiligen Firmen.

WARNHINWEIS BEZÜGLICH DES EINSATZES VON PRODUKTEN VON NATIONAL INSTRUMENTS FÜR MEDIZINISCHE UND KLINISCHE ZWECKE

Die Produkte von National Instruments sind aufgrund der verwendeten Komponenten und der angewendeten Testverfahren nicht dazu geeignet, die notwendige Zuverlässigkeit für den Einsatz in der Behandlung und Diagnose von Patienten zu gewährleisten. Durch den Einsatz von National Instruments-Produkten in Anwendungen, die für medizinische oder klinische Zwecke gedacht sind, können mögliche Verletzungen durch das Versagen eines Produkts oder durch Fehler des Benutzers oder Anwendungsentwicklers entstehen. Alle Einsätze und Anwendungen von Produkten von National Instruments, die im Rahmen einer medizinischen oder klinischen Behandlung verwendet werden, dürfen nur durch ordnungsgemäß ausgebildetes und qualifiziertes medizinisches Personal erfolgen. Alle gebräuchlichen medizinischen Sicherheitsmaßnahmen, Geräte und Verfahren, die im jeweiligen Einzelfall zum Schutz vor ernst

Verletzungen mit möglicher Todesfolge angemessen sind, müssen ununterbrochen angewendet werden, solange Produkte von National Instruments eingesetzt werden. Die Produkte von National Instruments sind NICHT als Ersatz für etablierte Verfahren, Vorgehensweisen oder Geräte gedacht, die zur Überwachung oder zum Schutz der Gesundheit und zur allgemeinen Sicherheit von Patienten und Personal in der medizinischen oder klinischen Behandlung eingesetzt werden.

Inhaltsverzeichnis

Über dieses Handbuch

Gliederung dieses Handbuchs.....	xxiii
Teil I—Grundlegende Konzepte der Sprache G.....	xxiii
Teil II—Frontpanel-Objekte.....	xxiv
Teil III—Blockdiagramm-Programmierung.....	xxv
Teil IV—Fortgeschrittene G-Themen.....	xxvi
Anhänge, Glossar und Stichwortverzeichnis.....	xxvii
Schreibkonventionen in diesem Handbuch.....	xxvii
Verwandte Dokumentationsmaterialien.....	xxix
Mitteilungen von Kunden.....	xxix

Teil I—Grundlegende Konzepte der Sprache G

Kapitel 1

Einführung in die Programmierung in G

Was ist G?	1-1
Komponenten eines VIs.....	1-1
Frontpanel.....	1-3
Blockdiagramm.....	1-4
Icon und Anschluß.....	1-6
So erhalten Sie Hilfe.....	1-7
Hilfe zu einem Frontpanel.....	1-8
Hilfe zu einem Blockdiagramm.....	1-8
Hilfe zu Attributen.....	1-10
Online-Referenz.....	1-10

Kapitel 2

Bearbeiten von VIs

G-Umgebung	2-1
Werkzeugpalette	2-4
Verwendung von Menüs.....	2-6
Popup-Menüs.....	2-6
Bearbeiten von VIs.....	2-7
Auswählen von Objekten	2-7
Ziehen und Ablegen von VIs, Bildern und Text	2-9
Positionieren von Objekten	2-10

Neu anordnen	2-11
Ausrichten von Objekten	2-12
Einteilen von Objekten	2-13
Duplizieren von Objekten	2-14
Löschen von Objekten	2-15
Beschriften von Objekten	2-15
Beschriftungen von Frontpanels	2-15
Freie Labels	2-17
Texteigenschaften	2-19
Anpassen der Größe von Objekten	2-25
Labels	2-26
Arbeitsbereich hinzufügen	2-26
Farbige Kennzeichnung von Objekten	2-26
Rückgängig machen	2-28
Erstellen von Objektbeschreibungen	2-29
Erstellen von VI-Beschreibungen	2-30
Speichern von VIs	2-31
Individuelle VI-Dateien	2-31
VI-Bibliotheken (.LLBs)	2-33
Erstellen von VI-Bibliotheken	2-34
VIs in bestehenden VI-Bibliotheken speichern	2-35
Bibliotheksinhalte bearbeiten	2-35

Kapitel 3

Verwendung von SubVIs

Hierarchischer Aufbau	3-1
Erstellen von SubVIs aus VIs	3-2
Erstellen des Icons	3-2
Definieren von Anschlußterminal-Mustern	3-5
Auswählen und Ändern von Terminal-Mustern	3-6
Zuweisen von Bedien- und Anzeigeelementen zu Terminals	3-7
Erforderliche, empfohlene und optionale Verbindungen für SubVIs	3-9
Löschen von Terminalverbindungen	3-9
Bestätigen von Terminalverbindungen	3-10
Erstellen von SubVIs anhand einer VI-Auswahl	3-10
Regeln und Empfehlungen	3-11
Anzahl der Verbindungen	3-12
Zyklen	3-12
Attributknoten mit Schleifen	3-12
Unlogische Auswahlsätze	3-13
Lokale Variable und Frontpanel-Terminals innerhalb von Schleifen	3-13

Case-Strukturen mit Attributknoten, lokalen Variablen oder Frontpanel-Terminals	3-13
Verwendung des Hierarchiefensters	3-15
Öffnen des Hierarchiefensters	3-15
Optionen im Hierarchiefenster	3-17
Optionen im Menü Betrachten	3-17
Tasten der Hierarchie-Werkzeugpalette	3-18
Popup-Menü von Hierarchieknoten	3-19
Mausklicksequenzen bei Hierarchieknoten	3-21
Suchen von VIs im Hierarchiefenster	3-21
Suchen von VIs, Objekten und Text	3-23
Das Dialogfeld "Suchen"	3-23
Suchen von VIs und anderen Objekten	3-24
Suchen von Text	3-25
Optionen für Suchstrings	3-26
Textsuchoptionen	3-26
Eingrenzen des Suchbereichs	3-27
Das Fenster "Suchergebnisse"	3-28
Suchen des nächsten und vorherigen Suchobjekts	3-29
Popup-Menü "Suchen" für globale und lokale Variablen und Attributknoten ..	3-29

Kapitel 4

Ausführung und Debugging von VIs und SubVIs

Ausführung von VIs	4-1
Ausführen von VIs	4-1
Anhalten von VIs	4-4
Wiederholte Ausführung von VIs	4-4
Datenprotokollierung auf dem Frontpanel	4-5
Programmatische Datenabfrage	4-7
Zugreifen auf Datenbanken	4-7
Abrufen von Daten mit Datei-I/O-Funktionen	4-8
Debugging von VIs	4-9
Debugging von nicht ausführbaren VIs	4-9
Auslegen von Fehlermeldungen	4-11
Debugging von ausführbaren VIs	4-14
Warnungen	4-16
Erkennen undefinierter Daten	4-16
Korrigieren von VI-Bereichsfehlern	4-18
Debugging-Funktionen	4-19
Einzelschrittausführung der VIs	4-19
Ein Beispiel der Einzelschrittausführung eines VIs	4-19
Verwendung der Schritttasten	4-21

Lesen von Aufrufketten	4-21
Highlight-Funktion	4-22
Verwendung des Probe-Werkzeugs	4-24
Erstellen von Proben	4-26
Plazieren von Breakpoint-Werkzeugen	4-27
Unterdrücken der Ausführung	4-30
Erkennen automatischer Unterdrückung	4-30
Verwendung von Werkzeugpaletten-Tasten bei Unterdrückung von SubVIs	4-31
Ansicht der Hierarchiefenster während der Unterdrückung ..	4-31
Deaktivieren von Debugging-Funktionen	4-32
Ausschließen von Diagrammabschnitten	4-32

Kapitel 5

Drucken und Dokumentieren von VIs

Drucken	5-1
Druckkonfiguration	5-2
PostScript-Drucken	5-2
Drucken des aktiven Fensters	5-2
Dokumentieren von VIs	5-3
Drucken der Dokumentation	5-3
Einstellen von Ausdruckformaten	5-4
Einstellen anderer Druckoptionen	5-5
Erstellen von benutzerspezifischen Druckeinstellungen	5-5
Programmatisches Drucken	5-8
Steuern des Ausdruckzeitpunkts	5-8
Verbesserung der Ausdrücke	5-9
Einstellen des Seitenlayouts	5-9
Verwendung von alternativen Druckmethoden	5-10
Drucken/Exportieren von Bedienelement- und VI-Beschreibungen in eine RTF- oder HTML-Datei	5-10
Programmatische RTF- und HTML-Dateien	5-14
Lokalisierungsthemen	5-15
Erstellen Ihrer eigenen Hilfe-Dateien	5-15

Kapitel 6

Einrichten von VIs und SubVIs

Erstellen von Popup-Panels	6-1
VI-Einrichtungsoptionen	6-2
Ausführungsoptionen	6-2
Fensteroptionen	6-3
Dokumentationsoptionen	6-5
Das Dialogfeld "Einstellungen SubVI"	6-7
Anpassen der Menüleiste	6-8
Menü-Editor	6-9
Menüoptionen des Menü-Editors	6-12
Symboleisten-Optionen des Menü-Editors	6-13
Handhabung der Menüauswahl	6-14
Funktionen zur Handhabung der Menüauswahl	6-15
Menüauswahl bekommen	6-15
Aktivieren der Menüüberwachung	6-16
Dynamische Menüfunktionen	6-16
Menüeinträge einfügen	6-16
Menüeinträge löschen	6-18
Menüpunkt-Info bekommen	6-19
Set Menu Item Info	6-19
Menü-Kurzinformation bekommen	6-20
Tags von Menüpunkten der Anwendung	6-20

Kapitel 7

Anpassen Ihrer Umgebung

Festlegen von Voreinstellungen	7-1
Pfadvoreinstellungen	7-2
Bibliothekverzeichnisse, temporäre Verzeichnisse, Standardverzeichnisse und Menüverzeichnisse	7-3
VI-Suchpfad	7-4
Voreinstellungen für Leistung und Speichermedium	7-6
Voreinstellungen für Frontpanels	7-9
Voreinstellungen für Blockdiagramme	7-11
Debug-Voreinstellungen	7-13
Voreinstellungen für Farben	7-14
Voreinstellungen für Schriftarten	7-15
Druckervoreinstellungen	7-16
Historie-Voreinstellungen	7-18
Voreinstellungen für Uhrzeit und Datum	7-21
Sonstige Voreinstellungen	7-22
Server: Konfiguration	7-23

Server: TCP/IP-Zugriff	7-24
Server: Exportierte VIs	7-27
Wie werden Voreinstellungen gespeichert?	7-30
Rückgängig machen	7-32
Anpassen der Elemente- und Funktionenpaletten	7-33
Hinzufügen von VIs und Bedienelementen zu user.lib und instr.lib	7-34
Installieren und Ändern von Ansichten	7-34
Paletten-Editor	7-35
Erstellen von Unterpaletten	7-36
Verschieben von Unterpaletten	7-37
Funktionsweise von Ansichten	7-37

Teil II—Frontpanel-Objekte

Kapitel 8

Einführung in Frontpanel-Objekte

Erstellen des Frontpanels	8-1
Optionen für Bedien- und Anzeigeelemente von Frontpanels	8-2
Ersetzen von Bedienelementen	8-4
Optionen der Tastenbelegung für Bedienelemente	8-6
Die Option “Panelreihenfolge”	8-8
Anpassen von Steuerungen in Dialogfeldern	8-9
Anpassen von Bedienelementen durch den Import von Grafiken	8-10

Kapitel 9

Numerische Bedien- und Anzeigeelemente

Digitale Bedien- und Anzeigeelemente	9-2
Optionen der Digitalnumerik	9-3
Anzeigen von Integer-Werten in anderen Radizes	9-4
Ändern der Darstellung von numerischen Werten	9-4
Bereichsoptionen für numerische Bedien- und Anzeigeelemente	9-5
Numerische Bereichsprüfung	9-6
Format und Präzision von digitalen Anzeigen	9-8
Numerische Bedien- und Anzeigeelemente mit Schieberegler	9-12
Schiebereglerkala	9-14
Achsenmarker	9-15
Ändern der Skalagrenzen	9-15
Auswählen der nicht-uniformen Achsenmarker-Verteilung	9-16
Textskala	9-18

Ausgefüllte Schieberegler und Schieberegler mit Mehrfachwerten.....	9-20
Runde numerische Bedien- und Anzeigeelemente	9-22
Farbfeld	9-24
Farbrampe	9-25
Einheitentyp	9-26
Eingeben von Einheiten.....	9-30
Überprüfen von Einheiten und strikten Typen	9-31
Polymorphe Einheiten	9-33

Kapitel 10

Boolesche Bedien- und Anzeigeelemente

Erstellen und Bedienen von Booleschen Bedien- und Anzeigeelementen	10-1
Konfigurieren von Booleschen Bedien- und Anzeigeelementen.....	10-3
Ändern der Booleschen Labels	10-3
Boolesche Datenbereichsprüfung	10-4
Konfigurieren des Schaltverhaltens der Booleschen Bedienelemente	10-4
Anpassen von Booleschen Bedien- und Anzeigeelementen mit importierten Bildern	10-6

Kapitel 11

Bedien- und Anzeigeelemente vom Typ String

Verwendung von Bedien- und Anzeigeelementen vom Typ String	11-1
Menüpunkte für Bedien- und Anzeigeelemente vom Typ String.....	11-3
Rollbalken-Menüpunkte	11-3
Anzeigetypen.....	11-4
Normale Anzeige	11-4
Umgekehrter Schrägstrich- (‘\’) Code-Anzeige.....	11-4
Anzeige des Paßwortes	11-6
Hex-Anzeige	11-6
Auf eine Zeile begrenzen	11-6
Wert während der Eingabe aktualisieren.....	11-6
Tabellen	11-7
Ändern der Größe von Tabellen, Zeilen und Spalten.....	11-7
Eingeben und Auswählen von Datentabellen.....	11-8

Kapitel 12

Bedien- und Anzeigeelemente vom Typ Pfad und Refnum

Bedien- und Anzeigeelemente vom Typ Pfad	12-1
Bedien- und Anzeigeelemente vom Typ Refnum.....	12-3

Kapitel 13

Bedien- und Anzeigeelemente vom Typ Liste und Ring

Bedienelemente vom Typ Listenfeld	13-2
Erstellen einer Liste mit Objekten	13-2
Auswählen von Listenfeld-Objekten	13-3
Listenfeld-Datentypen	13-3
Popup-Menüobjekte des Listenfeldes	13-4
Anzeigen	13-4
Auswahlmodus	13-5
Tastatur-Modus	13-6
Objekt deaktivieren	13-7
Objektsymbol und Trennlinie	13-7
Bedienelemente vom Typ Ring	13-8
Hinzufügen von Textobjekten zu Ringen	13-9
Hinzufügen von Bildobjekten zu Ringen	13-11
Ändern von Größe und Text von Text- & Bild-Ringen	13-11
Enumerated Type-Bedienelemente	13-12

Kapitel 14

Bedien- und Anzeigeelemente vom Typ Array und Cluster

Arrays	14-2
Erstellen von Array-Bedienelementen	14-5
Array-Dimensionen	14-8
Array-Indexanzeige	14-9
Darstellung von Arrays in Einzelelement- oder Tabellenform	14-10
Ausführen von Arrays	14-11
Standardgrößen und -werte von Arrays	14-12
Arrayelemente	14-14
Feststellen der Größe von Arrays	14-14
Verschieben oder Ändern der Größe von Arrays	14-14
Auswählen von Array-Zellen	14-15
Ein Beispiel für die Auswahl von Arrayzellen	14-15
G-Arrays und Arrays in anderen Systemen	14-17
Cluster	14-20
Erstellen von Clustern	14-21
Ausführen und Konfigurieren von Cluster-Elementen	14-22
Cluster-Elemente	14-22
Standardwerte von Clustern	14-23
Reihenfolge von Cluster-Elementen	14-23
Verschieben und Ändern der Größe von Clustern	14-24
Zusammensetzen von Clustern	14-25
Die Funktion "Bündeln"	14-25

Die Funktion "Nach Namen bündeln"	14-26
Die Funktion "Array in Cluster"	14-29
Zerlegen von Clustern	14-30
Die Funktion "Aufschlüsseln"	14-30
Die Funktion "Nach Namen aufschlüsseln"	14-31
Die Funktion "Cluster in Array"	14-33
Ersetzen von Cluster-Elementen	14-34

Kapitel 15

Bedien- und Anzeigeelemente vom Typ Graph und Diagramm

Erstellen von Kurven- und xy-Graphen	15-2
Plotten von Einzelplot-Graphen	15-3
Datentypen von Kurvengraphen	15-3
Datentypen für xy-Graphen	15-4
Plotten von Mehrplot-Graphen	15-5
Datentypen für Kurvengraphen	15-5
Datentypen für xy-Graphen	15-10
Einstellen benutzerspezifischer Optionen bei einem Graphen	15-11
Maßstaboptionen	15-12
Markierungsabstand	15-13
Formatieren	15-14
Automatische Skalierung	15-17
Angepaßt	15-17
Scroll- und Zoomoptionen	15-17
Legendenoptionen	15-19
Kurvendiagramme	15-21
Datentypen für Kurvendiagramme	15-22
Optionen für Kurvendiagramme	15-24
Aktualisierungsmodi von Diagrammen	15-25
Gestapelte und überlagerte Plots	15-27
Cursor des Graphen	15-28
Intensitätsdiagramme	15-33
Optionen für Intensitätsdiagramme	15-35
Farbzuordnung	15-37
Intensitätsgraphen	15-38
Datentypen für Intensitätsgraphen	15-38
Optionen von Intensitätsgraphen	15-38

Kapitel 16

ActiveX-Bedienelemente

ActiveX-Erweiterungen des Frontpanels	16-1
Bedien- und Anzeigeelement vom Typ “ActiveX-Datentyp”	16-2
ActiveX-Container	16-2
Erstellen von ActiveX-Paletten	16-5

Teil III—Blockdiagramm-Programmierung

Kapitel 17

Einführung in das Blockdiagramm

Terminals und Knoten	17-1
Terminals	17-2
Terminals für Bedien- und Anzeigeelemente	17-2
Konstanten	17-4
Benutzerdefinierte Konstanten	17-4
Universelle Konstanten	17-8
Knoten	17-8
Funktionen	17-9
Strukturen	17-11
Ersetzen und Einfügen von Objekten des Blockdiagramms	17-12
Automatisches Hinzufügen von Konstanten und Bedien- und Anzeigeelementen	17-13

Kapitel 18

Verbinden des Blockdiagramms

Verbindungstechniken	18-1
Dehnen von Verbindungen	18-5
Auswählen, Verschieben und Löschen von Verbindungen	18-6
Offscreen-Verbindungen	18-8
Beheben von Verbindungsproblemen	18-9
Ungültige Verbindungen	18-9
Typenkonflikte von Verbindungen	18-10
Mehrere Quellen von Verbindungen	18-11
Keine Verbindungsquelle	18-11
Offene Verbindungen	18-12
Verbindungen mit Endlosschleifen	18-13
Zu vermeidende Verbindungssituationen	18-13
Schleifen von Verbindungen	18-14

Verborgene Verbindungssegmente	18-14
Erzeugen von Verbindungen unterhalb von Objekten	18-15

Kapitel 19 Strukturen

Strukturen vom Typ For-Schleife und While-Schleife	19-3
For-Schleife	19-3
While-Schleife	19-4
Plazieren von Objekten in Strukturen	19-4
Plazierung und Größenanpassung von Strukturen im Blockdiagramm	19-5
Plazieren von Terminals in Schleifen	19-6
Auto-Indizierung	19-7
Auto-Indizierung zur Einstellung des For-Schleifenzählers	19-8
Auto-Indizierung bei While-Schleifen	19-9
Nullmalige Ausführung einer For-Schleife	19-9
Schieberegister	19-10
Case-Strukturen und Sequenzstrukturen	19-13
Case-Strukturen	19-14
Sequenzstrukturen	19-18
Bearbeiten von Case-Strukturen und Sequenzstrukturen	19-20
Verschiebungen zwischen Subdiagrammen	19-21
Hinzufügen von Subdiagrammen	19-21
Löschen von Subdiagrammen	19-22
Probleme bei Verbindungen von Strukturen	19-22
Zuweisen mehrerer Werte zu einer lokalen Sequenz-Variable	19-22
Keine Verbindung zu einem Tunnel in allen Cases einer Case-Struktur	19-23
Überlappende Tunnel	19-23
Verbindungen von mehreren Rahmen einer Sequenzstruktur	19-25
Verbindungen unterhalb einer Struktur statt durch eine Struktur	19-25
Entfernen von Strukturen, ohne Elemente in einer Struktur zu löschen	19-26

Kapitel 20 Formelknoten

Verwendung von Formelknoten	20-1
Funktionen und Operatoren des Formelknotens	20-5
Syntax von Formelknoten	20-8
Fehler in Formelknoten	20-10

Kapitel 21 VI-Server

Verwendung des VI-Servers.....	21-1
Fähigkeiten des VI-Servers	21-3
Anwendungs- und VI-Referenzen	21-4
Verwendung des Eigenschaftsknotens und Methodenknotens bei Anwendungs- und VI-Referenzen	21-4
Beispiele für Eigenschaften und Methoden der VI-Klasse und Anwendungsklasse	21-8
Manipulation von Eigenschaften und Methoden der VI-Klasse	21-8
Manipulation von Eigenschaften und Methoden der Anwendungsklasse	21-9
Manipulation von Eigenschaften und Methoden der VI-Klasse und Anwendungsklasse	21-10
Streng typisierte VI-Refnums	21-11
Beispiel für streng typisierte VI-Refnums	21-12

Kapitel 22 Attributknoten

Erstellen von Attributknoten	22-1
Hilfe zu Attributknoten	22-6
Grundattribute	22-7
Das Attribut “Visible” (Sichtbar)	22-7
Das Attribut “Disabled” (Deaktiviert)	22-7
Das Attribut “Key Focus” (Fokus)	22-8
Das Attribut “Blinking” (Blinken)	22-8
Das Attribut “Position”	22-9
Das Attribut “Bounds” (Grenzen) (schreibgeschützt)	22-9
Beispiele für Sonderattribute von Bedien- oder Anzeigeelementen.....	22-10
Ändern der Plotfarbe eines Diagramms.....	22-10
Einstellen der Strings eines booleschen Attributs	22-11
Einstellen der Strings für Ring-Steuerungen	22-11
Verwendung eines Doppelklick-Listenelementes	22-12
Selektive Optionsauswahl für Benutzer	22-13
Programmatisches Lesen der Cursor von Graphen	22-14

Kapitel 23

Globale und lokale Variablen

Globale Variablen	23-1
Lokale Variablen.....	23-4

Teil IV—Fortgeschrittene G-Themen

Kapitel 24

Benutzerspezifische Bedienelemente und Typendefinitionen

Benutzerspezifische Bedienelemente	24-2
Verwendung des Bedienelement-Editors	24-2
Verwenden von Änderungen eines benutzerspezifischen Bedienelements.....	24-4
Gültige benutzerspezifische Bedienelemente	24-4
Speichern benutzerspezifischer Bedienelemente	24-5
Verwendung von benutzerspezifischen Bedienelementen	24-5
Erstellen von Icons	24-5
Unabhängige Instanzen von benutzerspezifischen Bedienelementen	24-6
Anpassungsmodus	24-6
Unabhängige Bestandteile	24-7
Das Element-Fenster des Bedienelement-Editors	24-8
Popup-Menüs für verschiedene Elemente im Anpassungsmodus.....	24-10
Gestalterische Elemente	24-10
Gestalterische Elemente mit mehreren Bildern	24-13
Gestalterische Elemente mit unabhängigen Bildern	24-14
Textelemente	24-15
Bedienelemente als Elemente	24-16
Hinzufügen von gestalterischen Elementen zu benutzerspezifischen Bedienelementen	24-17
Vorbehalte gegen benutzerspezifische Bedienelemente	24-18
Typendefinitionen	24-19
Allgemeine Typendefinition: gültige Datentypen	24-19
Strikte Typendefinition: alle Parameter müssen übereinstimmen.....	24-20
Typendefinitionen im Blockdiagramm.....	24-21
Erstellen von Typendefinitionen	24-21
Verwendung von Typendefinitionen	24-21
Aktualisieren von Typendefinitionen	24-22
Suchen von Typendefinitionen.....	24-23
Cluster-Typendefinitionen.....	24-23

Kapitel 25

Aufrufen von Code aus anderen Sprachen

Ausführen anderer Anwendungen von Ihren VIs aus	25-1
Verwendung des Aufrufs externer Bibliotheken	25-2
Verwendung von CINs (Code Interface Nodes)	25-3
Aufruf externer Bibliotheken	25-3
Aufrufkonventionen (Windows)	25-5
Parameterlisten	25-5
Aufrufen von Funktionen, die andere Datentypen erwarten	25-8
LabWindows/CVI-Funktions-Panel-Konverter	25-9
Konvertierungsprozeß	25-10

Kapitel 26

Überblick über das G-Ausführungssystem

Multitasking	26-1
Multithreading	26-2
Das G-Ausführungssystem	26-3
Grundlegendes Ausführungssystem	26-4
Verwalten der Benutzerschnittstelle bei einer Singlethread-Anwendung	26-4
Multithread-Anwendung und mehrere Ausführungssysteme	26-5
Synchrone/blockierende Knoten	26-7
Festlegen von Taskprioritäten	26-8
Wartefunktionen und die Festlegung von Taskprioritäten	26-8
VI-Einstellungen und Priorität	26-9
Prioritäten im Thread der Benutzeroberfläche und im Singlethread-Ausführungssystem	26-10
Prioritäten in anderen Ausführungssystemen des Multithread-Systems	26-11
Prioritätsstufe "Subroutine"	26-12
Ablaufinvariante Ausführung	26-13
Anwendungsbeispiele der Ablaufinvarianz	26-15
Verwendung eines VIs mit Wartefunktion	26-15
Verwendung eines Speicher-VIs, das nicht auf die Freigabe von Daten ausgelegt ist	26-16
Abstimmung des Zugriffs auf globale und lokale Variablen und externe Ressourcen	26-18
Wettbewerbsbedingungen	26-18
Funktionale globale Variablen	26-19
Semaphore	26-21
Andere Abstimmungsfunktionen	26-23
Allgemeine Empfehlungen zur Verwendung von Ausführungssystemen und Prioritäten	26-24

Kapitel 27

Verwalten Ihrer Anwendungen

Dateianordnung unter Verwendung von VI-Bibliotheken.....	27-1
Sichern von Dateien	27-2
Verteilung von VIs.....	27-3
Paßwortgeschützte VIs	27-4
Dialogfeld “Mit Optionen speichern”	27-6
Entwerfen von Anwendungen durch mehrere Entwickler.....	27-9
Aufbewahrung der Originalkopien	27-9
VI-Historiefenster.....	27-10
Revisionsnummern.....	27-12
Rücksetzen der Historie-Informationen	27-13
Drucken der Historie-Informationen.....	27-14
Einstellung von verwandten Dialogfeldoptionen für VI-Einstellungen und Voreinstellungen.....	27-14

Kapitel 28

Leistung

VI-Leistungsprofile.....	28-1
Anzeigen der Ergebnisse	28-3
Timing-Informationen	28-4
Speicherinformationen	28-6
VI-Ausführungsgeschwindigkeit.....	28-7
Eingang/Ausgang	28-7
Bildschirmanzeige	28-8
Sonstige Themen	28-10
Parallele Diagramme.....	28-10
Overhead für SubVIs	28-11
Unnötige Berechnungen in Schleifen	28-12
VI-Speicherauslastung.....	28-14
Virtueller Speicher.....	28-15
Macintosh-Speicher.....	28-16
Speicherverwaltung der VI-Komponenten.....	28-16
Datenfluß-Programmierung und Datenpuffer	28-17
Überwachung der Speicherauslastung.....	28-20
Regeln zur besseren Speicherauslastung	28-21
Speicher bei Frontpanels.....	28-22
SubVIs können den Datenspeicher wieder verwenden.....	28-24
Lokale Variablen können den Datenspeicher nicht wieder verwenden	28-24
Globale Variablen behalten stets Kopien ihrer Daten.....	28-24
Wann Speicher freigegeben wird.....	28-25

Wann Ausgänge Eingangspuffer wieder verwenden können.....	28-26
Konsistente Datentypen.....	28-26
Wie Daten des richtigen Typs erzeugt werden.....	28-27
Vermeiden Sie eine ständige Größenänderung von Daten...	28-28
Beispiel 1: Erstellen von Arrays	28-29
Beispiel 2: Durchsuchen von Strings	28-31
Entwicklung effizienter Datenstrukturen	28-34
Fallstudie 1: Vermeiden komplizierter Datentypen	28-36
Fallstudie 2: Globale Tabelle gemischter Datentypen	28-38
Offensichtliche Implementierung	28-38
Alternative Implementierung 1.....	28-39
Alternative Implementierung 2.....	28-39
Fallstudie 3: Eine statische globale Tabelle mit Strings	28-41

Kapitel 29

Portierbarkeit und Lokalisierungsthemen

Portierbare und nicht-portierbare VIs.....	29-1
Portieren zwischen Plattformen.....	29-2
Trennzeichenunterschiede.....	29-3
Auflösungs- und Schriftartunterschiede.....	29-3
Überlappen von Labels	29-5
Bildunterschiede.....	29-5
VI-Lokalisierung	29-6
Importieren und Exportieren von VI-Strings.....	29-7
Syntax der VI-Stringdatei	29-8
Bearbeiten von VI-Fenstertiteln.....	29-14
Die Dezimaltrennzeichen Punkt und Komma.....	29-14
Formatiere Datum-/Zeit-String	29-15
Portieren zwischen Anwendungen in der Sprache G	29-15
LabVIEW in BridgeVIEW.....	29-15
BridgeVIEW in LabVIEW.....	29-16

Anhang A

Datenspeicherformate

Datenformate für Frontpanel-Bedien- und Anzeigeelemente	A-1
Boolesche Ausdrücke	A-1
Numerische Elemente	A-2
Extended	A-2
Double	A-3
Single	A-3
Long-Integer-Wert.....	A-3

Word-Integer-Wert.....	A-3
Byte-Integer-Wert	A-3
Arrays	A-4
Strings.....	A-5
Pfade.....	A-5
Cluster.....	A-5
Typ-Deskriptoren.....	A-7
Datentypen.....	A-8
Array	A-10
Cluster	A-11
Geglättete Daten	A-12
Skalare	A-13
Strings, Handles und Pfade.....	A-13
Arrays	A-14
Cluster.....	A-14

Anhang B

Häufig gestellte Fragen zu G

Diagramme und Graphen.....	B-1
Fehlermeldungen und Systemabstürze	B-4
Plattformthemen und Kompatibilitäten	B-6
Drucken	B-7
Verschiedenes	B-10

Anhang C

Kundenbetreuung

Glossar

Stichwörterverzeichnis

Tabellen

Tabelle 4-1.	Fehlermeldungen	4-11
Tabelle 4-2.	Breakpoint-Plazierung	4-28
Tabelle 5-1.	Resultierende JPEG-Dateien	5-14
Tabelle 5-2.	Beispiele für Bildbenennungsschemata	5-14
Tabelle 6-1.	Tags von Anwendungselementen	6-20

Tabelle 7-1.	Server: TCP/IP-Zugriff.....	7-27
Tabelle 7-2.	Server: Listeneinträge für exportierte VIs	7-29
Tabelle 9-1.	Bereichsoptionen für Fließkommazahlen	9-5
Tabelle 9-2.	Basiseinheiten	9-28
Tabelle 9-3.	Abgeleitete Einheiten mit Sondernamen	9-28
Tabelle 9-4.	Zusätzliche Einheiten, die mit SI-Einheiten verwendet werden	9-29
Tabelle 9-5.	CGS-Einheiten	9-29
Tabelle 9-6.	Sonstige Einheiten	9-30
Tabelle 11-1.	G ‘\’-Codes	11-4
Tabelle 17-1.	Terminalsymbole für Bedien- und Anzeigeelemente in G	17-2
Tabelle 20-1.	Funktionen des Formelknotens	20-5
Tabelle 20-2.	Fehler in Formelknoten	20-10
Tabelle 29-1.	Beschreibungen der VI-Tags	29-8
Tabelle 29-2.	Inhalte des Frontpanels.....	29-9
Tabelle 29-3.	Tags für das [Bedienelement]	29-10
Tabelle 29-4.	Standarddaten für Strings	29-12
Tabelle 29-5.	Tag-Beschreibungen für Tabellenzellen, Graphplotnamen und Cursornamen	29-13
Tabelle A-1.	Numerische Skalar-Datentypen	A-8
Tabelle A-2.	Nicht numerische Datentypen	A-9

Abbildungen

Abbildung 2-1.	Dialogfeld “Front Panel Label”	2-16
Abbildung 2-2.	Popup-Menü mit ausgewähltem Label	2-16
Abbildung 2-3.	Objekt im Blockdiagramm mit sichtbarem Titel	2-17
Abbildung 2-4.	Kontextsensitives Hilfe-Dialogfeld	2-17
Abbildung 5-1.	Dialogfeld Dokumentation drucken	5-11
Abbildung 5-2.	Dialogfeld Dokumentation drucken, RTF-Datei	5-12
Abbildung 5-3.	Dialogfeld Dokumentation drucken, HTML-Datei	5-12
Abbildung 6-1.	Menü-Editor.....	6-9
Abbildung 6-2.	Auswahl eines Anwendungselements	6-11
Abbildung 6-3.	Funktion “Menüauswahl bekommen”	6-14
Abbildung 6-4.	Funktion “Aktivieren der Menüüberwachung”	6-15
Abbildung 6-5.	Dynamisches Einfügen eines Menüs	6-18

Abbildung 14-1. Popup-Menü in der Elementanzeige aufrufen und ein Boolesches Bedienelement auswählen.....	14-6
Abbildung 14-2. Maustaste loslassen. Das Menü wird ausgeblendet, und der Cursor nimmt die Gestalt des Icons “Fenster rollen” an.....	14-7
Abbildung 14-3. Ziehen Sie das Icon “Fenster rollen” in die Elementanzeige.....	14-7
Abbildung 14-4. Klicken Sie die Maustaste. Das Bedienelement wird innerhalb der Anzeige plaziert.	14-7
Abbildung 19-1. Unterschiedliche Typen in einer Case-Struktur	19-15
Abbildung 19-2. Menüpunkte des Popup-Menüs für eine Case-Struktur	19-17
Abbildung 19-3. Dialogfeld “Cases neu anordnen”	19-17
Abbildung 29-1. VI-Lokalisierungsbeispiel	29-7

Über dieses Handbuch

Das *Referenzhandbuch zur Programmierung in G* beschreibt, wie mit der Programmiersprache G virtuelle Instrumente erstellt, bearbeitet und ausgeführt werden. Dieses Handbuch erläutert das Frontpanel und Blockdiagramm, numerische und Boolesche Bedien- und Anzeigeelemente und Bedien- und Anzeigeelemente vom Typ String, Array und Cluster, Graphen und Diagramme und das Verbinden des Blocksdiagramms.

Gliederung dieses Handbuchs

Dieses Handbuch behandelt vier Themenbereiche. Kapitel 1 bis 7 erklären grundlegende Konzepte der Sprache G. Kapitel 8 bis 16 erläutern die Verwendungsweise von Frontpanelobjekten. Kapitel 17 bis 23 stellen die Programmierung von Objekten über Blockdiagramme und entsprechende Techniken dar, und Kapitel 24 bis 27 behandeln fortgeschrittene Themen der Sprache G.

Teil I—Grundlegende Konzepte der Sprache G

Dieser Abschnitt enthält grundlegende Informationen über die Programmierung in G und die Erstellung, Bearbeitung und benutzerspezifische Anpassung von virtuellen Instrumenten (VIs).

Teil I, *Grundlegende Konzepte der Sprache G*, enthält folgende Kapitel.

- Kapitel 1, *Einführung in die Programmierung in G*, beschreibt den besonderen Ansatz von G zur Programmierung. Außerdem wird erklärt, wie man G zur Entwicklung von Programmen einsetzt.
- Kapitel 2, *Bearbeiten von VIs*, beschreibt die grundlegenden Aspekte der Erstellung und Verwendung von VIs, einschließlich Informationen über Paletten und Menüs. Außerdem werden so grundlegende Aufgaben wie die Erstellung von Objekten, das Wechseln von Werkzeugen und das Öffnen, Ausführen und Speichern von VIs behandelt.
- Kapitel 3, *Verwendung von SubVIs*, beschreibt das Konzept des hierarchischen Designs in G-Anwendungen und erklärt zwei Methoden zur Erstellung von SubVIs. Das Kapitel erläutert außerdem zwei Dienstprogramme—das Hierarchie-Fenster, welches die Hierarchie Ihrer VIs anzeigt, und das Such-Utility, welches Occurrences von SubVIs sowie andere von Ihnen spezifizierte Objekte oder Strings aus Text sucht.

- Kapitel 4, *Ausführung und Debugging von VIs und SubVIs*, beschreibt die Ausführung und das Debugging von VIs und erklärt die Einstellungen von VIs und SubVIs für besondere Ausführungsmodi.
- Kapitel 5, *Drucken und Dokumentieren von VIs*, beschreibt eine Vielzahl von Themen im Zusammenhang mit Druckvorgängen und der VI-Dokumentation.
- Kapitel 6, *Einrichten von VIs und SubVIs*, beschreibt, wie Sie das Verhalten von VIs mit den Dialogfeldern **VI-Einstellungen** und **Einstellungen SubVI-Knoten** benutzerspezifisch anpassen können.
- Kapitel 7, *Anpassen Ihrer Umgebung*, beschreibt, wie Sie Ihre Umgebung benutzerspezifisch anpassen können, indem Sie die Editor-Voreinstellungen für Funktionen wie Drucken, Anzeigen und **Rückgängig** einstellen können und indem Sie den Inhalt der Paletten **Elemente** und **Funktionen** ändern.

Teil II—Frontpanel-Objekte

Dieser Abschnitt enthält Informationen über Frontpanel-Objekte, Bedien- und Anzeigeelemente und ActiveX-Bedienelemente.

Teil II, *Frontpanel-Objekte*, enthält folgende Kapitel.

- Kapitel 8, *Einführung in Frontpanel-Objekte*, bietet eine Einführung in das Frontpanel und seine zwei Komponenten—Bedien- und Anzeigeelemente. Außerdem erläutert es, wie man Graphiken von anderen Programmen zur Verwendung in Ihren Bedienelementen importieren kann.
- Kapitel 9, *Numerische Bedien- und Anzeigeelemente*, beschreibt, wie man die verschiedenen Stile der numerischen Bedien- und Anzeigeelemente erstellt, ausführt und konfiguriert.
- Kapitel 10, *Boolesche Bedien- und Anzeigeelemente*, beschreibt, wie man Boolesche Bedien- und Anzeigeelemente erstellt, ausführt und konfiguriert.
- Kapitel 11, *Bedien- und Anzeigeelemente vom Typ String*, beschreibt, wie man Bedien- und Anzeigeelemente vom Typ String und die Tabelle verwendet. Diese Objekte können über die Palette **Elemente»String & Tabelle** aufgerufen werden.
- Kapitel 12, *Bedien- und Anzeigeelemente vom Typ Pfad und Refnum*, beschreibt, wie Dateipfad-Bedienelemente und Refnums verwendet werden, die in der Palette **Elemente»Pfad & RefNum** zur Verfügung stehen.

- Kapitel 13, *Bedien- und Anzeigeelemente vom Typ Liste und Ring*, beschreibt die Bedien- und Anzeigeelemente vom Typ Listenfeld und Ring, die in der Palette **Elemente»Liste & Ring** zur Verfügung stehen.
- Kapitel 14, *Bedien- und Anzeigeelemente vom Typ Array und Cluster*, beschreibt die Verwendungsweise von Arrays und Clustern. Der Zugriff auf Arrays und Cluster erfolgt über die Palette **Elemente»Array & Cluster**.
- Kapitel 15, *Bedien- und Anzeigeelemente vom Typ Graph und Diagramm*, beschreibt die Erstellung und Verwendung von Anzeigeelementen in Form von Graphen und Diagrammen mit Hilfe der Palette **Elemente»Graph**.
- Kapitel 16, *ActiveX-Bedienelemente*, erläutert die Funktionalität von ActiveX-Containern, mit denen die Kommunikation zwischen G-basierter Software und anderen Anwendungen verbessert wird.

Teil III—Blockdiagramm-Programmierung

Dieser Abschnitt enthält Informationen über die Komponenten, die zum Erstellen und Manipulieren eines Blockdiagramms in G erforderlich sind.

Teil III, *Blockdiagramm-Programmierung*, enthält folgende Kapitel.

- Kapitel 17, *Einführung in das Blockdiagramm*, beschreibt Terminals und Knoten—zwei der drei Elemente, aus denen Sie ein Blockdiagramm erstellen. Das dritte Element, die Verbindung, wird in Kapitel 18, *Verbinden des Blockdiagramms*, besprochen.
- Kapitel 18, *Verbinden des Blockdiagramms*, erläutert das Vorgehen beim Verbinden von Terminals im Blockdiagramm.
- Kapitel 19, *Strukturen*, beschreibt die Verwendung der For-Schleife, While-Schleife, Case-Struktur und Sequenzstruktur. Diese Strukturen befinden sich in der Palette **Funktionen»Strukturen**.
- Kapitel 20, *Formelknoten*, beschreibt die Verwendung von Formelknoten zur Ausführung von mathematischen Formeln im Blockdiagramm. Der Formelknoten steht in der Palette **Funktionen»Strukturen** zur Verfügung.
- Kapitel 21, *VI-Server*, beschreibt den Mechanismus zur programmatischen Steuerung von VIs und Anwendungen. Das Kapitel beschreibt außerdem das Verfahren der Remote-Steuerung von VIs oder Anwendungen.

- Kapitel 22, *Attributknoten*, beschreibt die Verwendung von Attributknoten, um Attribute von Frontpanel-Bedienelementen programmatisch zu setzen und zu lesen. Einige nützliche Eigenschaften umfassen Anzeigefarben, die Sichtbarkeit von Bedienelementen, Menü-Strings für ein Ring-Element, Plotfarben für Graphen oder Diagramme und Cursor für Graphen.
- Kapitel 23, *Globale und lokale Variablen*, beschreibt, wie globale und lokale Variablen definiert und verwendet werden. Verwenden Sie globale Variablen, um von mehreren VIs aus leicht auf einen bestimmten Satz von Werten zuzugreifen. Lokale Variablen dienen innerhalb eines einzigen VIs einem ähnlichen Zweck.

Teil IV—Fortgeschrittene G-Themen

Dieser Abschnitt enthält Informationen über fortgeschrittene Funktionen in G und Techniken, die zur Erstellung virtueller Instrumente verwendet werden können.

Teil IV, *Fortgeschrittene G-Themen*, enthält folgende Kapitel.

- Kapitel 24, *Benutzerspezifische Bedienelemente und Typendefinitionen*, bietet eine Einführung in benutzerspezifische Bedienelemente und Typendefinitionen.
- Kapitel 25, *Aufrufen von Code aus anderen Sprachen*, beschreibt verschiedene Methoden zum Aufrufen von Code, der in anderen Sprachen geschrieben wurde.
- Kapitel 26, *Überblick über das G-Ausführungssystem*, beschreibt das VI-Multitasking und die Ausführung von VIs.
- Kapitel 27, *Verwalten Ihrer Anwendungen*, beschreibt, wie die Dateien in Ihren G-Anwendungen verwaltet werden können.
- Kapitel 28, *Leistung*, ist in drei Abschnitte unterteilt. Der erste Abschnitt beschreibt den Performance Profiler, eine Funktion, mit der Daten zur Ausführungszeit Ihres VI angezeigt und Singlethread-, Multithread- und Mehrprozessor-Anwendungen überwacht werden können. Im zweiten Abschnitt werden die Faktoren beschrieben, die sich auf die Ausführungszeit auswirken. Im dritten Abschnitt werden Faktoren beschrieben, die sich auf die Speicherauslastung auswirken.
- Kapitel 29, *Portierbarkeit und Lokalisierungsthemen*, behandelt Themen im Zusammenhang mit der Übertragung von VIs zwischen Plattformen und die VI-Lokalisierung.

Anhänge, Glossar und Stichwörterverzeichnis

- Anhang A, *Datenspeicherformate*, beschreibt die Formate, in denen Sie Daten speichern können.
- Anhang B, *Häufig gestellte Fragen zu G*, enthält Antworten auf einige Fragen, die häufig von Benutzern von G gestellt werden.
- Anhang C, *Kundenbetreuung*, enthält Formulare, die es Ihnen erleichtern sollen, alle erforderlichen Informationen zusammenzustellen, damit National Instruments Ihnen bei der Lösung Ihrer technischen Probleme helfen kann. Außerdem finden Sie ein Formular, das Sie für Kommentare zur Produktdokumentation verwenden können.
- Das *Glossar* enthält eine alphabetische Liste und Beschreibung von Begriffen, die in diesem Handbuch verwendet werden.
- Der *Stichwörterverzeichnis* enthält eine alphabetische Liste von Schlüsselbegriffen und -themen in diesem Handbuch mit Seitenangabe.

Schreibkonventionen in diesem Handbuch

In diesem Handbuch werden die folgenden Schreibkonventionen verwendet:

- <> Spitze Klammern werden für Bezeichnungen einer Taste auf der Tastatur verwendet — zum Beispiel <Umschalt>-Taste. Spitze Klammern mit Zahlen, die durch eine Ellipse abgetrennt sind, stellen einen Wertebereich dar, der mit einem Bit oder einem Signalnamen assoziiert ist — zum Beispiel DBIO<3.0>.
- [] Eckige Klammern werden bei optionalen Elementen verwendet — zum Beispiel [Antwort].
- Ein Bindestrich zwischen zwei oder mehr Schlüsselbezeichnungen, die in spitzen Klammern stehen, bedeutet, daß Sie die bezeichneten Tasten gleichzeitig drücken müssen — zum Beispiel <Strg-Alt-Entf>.
- » Das Symbol » führt Sie durch verschachtelte Menüpunkte und Dialogfeldoptionen zu einer abschließenden Option. Die Sequenz **Datei»Seite einrichten»Optionen»Schriftarten ersetzen** weist Sie dazu an, das Menü **Datei** aufzurufen, den Menüpunkt **Seite einrichten** auszuwählen, **Optionen** zu wählen und schließlich die Option **Schriftarten ersetzen** aus dem letzten Dialogfeld auszuwählen.

- ◆ Das Symbol ◆ gibt an, daß der darauffolgende Text nur auf ein spezifisches Produkt, ein spezifisches Betriebssystem oder eine spezifische Softwareversion zutrifft.

- fett** Fettgedruckter Text wird für Bezeichnungen von Menüs, Menüpunkten, Parametern, Dialogfeldern, Dialogfeldschaltflächen oder -optionen, Icons, Fenstern, Windows 95-Registern oder LEDs verwendet.

- fett kursiv** Fettgedruckter Text in Kursivschrift wird für Aktivitätsziele, Hinweise, Achtungs- oder Warnhinweise verwendet.

- <Strg> Tastennamen fangen mit Großbuchstaben an.

- Kursiv* Kursivtext bezieht sich auf Variablen, Betonungen, Querverweise oder eine Einführung zu einem Schlüsselkonzept. Diese Schriftart wird auch für Text verwendet, für den Sie ein entsprechendes Wort oder einen entsprechenden Wert vorgeben, wie beispielsweise bei Windows 3.x.


- Konstantschrift Text in dieser Schriftart wird für Text oder Zeichen verwendet, die genauso auf der Tastatur eingegeben werden müssen, für Code-Abschnitte, Programmierbeispiele und Syntax-Beispiele. Diese Schriftart wird außerdem für Eigennamen von Diskettenlaufwerken, Pfaden, Verzeichnissen, Programmen, Unterprogrammen, Subroutinen, Gerätenamen, Funktionen, Operationen, Variablen, Dateinamen und -erweiterungen und für Anweisungen und Kommentare, die aus Programmen stammen, verwendet.

- Konstantschrift**
fett Fettgedruckter Text in dieser Schriftart wird für Meldungen und Reaktionen verwendet, die der Computer automatisch auf dem Bildschirm ausgibt. Diese Schriftart hebt außerdem Zeilen mit Code hervor, die sich von anderen Beispielen unterscheiden.

- Konstantschrift*
kursiv Kursivtext in dieser Schriftart bedeutet, daß Sie anstelle dieser Zeichen die entsprechenden Wörter oder Werte eingeben müssen.

- Pfade Pfade in diesem Handbuch werden durch umgekehrte Schrägstriche bezeichnet (\), um Laufwerknamen, Verzeichnisse, Ordner und Dateien abzugrenzen.

- Plattform** Text in dieser Schriftart bezeichnet Informationen, die sich auf eine spezifische Plattform beziehen.

-  Wenn sich dieses Symbol links von fettgedrucktem Text in Kursivschrift befindet, handelt es sich um einen Hinweis, der Sie auf wichtige Informationen aufmerksam macht.



Wenn sich dieses Symbol links von fettgedrucktem Text in Kursivschrift befindet, handelt es sich um eine Mahnung zur Vorsicht, die Sie auf Vorsichtsmaßnahmen aufmerksam macht, die getroffen werden müssen, um Verletzungen, Datenverluste oder einen Systemabsturz zu vermeiden.

Abkürzungen, Akronyme, metrische Präfixe, Mnemonik, Symbole und Begriffe sind im [Glossar](#) aufgeführt.

Verwandte Dokumentationsmaterialien

Die folgenden Unterlagen könnten sich bei der Lektüre dieses Handbuchs als nützlich erweisen:

- *LabVIEW Benutzerhandbuch*
- *BridgeVIEW User Manual*
- *Grundlagen der Datenerfassung mit LabVIEW*
- *LabVIEW Funktionen- und VI-Referenzhandbuch*
- *LabVIEW Kurzanleitung*
- *LabVIEW oder BridgeVIEW Online-Referenz* – verfügbar durch Wahl von **Hilfe»Online-Referenz**
- *LabVIEW Online Tutorial (nur für Windows)*, das Sie vom LabVIEW-Dialogfeld starten
- *Erste Schritte mit LabVIEW*
- *Referenzkarte G-Programmierung*
- *VXI VI-Referenzhandbuch*
- *LabVIEW Code Interface Reference Manual*, das im PDF-Format (portablen Dokumentformat) auf Ihren Softwareprogramm-Disketten bzw. der CD erhältlich ist.

Mitteilungen von Kunden

National Instruments begrüßt Ihre Kommentare zu unseren Produkten und Handbüchern. Wir sind besonders auch an den Anwendungen interessiert, die Sie mit unseren Produkten entwickeln, und wir möchten Ihnen behilflich sein, falls Sie auf Probleme stoßen. Um Ihnen den Kontakt mit uns so einfach wie möglich zu machen, enthält dieses Handbuch Formulare, in denen Sie Ihre Kommentare und Konfigurationsdaten eintragen können. Diese Formulare befinden sich in Anhang C, [Kundenbetreuung](#), am Ende dieses Handbuchs.

Grundlegende Konzepte der Sprache G

Dieser Abschnitt enthält grundlegende Informationen über die Programmierung in G und die Erstellung, Bearbeitung und benutzerspezifische Anpassung von virtuellen Instrumenten (VIs).

Teil I, *Grundlegende Konzepte der Sprache G*, enthält folgende Kapitel.

- Kapitel 1, *Einführung in die Programmierung in G*, beschreibt den besonderen Ansatz von G zur Programmierung. Außerdem wird erklärt, wie man G zur Entwicklung von Programmen einsetzt.
- Kapitel 2, *Bearbeiten von VIs*, beschreibt die grundlegenden Aspekte der Erstellung und Verwendung von VIs, einschließlich Informationen über Paletten und Menüs. Außerdem werden so grundlegende Aufgaben wie die Erstellung von Objekten, das Wechseln von Werkzeugen und das Öffnen, Ausführen und Speichern von VIs behandelt.
- Kapitel 3, *Verwendung von SubVIs*, beschreibt das Konzept des hierarchischen Designs in G-Anwendungen und erklärt zwei Methoden zur Erstellung von SubVIs. Das Kapitel erläutert außerdem zwei Dienstprogramme—das Hierarchie-Fenster, welches die Hierarchie Ihrer VIs anzeigt, und das Such-Utility, welches Occurrences von SubVIs sowie andere von Ihnen spezifizierte Objekte oder Strings aus Text sucht.
- Kapitel 4, *Ausführung und Debugging von VIs und SubVIs*, beschreibt die Ausführung und das Debugging von VIs und erklärt die Einstellungen von VIs und SubVIs für besondere Ausführungsmodi.
- Kapitel 5, *Drucken und Dokumentieren von VIs*, beschreibt eine Vielzahl von Themen im Zusammenhang mit Druckvorgängen und der VI-Dokumentation.

- Kapitel 6, *Einrichten von VIs und SubVIs*, beschreibt, wie Sie das Verhalten von VIs mit den Dialogfeldern **VI-Einstellungen** und **Einstellungen SubVI-Knoten** benutzerspezifisch anpassen können.
- Kapitel 7, *Anpassen Ihrer Umgebung*, beschreibt, wie Sie Ihre Umgebung benutzerspezifisch anpassen können, indem Sie die Editor-Voreinstellungen für Funktionen wie Drucken, Anzeigen und **Rückgängig** einstellen können und indem Sie den Inhalt der Paletten **Elemente** und **Funktionen** ändern.

Einführung in die Programmierung in G

Dieses Kapitel erläutert den einzigartigen Ansatz der Programmierung mit G. Es erläutert ferner den Einstieg in die Entwicklung von Programmen mit G.

Was ist G?

G ist die Programmiersprache im Herzen von LabVIEW. Es ist außerdem wesentlicher Bestandteil von BridgeVIEW, der Umgebung zur Entwicklung von Anwendungen in der Automatisierung von National Instruments.

Wie bei C oder BASIC handelt es sich bei G um eine Allzweck-Programmiersprache mit umfangreichen Funktionsbibliotheken für alle Aufgaben der Programmierung. G beinhaltet Bibliotheken für Datenerfassung, GPIB und serielle Instrumentensteuerung, Datenanalyse, Datenpräsentation und Datenspeicherung. G verfügt außerdem über herkömmliche Debugging-Werkzeuge für Programme, damit Sie Breakpoints setzen können, eine Animation der Ausführung erstellen können, um zu sehen, wie Daten das Programm durchlaufen, und das Programm in Einzelschritten ausführen können, um Debugging und Programmentwicklung zu erleichtern.

G unterscheidet sich von diesen jedoch in einer wichtigen Hinsicht. Während andere Programmiersprachen *textbasiert* sind, ist G *graphisch*.

Komponenten eines VIs

G ist ein Allzweck-Programmiersystem, verfügt aber auch über Funktionsbibliotheken und Entwicklungswerkzeuge, die speziell für Datenerfassung und Instrumentensteuerung vorgesehen sind. G-Programme werden als *Virtuelle Instrumente (VIs)* bezeichnet, da sie in ihrer Erscheinungsform und Arbeitsweise tatsächliche Instrumente

nachahmen können. VIs sind Funktionen herkömmlicher Programmiersprachen ähnlich.

Ein VI besteht aus einer interaktiven Benutzerschnittstelle, einem Datenflußdiagramm, das als Quellcode dient, und Icon-Verbindungen, mit denen das VI so eingerichtet wird, daß es von VIs höherer Ebenen aufgerufen werden kann. Genau genommen sind VIs folgendermaßen aufgebaut:

- Die interaktive Benutzerschnittstelle eines VIs wird als *Frontpanel* bezeichnet, da es das Panel eines tatsächlichen Instruments simuliert. Am Frontpanel können sich Drehknöpfe, Drucktasten, Graphen und andere Bedien- und Anzeigeelemente befinden. Sie geben Daten mit Hilfe einer Maus und Tastatur ein und sehen sich die Ergebnisse dann am Computerbildschirm an.
- Das VI erhält Anweisungen von einem *Blockdiagramm*, das Sie in G erstellen. Das Blockdiagramm ist eine bildliche Lösung einer Programmieraufgabe. Es stellt außerdem den Quellcode des VIs dar.
- VIs sind hierarchisch und modular. Sie können als Programme höchster Ebene oder als Unterprogramme innerhalb von anderen Programmen und Unterprogrammen verwendet werden. Ein VI innerhalb eines anderen VIs wird als *SubVI* bezeichnet. *Icon und Anschluß* eines VIs funktionieren wie eine graphische Parameterliste, die es anderen VIs ermöglicht, Daten an ein SubVI zu übergeben.

G macht mit diesen Merkmalen den besten Gebrauch vom Begriff der *modularen Programmierung*. Sie teilen dabei eine Anwendung in eine Reihe von Aufgaben auf, die Sie wiederum solange aufteilen, bis aus einer komplizierten Anwendung eine Reihe von einfachen Unteraufgaben wird. Sie erstellen für jede auszuführende Unteraufgabe ein VI und kombinieren diese VIs in einem anderen Blockdiagramm, um eine größere Aufgabe zu bewältigen. Das VI der höchsten Ebene enthält schließlich eine Sammlung von SubVIs, die Anwendungsfunktionen darstellen.

Da Sie jedes SubVI für sich und unabhängig vom Rest der Anwendung ausführen können, ist das Debugging erheblich einfacher. Außerdem führen viele SubVIs der unteren Ebenen Aufgaben durch, die in mehreren Anwendungen üblich sind, so daß Sie einen Sondersatz von SubVIs entwickeln können, der sich gut für zukünftige Anwendungen eignet.

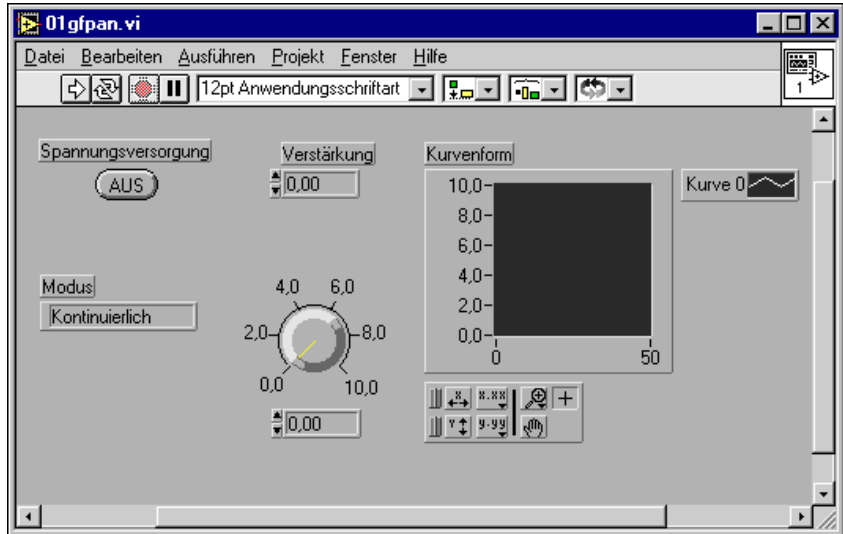
In den nachstehenden Abschnitten werden das Frontpanel, Blockdiagramm, Icon, der Anschluß und andere verwandte Funktionsmerkmale näher erläutert.

**Hinweis**

Beispiele verschiedener Funktionen, die G in Ihrer LabVIEW- oder BridgeVIEW-Anwendung verwendet, finden Sie im Verzeichnis `examples`.

Frontpanel

Die Benutzerschnittstelle eines VIs ist mit der Benutzerschnittstelle eines tatsächlichen Instruments — dem Frontpanel — vergleichbar. Ein Frontpanel eines VIs könnte etwa wie die folgende Abbildung aussehen.



Bei dem Frontpanel eines VIs handelt es sich hauptsächlich um eine Kombination von *Bedien-* und *Anzeigeelementen*. Bedienelemente simulieren Instrumenten-Eingabegeräte und versorgen das Blockdiagramm des VIs mit Daten. Anzeigeelemente simulieren Instrumenten-Ausgabegeräte, die vom Blockdiagramm erfasste oder generierte Daten anzeigen.

Sie bringen auf dem Frontpanel zusätzliche Bedien- und Anzeigeelemente an, indem Sie diese aus der floatenden Palette **Elemente** wählen, die in der folgenden Abbildung zu sehen ist.



Größe, Form und Position eines Bedien- oder Anzeigeelements können angepaßt werden. Außerdem verfügt jedes Bedien- oder Anzeigeelement über ein Popup-Menü, über das Sie verschiedene Attribute ändern oder andere Menüpunkte auswählen können. Der Zugriff auf dieses Popup-Menü erfolgt

- **(Windows und UNIX)** durch Klicken der rechten Maustaste auf dem Objekt.
- **(Macintosh)** durch Klicken der Maustaste und gleichzeitiges Drücken der <Befehlstaste>.

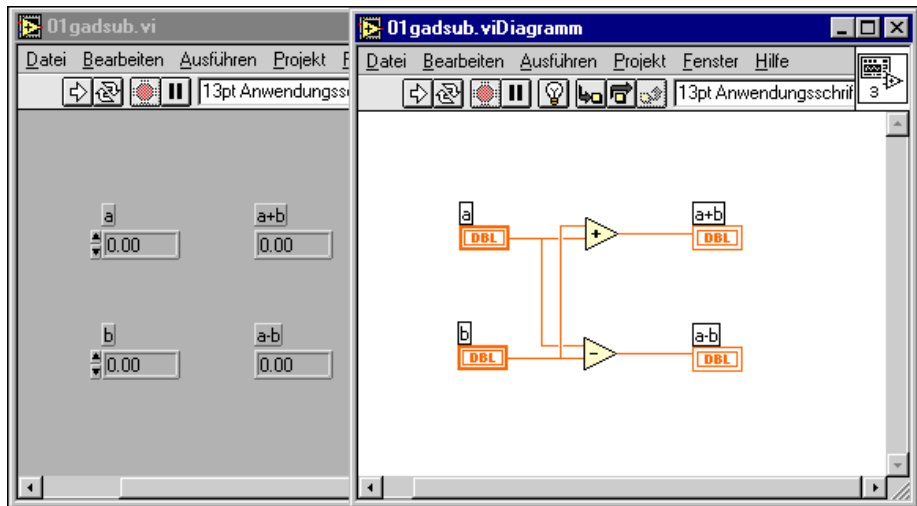
Weitere Informationen zum Erstellen eines Frontpanels finden Sie in Kapitel 2, *Bearbeiten von VIs*, und im Abschnitt *Frontpanel-Objekte* in diesem Handbuch (Kapitel 8 bis 16).

Blockdiagramm

Sie können vom Frontpanel zum Blockdiagramm wechseln, indem Sie **Fenster»Diagramm** aus dem Menü wählen.

Das Diagramm-Fenster beinhaltet das Blockdiagramm des VIs, das gleichzeitig auch der graphische Quellcode eines VIs ist. Sie erstellen das Blockdiagramm durch *Verbinden* der Objekte, die Daten senden oder empfangen, spezielle Funktionen durchführen und den Ausführungsablauf steuern.

Das folgende einfache VI berechnet die Summe bzw. Differenz zweier Zahlen. Das Diagramm zeigt mehrere primäre Programmobjekte des Blockdiagramms — *Knoten*, *Terminals* und *Verbindungen*.



Wenn Sie ein Bedien- oder Anzeigeelement auf das Frontpanel plazieren, erscheint im Blockdiagramm ein dementsprechendes Terminal. Ein zu einem Bedien- oder Anzeigeelement gehörendes Terminal kann nicht gelöscht werden. Das Terminal wird nur dann entfernt, wenn Sie dessen zugehöriges Bedien- oder Anzeigeelement löschen.

Auch für die Icons der Funktionen “Addieren” und “Subtrahieren” gibt es Terminals. Stellen Sie sich Terminals als Eingangs- und Ausgangsanschlüsse vor. Daten, die an den Bedienelementen eingegeben werden (a und b), verlassen das Frontpanel über die Terminals der Bedienelemente im Blockdiagramm. Die Daten treten dann in die Funktionen “Addieren” und “Subtrahieren” ein. Nach Abschluß der internen Rechenvorgänge der Funktionen “Addieren” und “Subtrahieren” erzeugen diese an ihren *Ausgangsterminals* neue Datenwerte. Die Daten werden an die Anzeige-Terminals weitergeleitet und treten erneut in das Frontpanel ein, wo sie angezeigt werden. Terminals, die Daten erzeugen, werden als *Datenausgangsterminals* bezeichnet, und Terminals, die Daten empfangen, werden als *Dateneingangsterminals* bezeichnet.

Knoten sind Elemente der Programmausführung. Sie verhalten sich analog zu Anweisungen, Operatoren, Funktionen und Unterprogrammen in herkömmlichen Programmiersprachen. Bei den Funktionen “Addieren” und “Subtrahieren” handelt es sich um Knotentypen. G verfügt über eine

umfangreiche Funktionsbibliothek für mathematische Operationen, Vergleichsoperationen, Umwandlung, I/O u. a.

Ein anderer Knotentyp ist die *Struktur*. Strukturen sind graphische Darstellungen von Schleifen und Case-Anweisungen herkömmlicher Programmiersprachen. Die Programmiersprache G verfügt über spezielle Knoten, die Links zu externem textbasiertem Code und eine Auswertung textbasierter Formeln ermöglichen.

Verbindungen sind die Datenpfade zwischen Ausgangs- und Auffangterminals. Sie können kein Ausgangsterminal mit einem anderen Ausgangsterminal verbinden, und Sie können kein Auffangterminal mit einem anderen Auffangterminal verbinden. Sie können allerdings eine Quelle mit mehreren Zielen verbinden. Die Farbe oder der Stil jeder Verbindung unterscheidet sich je nach dem Datentyp, der sie durchläuft. Das vorherige Beispiel zeigt den Verbindungsstil eines numerischen skalaren Werts — und zwar eine dünne durchgehende Linie.

Datenfluß ist das beherrschende Prinzip der Programmausführung bei G. Einfach ausgedrückt: ein Knoten wird nur ausgeführt, wenn an all seinen Eingabe-Terminals Daten eingehen; der Knoten versorgt bei Abschluß der Ausführung alle Ausgabe-Terminals mit Daten, und die Daten werden vom Ausgangsterminal sofort an die Auffangterminals übergeben. Die Datenflußmethode der Programmierung unterscheidet sich von der Steuerungsflußmethode der Ausführung eines herkömmlichen Programms insofern, als in letzterem Anweisungen in der Reihenfolge, in der sie verfaßt sind, ausgeführt werden. Die Ausführung nach Steuerungsfluß ist anweisunggetrieben. Die Ausführung nach Datenfluß ist *datengetrieben* oder *datenabhängig*.

Ausführliche Informationen zum Erstellen eines Programms unter Verwendung von Blockdiagramm-Elementen finden Sie im Teil [Blockdiagramm-Programmierung](#) in diesem Handbuch (Kapitel 17 bis 23).

Icon und Anschluß

Wenn ein Icon eines VIs in das Diagramm eines anderen VIs plaziert wird, wird es damit zu einem *SubVI*, dem G-Äquivalent eines Unterprogramms. Die Bedien- und Anzeigeelemente eines SubVIs empfangen Daten vom aufrufenden VI und liefern Daten an das aufrufende VI zurück.

Beim *Anschluß* handelt es sich um einen Satz von Terminals, der den Bedien- und Anzeigeelementen des SubVIs entspricht. Das *Icon* ist

entweder eine bildliche Darstellung für den Zweck des VIs oder eine verbale Beschreibung des VIs oder dessen Terminals.

Der Anschluß ähnelt der Parameterliste eines Funktionsaufrufs; die Anschluß-Terminals fungieren als Parameter. Jedes Terminal entspricht einem bestimmten Bedien- oder Anzeigeelement am Frontpanel. Ein Anschluß empfängt Daten an seinen Eingabe-Terminals und übergibt diese über die Bedienelemente des SubVIs an den Code des SubVIs, oder er empfängt die Ergebnisse von den Anzeigeelementen des SubVIs an seinen Ausgabe-Terminals.

Zu jedem VI wird im Fenster des Frontpanels und des Blockdiagramms in der rechten oberen Ecke des Icon-Felds ein Standard-Icon angezeigt. Das Icon des VIs ist in der folgenden Abbildung zu sehen.



Jedes VI besitzt außerdem einen Anschluß, der durch die Auswahl von **Anschluß anzeigen** aus dem Popup-Menü des Icon-Felds auf dem Frontpanel aufgerufen wird. Wenn Sie den Anschluß zum ersten Mal aufrufen, sehen Sie ein Anschluß-Pattern. Auf Wunsch können Sie ein anderes Pattern auswählen. Der Anschluß besitzt im allgemeinen pro Bedien- oder Anzeigeelement am Frontpanel ein Terminal. Es können bis zu 28 Terminals zugewiesen werden. Sind zukünftige Änderungen am VI wahrscheinlich, bei denen ein neuer Eingang oder Ausgang erforderlich wäre, lassen Sie einige Anschlüsse von Terminals offen.

Weitere Informationen hierzu finden Sie in Kapitel 3, [Verwendung von SubVIs](#).

So erhalten Sie Hilfe

Das Hilfefenster enthält Hilfeinformationen zu Funktionen, Konstanten, SubVIs, Bedien- und Anzeigeelementen und Menüpunkten in Dialogfeldern. Wählen Sie zum Anzeigen des Fensters **Hilfe anzeigen** aus dem Menü **Hilfe**, oder drücken Sie <Strg-h> (**Windows**); <Befehlstaste-h> (**Macintosh**); <Meta-h> (**Sun**) oder <Alt-h> (**HP-UX**). Wenn Ihre Tastatur eine <Help>-Taste besitzt, drücken Sie diese statt dessen. Verschieben Sie den Cursor auf das Icon einer Funktion, einen

SubVI-Knoten oder ein VI-Icon (einschließlich des offenen VIs, das rechts oben im VI-Fenster zu sehen ist), um die Hilfeinformationen anzuzeigen.



Durch Wählen von **Hilfe»Hilfe fixieren** oder Klicken auf das Fixiericon am unteren Rand des Fensters wird der aktuelle Inhalt des Hilfefensters fixiert. Wird dieses fixiert, ändert sich die Anzeige im Hilfefenster nicht, wenn Sie den Cursor auf eine andere Funktion oder ein anderes Icon verschieben. Sie müssen erneut **Hilfe fixieren** wählen oder auf das Fixiericon klicken, um diesen Menüpunkt zu deaktivieren.

Hilfe zu einem Frontpanel

Wenn Sie den Cursor auf ein Bedien- oder Anzeigeelement verschieben, wird die Beschreibung des betreffenden Bedien- oder Anzeigeelements im Hilfefenster angezeigt. Es ist ratsam, bei der Erstellung eines VIs zu jedem Bedien- oder Anzeigeelement eine Beschreibung einzugeben. Weitere Informationen hierzu finden Sie im Abschnitt [Erstellen von Objektbeschreibungen](#) in Kapitel 2, [Bearbeiten von VIs](#).

Wenn Sie mit dem Cursor auf einem VI-Icon rechts oben auf einem Frontpanel kurz pausieren, wird im Hilfefenster Hilfe zu diesem VI angezeigt.

Hilfe zu einem Blockdiagramm

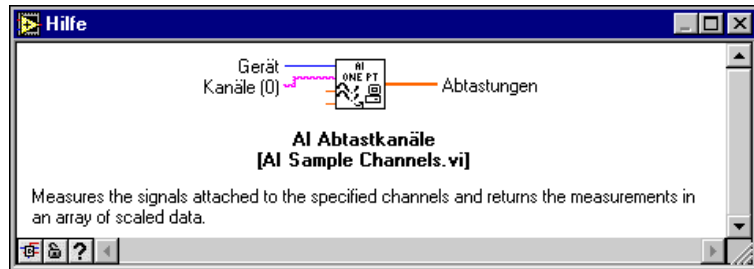
Bei Funktionen und SubVI-Knoten werden das Icon, Eingänge und Ausgänge sowie eine Beschreibung der Funktionsweise des Knotens im Hilfefenster angezeigt.



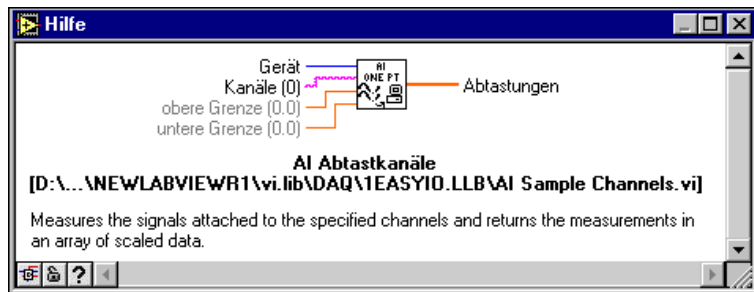
Durch Drücken der zweiten Taste am unteren Rand des Hilfefensters oder Umschalten des Menüpunkts **Hilfe»Einfache Hilfe** kann das Hilfefenster entweder als einfache oder detaillierte Ansicht angezeigt werden.

Die einfache Ansicht betont die wichtigen Verbindungen und stellt die weniger kritischen Verbindungen in den Hintergrund. In dieser Ansicht sind Labels von erforderlichen Verbindungen fett und von empfohlenen Verbindungen normal gedruckt. Bei optionalen Verbindungen wird der Name nicht angezeigt und nur ein kleiner Verbindungsabschnitt gezeichnet, um anzuzeigen, daß weitere Optionen bei Bedarf zur

Verfügung stehen. Die folgende Abbildung zeigt die einfache Ansicht eines VIs zur Datenerfassung.



Bei der detaillierten Ansicht zeigt das Hilfenfenster alle Eingänge mit nach links weisenden Verbindungen und alle Ausgänge mit nach rechts weisenden Verbindungen an. Labels von optionalen Eingaben sind grau geschrieben. Die nachstehende Abbildung zeigt dasselbe VI zur Datenerfassung in der detaillierten Ansicht.



Im detaillierten Modus wird der Standort des Speichermediums des SubVIs unterhalb des Icons angezeigt. Im einfachen Modus wird nur der Name des VIs angezeigt.

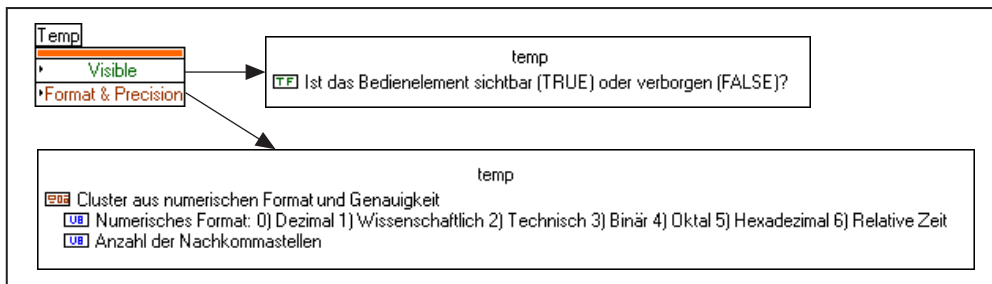
Erfordert eine Funktionseingabe keine Verbindung, wird der Standardwert gewöhnlich neben dem Namen in Klammern angegeben. Kann die Funktion mehrere Datentypen annehmen, zeigt das Hilfenfenster den gebräuchlichsten an.

Die Terminalnamen für SubVI-Knoten sind die Labels der entsprechenden Bedien- und Anzeigeelemente des Frontpanels. Die Standardwerte der SubVI-Eingaben erscheinen nicht automatisch im Hilfenfenster. Es ist daher ratsam, den Standardwert beim Erstellen eines SubVIs bei den Namen von Bedienelementen in Klammern hinzuzufügen.

Wenn Sie das Verbindungswerkzeug auf eine Verbindung plazieren, zeigt das Hilfenfenster den Datentyp der Verbindung an. Verschieben Sie das Verbindungswerkzeug außerdem über verschiedene Bereiche des VI-Icons, wird das entsprechende Anschluß-Terminal im Hilfenfenster hervorgehoben.

Hilfe zu Attributen

Wenn Sie sich über die Bedeutung eines bestimmten Attributs nicht in klaren sind, können Sie das Hilfenfenster verwenden, um die Bedeutung des Attributs, dessen Datentyp und zulässige Werte herauszufinden. Handelt es sich um einen komplizierteren Attributtyp, wie z.B. ein Cluster, zeigt das Hilfenfenster eine hierarchische Beschreibung der Datenstruktur an. Die folgende Abbildung zeigt einen Attributknoten und die zugehörigen Hilfeinformationen, die beim Verschieben des Cursors über die verschiedenen Attributnamen angezeigt werden.



Online-Referenz

Das Hilfenfenster zeigt die am häufigsten gebrauchten Hilfeinformationen in verkürzter Fassung an. Ausführlichere Online-Informationen erhalten Sie, wenn Sie **Hilfe>Online-Referenz** wählen. Bei den meisten Objekten können Sie **Online-Referenz** aus dem Popup-Menü des Objekts wählen. Zugriff auf diese Informationen erhalten Sie auch durch Drücken des Fragezeichens (siehe Abbildung links), das sich am unteren Rand des Hilfenfensters befindet.



Weitere Informationen über das Erstellen eigener Online-Hilfedateien finden Sie in Kapitel 5, *Drucken und Dokumentieren von VIs*.

Bearbeiten von VIs

Dieses Kapitel erläutert die grundlegenden Funktionen beim Erstellen und Anwenden von VIs und enthält darüber hinaus Informationen über Paletten und Menüs. Es erläutert ferner grundlegende Aufgaben wie z.B. das Erstellen eines Objekts, Wechseln von Werkzeugen und Öffnen, Ausführen und Speichern von VIs.

G-Umgebung

Zum Erstellen von VIs werden Paletten und Menüs verwendet. Sie können Objekte aus Paletten wählen, um sie auf das Frontpanel oder Blockdiagramm zu plazieren. Diese Objekte können mit Hilfe von Werkzeugen oder Menüs verschoben oder geändert werden.

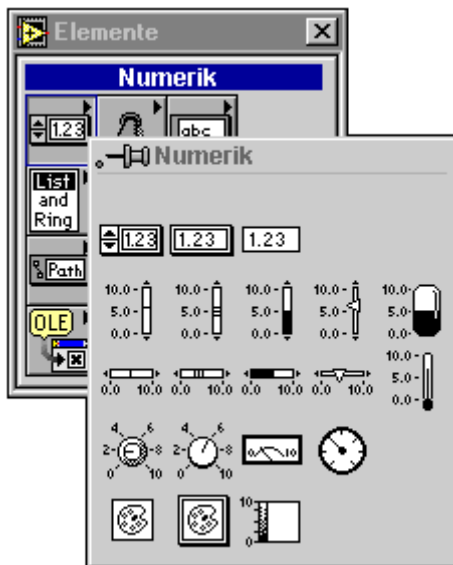


Hinweis

BridgeVIEW-Benutzer müssen sicherstellen, daß ihre Paletten auf Basis umgestellt sind. Wählen Sie hierzu Bearbeiten»Palettensatz auswählen. Wählen Sie anschließend die Palette Basis aus dem Menü der Palettensätze.

Sie erstellen Objekte am Frontpanel und im Blockdiagramm, indem Sie sie aus den floatenden Paletten **Elemente** und **Funktionen** wählen. Wenn Sie beispielsweise am Frontpanel einen Drehknopf erstellen wollen,

wählen Sie diesen, wie in der nachstehende Abbildung gezeigt, aus der Palette **Numerik** der Palette **Elemente**.



Wenn Sie den Auswahlpfeil über ein Objekt auf der Palette verschieben, erscheint der Name des Objekts oben auf der Palette. In der obigen Abbildung ist der Drehknopf gewählt worden. Wenn Sie zu diesem Zeitpunkt die Maustaste klicken und die Maus auf das Frontpanel verschieben, erscheint an dem Punkt, an dem Sie die Maustaste loslassen, ein Drehknopf. Durch Klicken und Ziehen des Cursors beim Plazieren können Sie die Größe eines Objekts gleich beim Erstellen anpassen.



Hinweis

Wenn Sie mehrere Funktionen aus derselben Palette benötigen, können Sie diese geöffnet lassen. Um eine Palette geöffnet zu lassen, wählen Sie die Stecknadel in der linken oberen Ecke der Palette. Wenn Sie die Palette festgestellt haben, besitzt sie eine Titelleiste, damit Sie diese leicht bewegen können. Wenn Sie die G-Entwicklungsumgebung verlassen, werden die Positionen der Paletten gespeichert und in der nächsten Arbeitssitzung wieder an derselben Stelle geöffnet.

Beim Aufrufen des Popup-Menüs im freien Raum eines Frontpanel- oder Blockdiagramm-Fensters erscheint eine temporäre Kopie der Paletten Elemente und Funktionen. Wenn Sie an einem kleineren Monitor arbeiten, können Sie die floatenden Paletten Elemente und Funktionen schließen und statt dessen Objekte mit diesen Popup-Paletten erstellen.

Durch Wählen eines der betreffenden Menüpunkts für nebeneinander oder untereinander im Menü Fenster können Sie Frontpanel und Blockdiagramm nebeneinander oder untereinander anordnen.

Wenn Sie Objekte für ein Frontpanel erstellen, erscheinen diese mit einem leeren Label, so daß Sie den Namen des neuen Objekts eingeben können. Wenn Sie dem Objekt einen Namen geben wollen, geben Sie den Namen ein. Schließen Sie die Eingabe durch Drücken der <Eingabetaste> auf dem Ziffernblock ab. Verfügen Sie nicht über einen Ziffernblock, klicken Sie in der Werkzeugpalette auf die Taste **Enter** oder an einer beliebigen Stelle außerhalb des Labels.



Hinweis

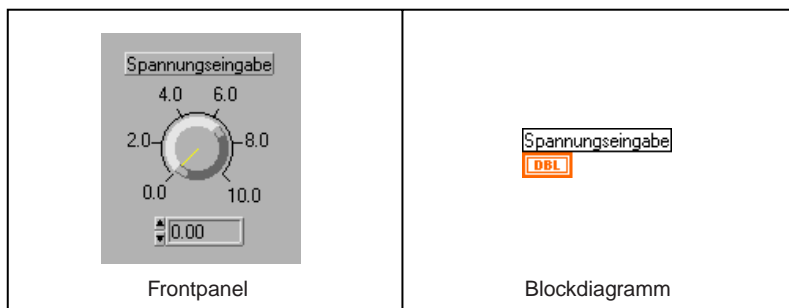
Wird beim ersten Erstellen eines Bedienelements kein Text in das Label eingegeben, verschwindet dieses, wenn an einer beliebigen anderen Stelle die Maustaste geklickt wird. Das Label kann über das Popup-Menü des Bedienelements mit Anzeigen»Label wieder angezeigt werden.

Die folgende Abbildung zeigt ein Beispiel für das Ergebnis der Betätigung des Popup-Menüs auf dem Bedienelement und Wählen der Option **Anzeigen»Label**.



Wenn Sie auf dem Frontpanel ein Objekt erstellen, wird im Blockdiagramm ein entsprechendes Terminal erstellt. Über dieses Terminal werden Daten von einem Bedienelement abgelesen oder an ein Anzeigeelement gesendet.

Wenn Sie **Fenster»Diagramm anzeigen** wählen, können Sie das Blockdiagramm für das Frontpanel sehen. Das Blockdiagramm beinhaltet Terminals für alle Bedien- und Anzeigeelemente des Frontpanels.



Werkzeugpalette

Ein *Werkzeug* ist ein bestimmter Arbeitsmodus des Mauscurtors. Mit Hilfe von Tools werden die Operationen definiert, die bei anschließend auf dem Panel oder im Diagramm erfolgenden Mausklicks ablaufen.

In der folgenden Abbildung sind viele der auf der floatenden Palette **Werkzeuge** enthaltenen Werkzeuge zu sehen. Sie können diese Palette beliebig verschieben oder durch Klicken auf das Feld "Schließen" (das kleine Feld mit dem "x" in der rechten oberen Ecke von Paletten) vorübergehend schließen. Auf eine ausgeblendete Palette kann durch Wählen von **Fenster»Werkzeugpalette anzeigen** wieder zugegriffen werden. Sie können durch Klicken und gleichzeitiges Drücken von <Strg-Umschalttaste> (**Windows**), <Befehlstaste-Umschalttaste> (**Macintosh**), <Meta-Umschalttaste> (**Sun**) oder <Alt-Umschalttaste> (**HP-UX**) an der Position des Cursors auch eine temporäre Version der Palette **Werkzeuge** aufrufen.



Die Werkzeuge der Palette **Werkzeuge** und ihre Funktionsweise sind im folgenden kurz beschrieben:



Bedienwerkzeug — Ändert den Wert eines Bedienelements oder wählt den Text in einem Bedienelement.



Positionierwerkzeug — Positioniert und wählt Objekte und paßt ihre Größe an.



Beschriftungswerkzeug — Bearbeitet Text und erstellt leere Labels.



Verbindungswerkzeug — Verbindet Objekte im Blockdiagramm miteinander.



Popup-Menü-Werkzeug für Objekte — Öffnet das Popup-Menü eines Objekts.



Fensterrollwerkzeug — Rollt das Fenster ohne Benutzung der Rollbalken weiter.



Breakpointwerkzeug — Setzt Breakpoints in VIs, Funktionen, Schleifen, Sequenzen und Cases.



Probewerkzeug — Erstellt Proben in Verbindungen.



Farbenkopierwerkzeug — Kopiert Farben zum Einfügen mit dem Farbenwerkzeug.



Farbenwerkzeug — Legt Vorder- und Hintergrundfarben fest.

Im Bearbeitungsmodus ist es möglich, mit einem der folgenden Verfahren von einem Werkzeug zu einem anderen zu wechseln:

- Klicken Sie auf der Palette **Werkzeuge** auf das gewünschte Werkzeug.
- Gehen Sie die am meisten gebrauchten Werkzeuge nacheinander mit der <Tab>-Taste durch.
- Drücken Sie die Leertaste, um von Bedienwerkzeug auf Positionierwerkzeug umzuschalten, wenn das Frontpanel aktiv ist, und um von Verbindungswerkzeug auf Positionierwerkzeug umzuschalten, wenn das Blockdiagramm aktiv ist.

Verwendung von Menüs

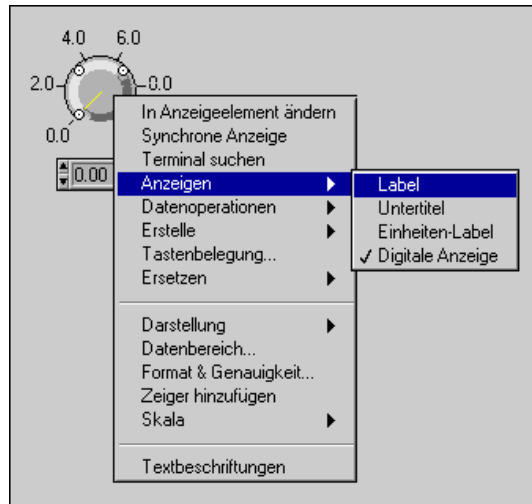
Die *Menüleiste* am oberen Rand eines VI-Fensters (oder am oberen Rand des Macintosh-Bildschirms) enthält mehrere *Pulldown-Menüs*. Sie können von einem Menübalken aus auf Pulldown-Menüs zugreifen, indem Sie auf einen Menüpunkt klicken. Die Pulldown-Menüs sind in der Regel allgemeiner Natur und enthalten Menüpunkte, die auch in anderen Anwendungen gebräuchlich sind, wie z.B. **Öffnen**, **Speichern**, **Kopieren** und **Einfügen**, sowie viele andere, die Editor-spezifisch sind. Manche Menüs verfügen auch über Tastenkombinationen.

Das am häufigsten gebrauchte Menü ist das *Popup-Menü* von Objekten. Beinahe jedes G-Objekt, einschließlich des Freiraums von Frontpanels und Blockdiagrammen, besitzt ein kontextsensitives Popup-Menü mit Menüpunkten und Befehlen. Wenn dies möglich ist, empfiehlt es sich, einen Befehl oder einen Menüpunkt aus dem Popup-Menü eines Objekts zu wählen.

Popup-Menüs

Wie Sie im vorherigen Abschnitt erfahren haben, sind alle G-Objekte mit Popup-Menüs verknüpft. Sie öffnen ein Popup-Menü, indem Sie auf das Objekt klicken und dann die rechte Maustaste drücken (beim Macintosh halten Sie die <Befehlstaste> gedrückt, während Sie auf das Objekt

klicken). Mit den Menüpunkten des dadurch angezeigten Menüs läßt sich das Aussehen und die Verhaltensweise des Objekts anpassen.



Bearbeiten von VIs

Wenn Sie sich mit den Bearbeitungsverfahren vertraut gemacht haben, finden Sie im Abschnitt *Frontpanel-Objekte* (Kapitel 8 bis 16) weitere Informationen zum Erstellen von Frontpanels, und im Abschnitt *Blockdiagramm-Programmierung* dieses Handbuchs (Kapitel 17 bis 23) finden Sie Hilfe zum Erstellen von Blockdiagrammen.

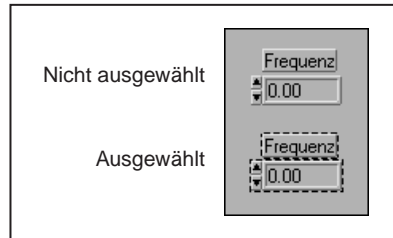
Auswählen von Objekten

Bei vielen Bearbeitungsoperationen, wie z.B. beim Verschieben, Kopieren und Löschen, muß ein Objekt *ausgewählt* werden. Verwenden Sie zum Auswählen eines Objekts unbedingt das Positionierwerkzeug, und klicken

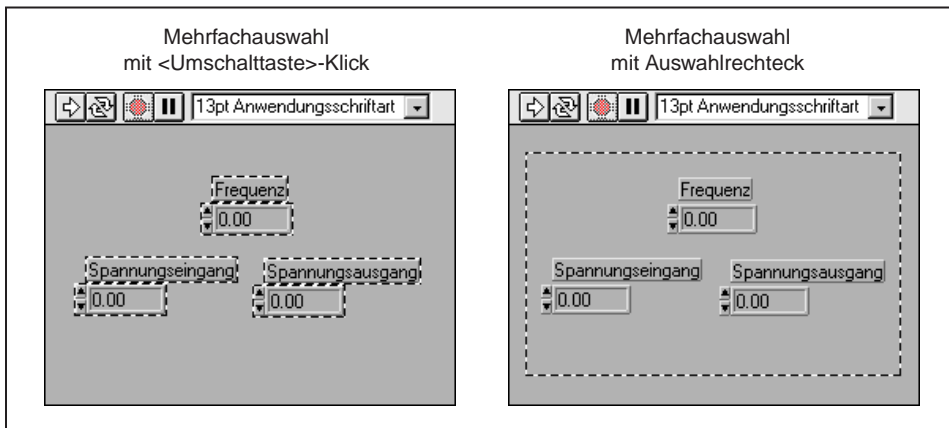


Positionierwerkzeug

Sie die Maustaste, wenn sich die Pfeilspitze auf dem Objekt befindet. Sobald ein Objekt ausgewählt ist, wird es von einem dynamischen, gestrichelten Umriß umgeben, einem sogenannten *Auswahlrechteck*.



Um mehrere Objekte gleichzeitig auszuwählen, klicken Sie auf jedes weitere Objekt, während Sie die <Umschalttaste> gedrückt halten. Sie können die Auswahl eines Objekts auf dieselbe Weise rückgängig machen. Eine andere Methode der Auswahl mehrerer Objekte besteht darin, ein Auswahlrechteck um die Objekte zu ziehen, wie dies in der folgenden Abbildung rechts illustriert wird.



Ungefüllte Objekte müssen üblicherweise vollständig von dem Rechteck eingeschlossen werden, um ausgewählt zu werden. Die meisten Objekte werden bereits ausgewählt, wenn ein Teil davon im Auswahlrechteck liegt. Durch Drücken und Halten der <Umschalttaste> und gleichzeitigem Ziehen des Auswahlrechtecks können zusätzliche Objekte ausgewählt oder ein Objekt im Rechteck von der Auswahl ausgenommen werden.

Durch Klicken auf ein nicht gewähltes Objekt oder Klicken im freien Raum wird die gesamte gegenwärtig bestehende Auswahl aufgehoben. Ein Objekt eines Frontpanels und ein Objekt eines Blockdiagramms können

nicht gleichzeitig ausgewählt werden. Sie können jedoch mehrere Objekte desselben Frontpanels oder Blockdiagramms gleichzeitig ausgewählt werden.

Ziehen und Ablegen von VIs, Bildern und Text

Sie können VIs vom Dateisystem in ein Blockdiagramm ziehen, um einen Aufruf eines SubVIs zu diesem VI zu erstellen. Diese Funktion ist auch bei benutzerspezifischen Bedienelementen, Typendefinitionen und globalen Variablen anwendbar. Außerdem können Sie Text und Bilder von anderen Anwendungen (durch Ziehen) platzieren, um sie auf Frontpanels und in Blockdiagramme zu kopieren.

In Windows 3.1 ist die Unterstützung externer Ziehoperationen auf das Ziehen von VIs und Bedienelementen vom Dateimanager aus beschränkt. In Windows 95/NT, wo 32-Bit-Unterstützung für Verknüpfung und Einbettung von Objekten (object linking and embedding, OLE) gegeben ist, können Sie zusätzlich zum Ziehen und Ablegen vom Datei-Manager/ Explorer aus Text und Bilder von OLE-fähigen Anwendungen durch Ziehen und Ablegen platzieren.

Beim Macintosh muß das System über den Drag Manager verfügen, damit externes Ziehen und Ablegen ermöglicht wird. Der Drag Manager ist ab System 7.5 fester Bestandteil des Betriebssystems, und er wird für System 7.0 bis System 7.5 von Apple als Erweiterung angeboten.

Intern sind folgende Ziehen-und-Ablegen-Operationen möglich.

- Bedien- und Anzeigeelemente von Frontpanels in Blockdiagramme ziehen, um Konstanten für Blockdiagramme zu erstellen, und umgekehrt.
- Ein SubVI auf ein Bedienelement oder eine Konstante eines Pfades ziehen, um den vollständigen Pfadnamen des VIs innerhalb des Bedienelements oder der Konstante abzulegen.

Extern (von anderen Anwendungen aus) sind die folgenden Ziehen-und-Ablegen-Operationen möglich.

- Eine Datei in ein Bedienelement oder eine Konstante eines Pfades ziehen, um dessen vollständigen Pfadnamen abzulegen.
- Eine Datenprotokolldatei in eine Refnum einer Datenprotokolldatei ziehen, um ein Cluster zu erstellen, das die Datenstruktur der Datenprotokolldatei enthält.
- Eine Graphikdatei in einen Bild-Ring, ein Frontpanel oder Blockdiagramm ziehen, um das darin enthaltene Bild abzulegen.

- Eine VI-Datei in das Blockdiagramm eines VIs ziehen, um sie als SubVI abzulegen.
- Ein benutzerspezifisches Bedienelement auf das Frontpanel ziehen, um das in der betreffenden Datei gespeicherte benutzerspezifische Bedienelement abzulegen.
- **(Windows 95/NT)** Text von einer OLE-Quelle in ein Bedienelement oder eine Konstante einer Zeichenfolge, ein Frontpanel, Blockdiagramm oder eine Beschriftung ziehen und dort abzulegen.
- **(Windows 95/NT)** Ein Bild von einer OLE-Quelle in einen Bild-Ring, ein Frontpanel oder Blockdiagramm ziehen und abzulegen.
- **(Macintosh)** Ausgeschnittenen oder in einer beliebigen Anwendung markierten Text in ein Bedienelement oder eine Konstante einer Zeichenfolge, ein Frontpanel, Blockdiagramm oder eine Beschriftung ziehen und abzulegen.
- **(Macintosh)** Ein ausgeschnittenes oder in einer anderen Anwendung ausgewähltes Bild in einen Bild-Ring, ein Frontpanel oder Blockdiagramm ziehen und abzulegen.



Hinweis

Kann ein Objekt erfolgreich abgelegt werden, wird das Ziel markiert.

Weitere Informationen über die Funktionen Ziehen und Ablegen (Plazieren) finden Sie im Windows- oder Macintosh-Systemhandbuch.

Positionieren von Objekten



Sie können ein Objekt plazieren, indem Sie mit dem **Positionierwerkzeug** darauf klicken und es dann an die gewünschte Position verschieben.

Wenn Sie die <Umschalttaste> gedrückt halten und ein Objekt ziehen, wird die Verschiebungsrichtung, je nachdem, in welche Richtung das Objekt zuerst verschoben wird, auf horizontale oder vertikale Richtung beschränkt.

Sie können gewählte Objekte in kleinen, präzisen Inkrementen verschieben, indem Sie auf der Tastatur für jedes gewünschte Pixel einmal die betreffende Pfeiltaste drücken. Halten Sie die Pfeiltaste gedrückt, um den Vorgang zu wiederholen. Wenn Sie die <Umschalttaste> und eine Pfeiltaste gleichzeitig gedrückt halten, können Sie das Objekt zügiger verschieben.

Wenn Sie Ihre Meinung zum Verschieben eines Objekts während des Ziehens ändern, setzen Sie den Ziehvorgang so weit fort, bis der Cursor sich nicht mehr in einem der offenen Fenster befindet und der gestrichelte

Umriß verschwindet, und lassen Sie dann die Maustaste los. Dadurch wird die Verschiebung abgebrochen, und das Objekt bleibt an seiner ursprünglichen Position. Ist Ihr Bildschirm unübersichtlich voll, empfiehlt sich der Menübalken als bequemer und sicherer Ort, um die Maustaste loszulassen, falls Sie einen Verschiebungsvorgang abbrechen möchten.



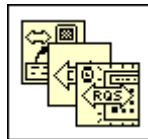
Hinweis

Das Positionierwerkzeug dient auch zum Vergrößern von Objekten. Achten Sie beim Klicken der Maustaste darauf, daß dies nicht versehentlich geschieht. Klicken Sie zum Verschieben eines Objekts in dessen Mitte und zum Vergrößern an dessen Ecken.

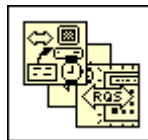
Neu anordnen



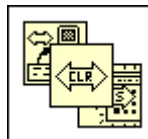
Sie können Objekte auch über andere Objekte plazieren. Unter **Neuordnen** ganz rechts in der Werkzeugleiste, die sich quer über das Frontpanel erstreckt, gibt es mehrere Befehle, mit denen Objekte relativ zueinander verschoben werden können. Nehmen wir z.B. an, daß Sie drei Objekte übereinander gestapelt haben, wobei Objekt 1 das unterste und Objekt 3 das oberste Objekt im Stapel ist.



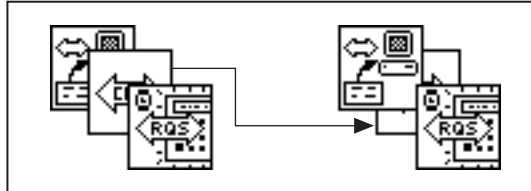
Ganz nach vorne verschiebt das gewählte Objekt in den Vordergrund des Stapels. Wenn Objekt 1 gewählt wird, so wird es an oberste Stelle im Stapel verschoben, gefolgt von Objekt 3 und schließlich Objekt 2.



Wird als nächstes Objekt 2 gewählt, ändert **Ganz nach vorne** die Reihenfolge dahingehend, daß Objekt 2 im Vordergrund abgebildet wird, darunter Objekt 1 und als unterstes schließlich Objekt 3.



Eins nach vorne verschiebt das gewählte Objekt im Stapel um eine Stelle höher. Geht man von der ursprünglichen Reihenfolge aus, bei der sich Objekt 1 an unterster Stelle und Objekt 3 an oberster Stelle im Stapel befand, wird mit dieser Option bei gewähltem Objekt 1 Objekt 2 an unterste Stelle, Objekt 3 an oberste Stelle und Objekt 1 in die Mitte verschoben.

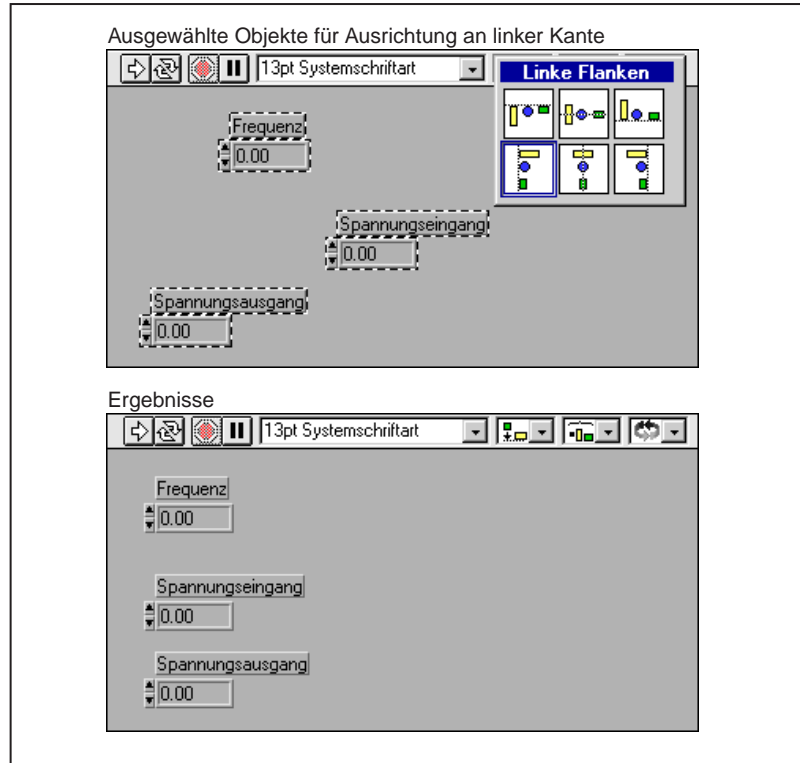


Ganz nach hinten und **Eins nach hinten** funktionieren ähnlich wie **Ganz nach vorne** und **Eins nach vorne**, mit dem Unterschied, daß hier die Objekte nicht in den Vordergrund, sondern den Hintergrund des Stapels verschoben werden.

Ausrichten von Objekten



Wählen Sie die Objekte aus, die Sie ausrichten wollen, und wählen Sie anschließend die Achse, an der sie ausgerichtet werden sollen, aus dem Ring **Objekte ausrichten** (siehe Abbildung links). Die folgende Abbildung zeigt Ihnen, wie auszurichtende Objekte ausgewählt werden, sowie die Ergebnisse der Auswahl.



Sie können ein Objekt mit seiner linken, mittleren oder rechten Kante vertikal und analog mit der oberen, mittleren oder unteren Kante horizontal ausrichten. Ihre Auswahl wird zur aktuellen Ausrichtungsoption, die in der Palette durch eine hervorgehobene Umrandung angezeigt wird. Auf der Macintosh-Plattform kann die Ausrichtungsauswahl unter Umgehung des Menüs durch <Befehlstaste-A> wiederholt werden.

Einteilen von Objekten



Um Objekte gleichmäßig anzuordnen bzw. einzuteilen, wählen Sie die Objekte aus, die Sie einteilen wollen, und wählen Sie die gewünschte Einteilungsmethode aus dem Ring **Objekte einteilen**. Zusätzlich zur Einteilung von ausgewählten Objekten durch die Festlegung gleicher Abstände zwischen ihren Kanten oder Mittelachsen ermöglichen vier Menüpunkte auf der rechten Seite des Rings Ihnen, horizontal oder vertikal Lücken zwischen den Objekten hinzuzufügen oder zu löschen. Auf der Macintosh-Plattform wird die Einteilung einer beliebigen Auswahl durch Drücken der <Befehlstaste-D> wiederholt.

Duplizieren von Objekten

Es gibt drei grundlegende Methoden zum Duplizieren eines Objekts — durch Kopieren und Einfügen, durch Klonen und durch Ziehen und Ablegen.

Um Objekte innerhalb eines VIs oder zwischen VIs zu kopieren und einzufügen, wählen Sie das Objekt mit dem Positionierwerkzeug aus, und wählen Sie dann **Bearbeiten»Ausschneiden** oder **Bearbeiten»Kopieren**. Klicken Sie dann in dem Bereich, in dem Sie das Duplikat plazieren wollen, und wählen Sie **Bearbeiten»Einfügen**. Sie können mehrere Objekte gleichzeitig duplizieren, indem Sie ein Auswahlrechteck um die Objekte ziehen, bevor Sie sie duplizieren. Sie können auch Text oder Bilder aus anderen Anwendungen kopieren und in Ihr Frontpanel oder Blockdiagramm einfügen.

Um ein Objekt zu klonen, klicken Sie mit dem Positionierwerkzeug darauf, während Sie die <Strg-Taste> (**Windows**), <Optionstaste> (**Macintosh**), <Metataste> (**Sun**) oder <Alt-Taste> (**HP-UX**) gedrückt halten, und ziehen Sie die Kopie an ihre neue Position. In UNIX können Sie Objekte auch klonen, indem Sie auf dem gewählten Objekt ohne weitere Zusatz Tasten die mittlere Maustaste klicken und das Objekt ziehen.

Wenn Sie Objekte, die Labels besitzen, klonen oder kopieren, bezeichnet die Software der G-Sprache Kopien mit demselben Namen wie das ursprüngliche Objekt plus dem Zusatz *Kopie* (oder *Kopie 1*, *Kopie 2* usw. für Kopien von Kopien).

Wenn Sie ein Bedienelement eines Frontpanels kopieren, wird im Diagramm ein neues Terminal erstellt. Beachten Sie, daß lokale Variable und Attributknoten des ursprünglichen Bedienelements nicht mitkopiert werden. Wenn Sie für eine lokale Variable **Bearbeiten»Kopieren** bzw. **Bearbeiten»Einfügen** wählen, wird das Objekt des Frontpanels mitkopiert. Wenn Sie eine lokale Variable klonen, wird ein neuer Bezug zum ursprünglichen Bedienelement erstellt. Weitere Informationen hierzu finden Sie in Kapitel 22, *Attributknoten*, und Kapitel 23, *Globale und lokale Variablen*, in diesem Handbuch.

Sie können Bedienelemente eines Frontpanels durch Kopieren, Klonen oder Ziehen und Ablegen in ein beliebiges Diagramm plazieren, um eine dementsprechende Konstante zu erstellen. In derselben Weise können Sie benutzerdefinierte Konstanten vom Blockdiagramm auf das Frontpanel ziehen, um Bedienelemente zu erstellen.

Löschen von Objekten

Um ein Objekt zu löschen, wählen Sie das Objekt und **Bearbeiten» Löschen** aus, oder drücken Sie die <Rücktaste> oder die <Löschtaste>. Terminals eines Blockdiagramms können nur durch Löschen der entsprechenden Bedien- oder Anzeigeelemente am Frontpanel gelöscht werden.

Zwar können die meisten Objekte gelöscht werden, Teile eines Bedien- oder Anzeigeelements, wie z.B. Beschriftungen oder digitale Anzeigen, können nicht gelöscht werden. Diese Komponenten lassen sich jedoch durch Deaktivieren des Menüpunkts **Anzeigen»Label** oder **Anzeigen» Digitale Anzeige** im Popup-Menü des Objekts verbergen.

Wenn Sie eine Struktur wie z.B. eine While-Schleife löschen, wird deren Inhalt mitgelöscht. Wollen Sie lediglich die Struktur löschen und deren Inhalt erhalten, wählen Sie auf der Kante der Struktur das Popup-Menü, und klicken Sie auf **While-Schleife entfernen** (oder den Namen der jeweiligen Struktur). Hiermit wird zwar die Struktur, nicht aber deren Inhalt gelöscht und Verbindungen, die über die Struktur hinweg bestanden, werden automatisch wiederhergestellt.

Beschriften von Objekten

Labels (Beschriftungen) sind Textblocks mit Anmerkungen zu Komponenten von Frontpanels und Blockdiagrammen. Es gibt zwei Arten von Labels — *gebundene* und *freie* Labels. Gebundene Labels gehören zu einem bestimmten Objekt, sie werden mit dem Objekt zusammen verschoben und bezeichnen nur dieses spezielle Objekt. Diese Labels können verborgen werden, sie können jedoch nicht unabhängig von ihren "Eigentümern" kopiert oder gelöscht werden. Freie Labels sind nicht mit einem Objekt verbunden und können daher beliebig erstellt, verschoben oder gelöscht werden. Verwenden Sie derartige Labels zur Beschriftung Ihrer Frontpanels und Blockdiagramme. Zum Erstellen freier Labels oder zur Bearbeitung beider Arten von Labels verwenden Sie das Beschriftungswerkzeug, dessen Cursor der vertikale I-Cursor und der gefüllte Blockcursor sind.



Beschriftungen von Frontpanels

Mit jedem Objekt eines Frontpanels ist ein Name verknüpft, der es von anderen Objekten unterscheidet. Der Name wird außerdem am Terminal, bei lokalen Variablen und Attributknoten des Objekts verwendet

Handelt es sich bei dem Objekt um ein Bedienelement, gibt der Text des Labels die Verbindung des Bedienelements im Blockdiagramm an. Handelt es sich bei dem Objekt um ein Anzeigeelement und ist es mit dem Anschlußfeld verbunden, gibt das Label Verbindungen von dessen Anschlußfeld-Terminal in Blockdiagrammen der Aufrufenden der VIs an. Dies bedeutet, daß LabVIEW das VI und möglicherweise sogar auch die Aufrufenden des VIs neu kompilieren muß, wenn sich der Text des Namen-Labels ändert.

Objekte von Frontpanels können ebenfalls Titel-Labels haben. Der Titel hat keinen Einfluß auf den Namen des Objekts und Sie können ihn als deskriptivieren Objektnamen gebrauchen. Sie können den Titel mit Hilfe von Attributknoten programmatisch anzeigen, verbergen und ändern. Bei der Änderung des Titels müssen weder das VI noch die Aufrufenden des VIs neu kompiliert werden.

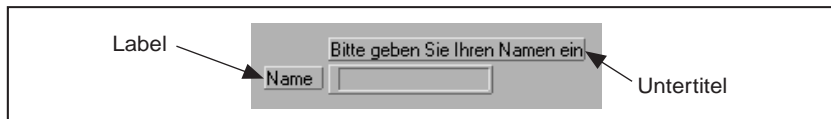


Abbildung 2-1. Dialogfeld "Frontpanel-Label"

Wenn Sie ein Objekt auf ein Frontpanel plazieren, wird das Label angezeigt. Um beim Bearbeiten des Objekts auf den Titel zuzugreifen, aktivieren Sie das Popup-Menü des Objekts, und wählen Sie **Anzeigen» Untertitel** (siehe folgende Abbildung).

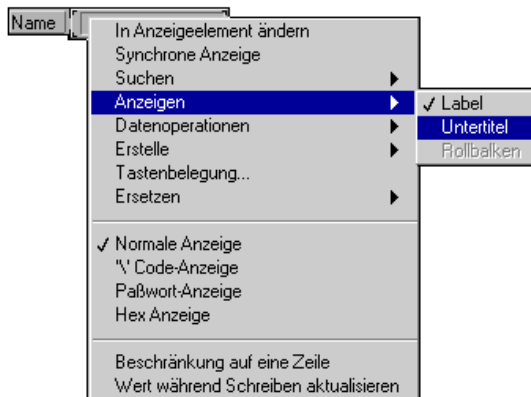


Abbildung 2-2. Popup-Menü mit ausgewähltem Label

Wenn Sie den Titel programmatisch anzeigen, verbergen oder ändern wollen, verwenden Sie hierzu die in der folgenden Abbildung gezeigten Attribute.

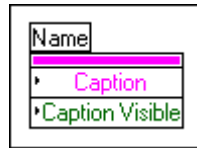


Abbildung 2-3. Objekt im Blockdiagramm mit sichtbarem Titel

Wenn Sie das Hilfefenster zum Objekt anzeigen (durch Wählen von **Hilfe» Hilfe anzeigen**), erscheint der Titel des Frontpanel-Objekts in einem Kasten und darunter dessen Label in rechteckigen Klammern (siehe folgende Abbildung).

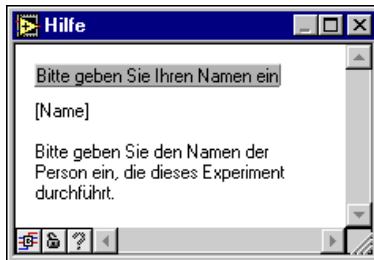
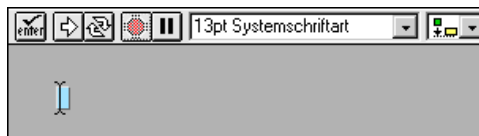


Abbildung 2-4. Kontextsensitives Hilfe-Dialogfeld

Freie Labels

Wählen Sie zum Erstellen eines freien Labels das Beschriftungswerkzeug aus der Palette **Werkzeuge**, und klicken Sie in einen offenen Bereich.



Ein kleines Feld erscheint, in dem sich am linken Rand ein Textcursor befindet, so daß Sie gleich Text eingeben können. Geben Sie den Text ein, der im Label erscheinen soll. Schließen Sie die Bearbeitungsoperation

durch Klicken auf die Schaltfläche **Enter** der Werkzeugpalette oder das Drücken der Eingabetaste des Ziffernblocks ab.



Sie können die Texteingabe auch durch Klicken an einer beliebigen Stelle außerhalb des Labels abschließen. Wenn Sie keinen Text in das Label eingeben, wird das Label gelöscht, sobald Sie an einer anderen Stelle die Maustaste klicken.



Hinweis

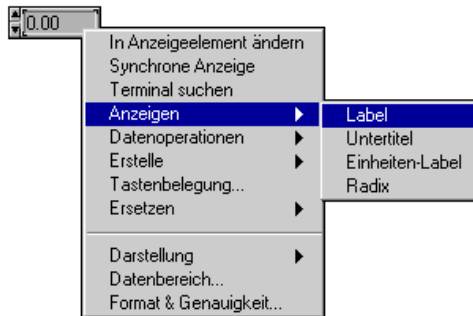
Wenn Sie ein Label oder ein beliebiges anderes Objekt auf ein Bedien- oder Anzeigeelement platzieren (und dies damit teilweise verdecken), beeinträchtigt dies die Aktualisierung der Bildschirmanzeige und kann dazu führen, daß das Bedien- oder Anzeigeelement flackert. Um dieses Problem zu vermeiden, achten Sie darauf, daß auf dem Frontpanel kein Objekt kein Label oder anderes Objekt überlappt.

Sie können den Text eines Labels kopieren, indem Sie ihn mit dem Beschriftungswerkzeug auswählen. Doppelklicken Sie zur Auswahl einzelner Wörter mit dem Beschriftungswerkzeug auf den Text. Klicken Sie dreimal auf den Text, um das ganze Label hervorzuheben. Wählen Sie zum Kopieren des Texts in die Zwischenablage **Bearbeiten»Kopieren**, sobald der Text ausgewählt ist. Jetzt können Sie den Text eines zweiten Labels hervorheben. Wählen Sie anschließend **Bearbeiten»Einfügen**, um den hervorgehobenen Text des zweiten Labels mit dem Text in der Zwischenablage zu ersetzen. Um mit dem Text in der Zwischenablage ein neues Label zu erstellen, klicken Sie mit dem Beschriftungswerkzeug an der Stelle auf dem Bildschirm, an der das neue Label positioniert werden soll. Wählen Sie anschließend **Bearbeiten»Einfügen**.

Wenn Sie auf dem Frontpanel ein Bedien- oder Anzeigeelement erstellen, wird gleichzeitig ein leeres, gebundenes Label angezeigt, das darauf wartet, daß Sie den Namen des neuen Bedien- oder Anzeigeelements eingeben. Das Label verschwindet, wenn Sie keinen Text eingeben, bevor Sie an einer anderen Stelle die Maustaste klicken. Sie können das Label wieder anzeigen, indem Sie das Popup-Menü des Bedien- oder Anzeigeelements aktivieren und **Anzeigen»Label** wählen.

Strukturen und Funktionen besitzen ein Standard-Label, das verborgen ist, bis Sie es anzeigen.

Um ein verborgenes Label anzuzeigen, aktivieren Sie das Popup-Menü des Objekts, und wählen Sie aus dem Popup-Menü **Anzeigen»Label** (siehe folgende Abbildung).



Das Label wird angezeigt und ist bereit für die Texteingabe. Das Label verschwindet, wenn Sie keinen Text eingeben, bevor Sie außerhalb des Labels die Maustaste klicken.



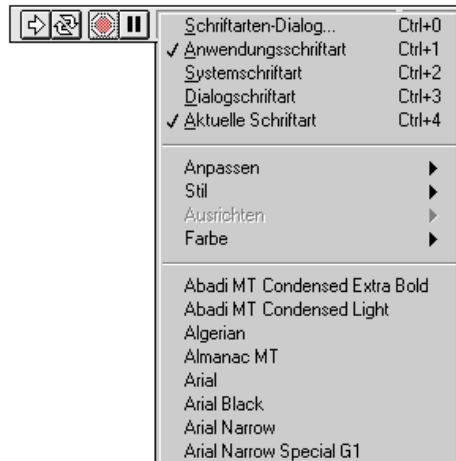
Hinweis

SubVIs weisen im Blockdiagramm keine Labels auf, die Sie bearbeiten können. Das Label eines SubVIs zeigt stets dessen Namen an. Labels von Funktionen hingegen können bearbeitet werden, um den Gebrauch der Funktion im Blockdiagramm widerzuspiegeln. Sie können beispielsweise das Label einer Additionsfunktion verwenden, um zu dokumentieren, welche Beträge addiert werden oder weshalb sie an diesem Punkt des Blockdiagramms addiert werden. Weitere Informationen hierzu finden Sie im Hinweis des Abschnitts “Erstellen von VI-Beschreibungen” dieses Kapitels.

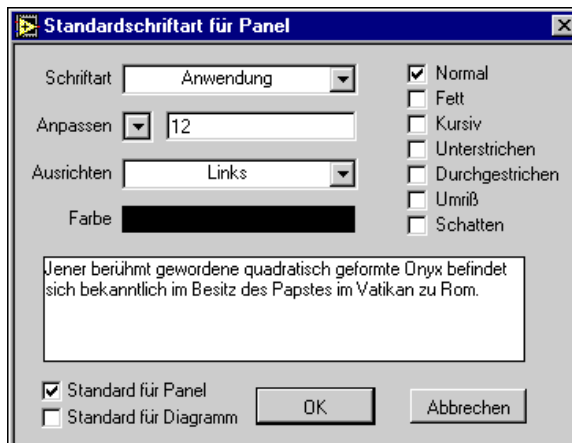
Texteigenschaften

Sie können Textattribute mit Hilfe der Menüpunkte aus dem Ring **Schriftart** der Werkzeugpalette ändern (siehe Abbildung unten). Wenn Sie Objekte oder Text auswählen und die Auswahl aus diesem Ring vornehmen, gelten die Änderungen für alle ausgewählten Objekte. Wird nichts ausgewählt, gelten die Änderungen für die Standardschriftart, d.h. neue Labels verwenden die neue Standardschriftart. Das Ändern der Standardschriftart wirkt sich nicht auf die Schriftart bestehender Labels

aus; es wirkt sich nur auf Labels aus, die von diesem Zeitpunkt an erstellt werden.



Wenn Sie **Dialogsschriftart...** aus dem Schriftarttring wählen, während ein Frontpanel aktiv ist, erscheint das in der folgenden Abbildung gezeigte Dialogfeld. Ist statt dessen ein Blockdiagramm aktiv, unterscheidet sich das Dialogfeld lediglich darin, daß statt des Kontrollkästchens **Standardwerte für Panel** das Kontrollkästchen **Standarddiagramm** am unteren Rand des Dialogfeldes gewählt ist.



Ist entweder das Kontrollkästchen **Standardwerte für Panel** oder **Standarddiagramm** gewählt, werden die anderen Änderungen, die Sie in diesem Dialogfeld vornehmen, bei neuen Labels auf dem Frontpanel oder im Blockdiagramm verwendet.

In der obigen Abbildung erscheint das Wort **Anwendung** im Schriftarttring. Dieser Ring enthält außerdem die Menüpunkte **Systemschriftart**, **Dialog-Schriftart** und **Aktuelle Schriftart**. Der letzte Punkt im Ring, **Aktuelle Schriftart**, bezieht sich auf die zuletzt gewählte Schriftart.

Anwendungsschriftart, Systemschriftart und Dialogschriftart gelten für bestimmte Bereiche der Schnittstelle. Diese Schriftarten sind vordefiniert, damit sie plattformübergreifend am besten aufgelistet werden können. Wird ein VI, das eine dieser Schriftarten verwendet, auf eine andere Plattform übertragen, lassen sich die Schriftarten am besten zuordnen.

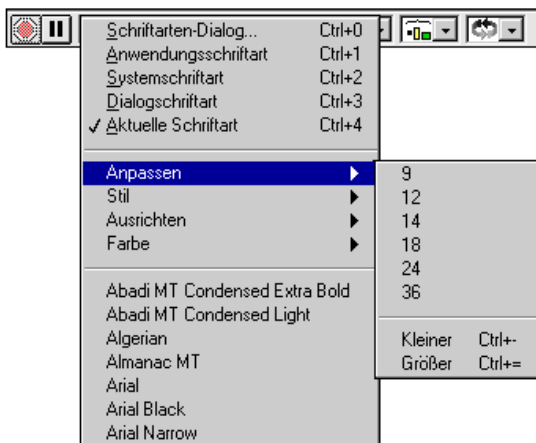
Die folgenden Schriftarten sind vordefiniert:

- Die Anwendungsschriftart ist die am meisten gebrauchte *Standardschriftart*. Sie kommt in der Palette **Elemente**, **Funktionen** und bei Text in neuen Bedienelementen zur Anwendung.
- Die Systemschriftart wird in Menüs verwendet.
- Die Dialogschriftart wird für Text in Dialogfeldern verwendet.

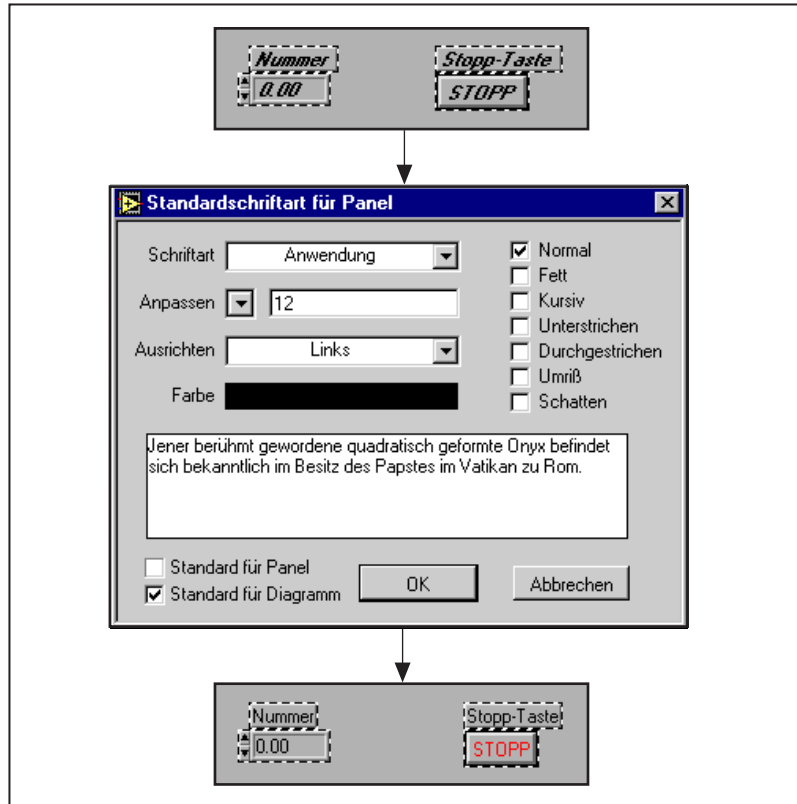
Wenn Sie die Kontrollkästchen **Standardwerte für Panel** und **Standarddiagramm** wählen, wird die gewählte Schriftart aktuelle Schriftart des Frontpanels, Blockdiagramms oder beider. Diese Kontrollkästchen geben Ihnen die Möglichkeit, für das Frontpanel und das Blockdiagramm verschiedene Schriftarten festzulegen. Sie können im Blockdiagramm beispielsweise eine kleine und auf dem Frontpanel eine große Schriftart verwenden.

Informationen zu Fragen der Portierbarkeit in Bezug auf die drei Schriftarttypen finden Sie im Abschnitt *Auflösungs- und Schriftartunterschiede* in Kapitel 29, *Portierbarkeit und Lokalisierungsthemen*. Informationen zum Ändern der Standardeinstellungen für die drei Kategorien von Schriftarten finden Sie im Abschnitt *Voreinstellungen für Schriftarten* in Kapitel 7, *Anpassen Ihrer Umgebung*.

Der Schriftartring besitzt außerdem die Menüpunkte **Größe**, **Stil**, **Ausrichten** und **Farbe** (siehe nachstehende Abbildung).



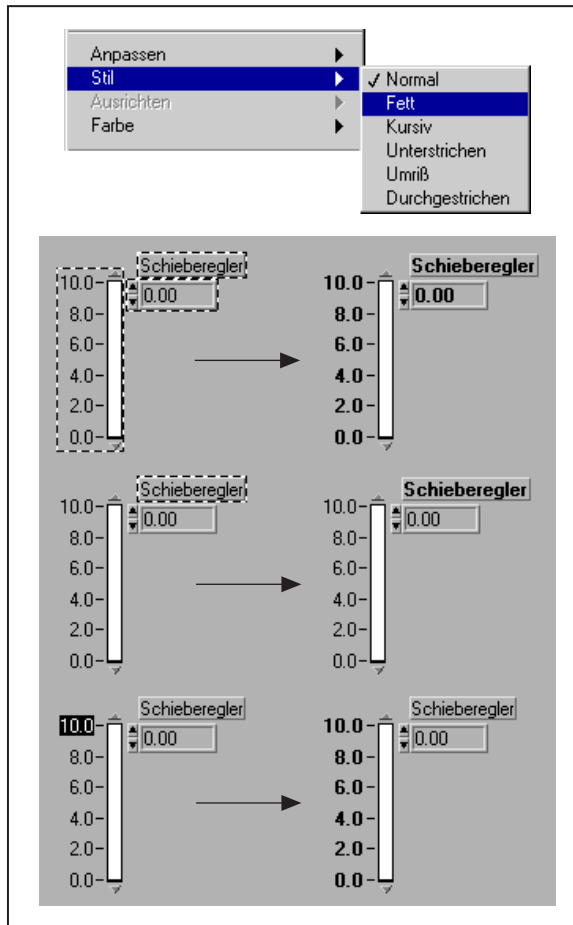
Die in den jeweiligen Untermenüs vorgenommene Schriftartauswahl gilt für Objekte, die Sie auswählen. Wenn Sie z.B. eine neue Schriftart wählen, während ein Drehknopf und ein Graph gewählt sind, ändert sich die Schriftart von deren Labels, Skalierung und digitalen Anzeige dementsprechend. Die folgende Abbildung zeigt ein numerisches und ein Boolesches Bedienelement, deren Schriftart von der aktuellen Schriftart in die Systemschriftart geändert wird.



Wenn Sie eine Änderung vornehmen, behält G so viele Schriftartattribute wie möglich bei. Wenn Sie die Schriftart mehrerer Objekte in "Courier" ändern, behalten die Objekte, sofern möglich, ihre Größe und ihren Stil bei. Gleichmaßen bewirkt die Änderung der Größe bei Auswahl mehrerer Textabschnitte nicht, daß alle Abschnitte dieselbe Schriftart annehmen. Diese Regeln gelten nicht, wenn Sie eine der vordefinierten Schriftarten oder die aktuelle Schriftart auswählen oder wenn Sie das Dialogfeld "Schriftart" verwenden, wobei die gewählten Objekte an die gewählte Schriftart und den gewählten Satz von Attributen angepaßt werden.

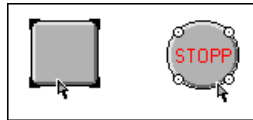
Beim Arbeiten mit Objekten wie z.B. Schieberegler, die mehrere Textelemente aufweisen, ist zu beachten, daß die Schriftartauswahl für Objekte oder gegenwärtig ausgewählten Text gilt. Ist z.B. der gesamte Schieberegler ausgewählt, wenn Sie die Option **Fett** wählen, werden die Skalierung, die digitale Anzeige und das Label dementsprechend angepaßt. Ist jedoch nur das Label ausgewählt, wenn Sie die Option **Fett** wählen, wird nur das Label fettgedruckt. Ist Text eines Achsenmarkers ausgewählt, wenn

Sie die Option **Fett** wählen, ändern sich alle Marker dementsprechend. Diese drei Arten von Änderungen werden in der nachstehenden Abbildung gezeigt.



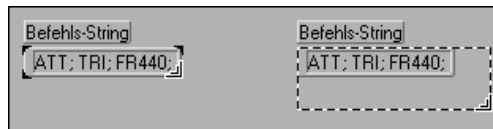
Anpassen der Größe von Objekten

Die Größe der meisten Objekte kann geändert werden. Wenn Sie das Positionierwerkzeug über ein Objekt bewegen, dessen Größe veränderbar ist, werden an den Ecken eines rechteckigen Objekts und an einem kreisförmigen Objekt Ziehpunkte angezeigt (siehe nachstehende Abbildung).



Die Größe eines kreisförmigen Objekts wird durch Ziehen der Ziehpunkte mit dem Cursor angepaßt, und die Größe eines rechteckigen Objekts wird analog durch Ziehen der Ziehpunkte an den Ecken angepaßt.

Wenn Sie das Werkzeug über einen Ziehpunkt bewegen, nimmt das Positionierwerkzeug die Gestalt des *Resizing-Cursors* an. Halten Sie die Maustaste gedrückt, und ziehen Sie den Cursor, bis der gestrichelte Rahmen die gewünschte Größe hat (siehe nachstehende Abbildung).



Um den Vorgang der Größenanpassung abzubrechen, ziehen Sie die Rahmenecke über den Fensterbereich hinaus, bis der gestrichelte Rahmen verschwindet. Lassen Sie dann die Maustaste los. Das Objekt behält in diesem Fall die ursprüngliche Größe bei.

Bei manchen Objekten kann die Größe nur horizontal oder vertikal angepaßt werden, oder sie behalten bei der Größenanpassung die Größenverhältnisse bei, wie z.B. bei einem Drehknopf. Der Resizing-Cursor sieht gleich aus, aber der gestrichelte Vergrößerungsumriß verschiebt sich nur in einer Richtung. Um die vertikale oder horizontale Größenzunahme eines beliebigen Objekts zu begrenzen oder die aktuellen Größenverhältnisse beizubehalten, halten Sie die `<Umschalttaste>` gedrückt, während Sie auf das Objekt klicken und dieses ziehen.

Labels

Die Größenanpassung von Labels erfolgt wie bei anderen Objekten auch mit Hilfe des Resizing-Cursors. Normalerweise erfolgt die Größenanpassung bei Labels *automatisch*, was bedeutet, daß das Feld automatisch an den Text, den Sie eingeben, angepaßt wird. Der Text von Labels bleibt auf einer Zeile, es sei denn, Sie geben einen Wagenrücklauf ein oder passen das Textfeld des Labels an. Wählen Sie **Größenanpassung an Text** aus dem Popup-Menü des Labels, um die automatische Größenanpassung wieder einzuschalten.

Arbeitsbereich hinzufügen

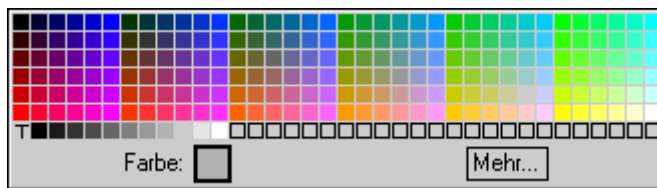
Um auf dem Frontpanel oder im Blockdiagramm zusätzlichen Arbeitsbereich zu schaffen, klicken und drücken Sie gleichzeitig zuerst die <Strg-Taste> (**Windows**), <Optionstaste> (**Macintosh**) <Meta-Taste> (**Sun**) bzw. <Alt-Taste> (**HP-UX**), und ziehen Sie dann mit dem Positionierwerkzeug einen Zaun. Auf der UNIX-Plattform können Sie ohne weitere Zusatztasten mit der mittleren Maustaste einen Zaun ziehen. Nach dem Klicken sehen Sie ein mit einer gestrichelten Linie gezeichnetes Rechteck, das den neuen Bereich definiert.

Farbige Kennzeichnung von Objekten

Die G-Entwicklungsumgebung erscheint je nach der Fähigkeit Ihres Monitors in Schwarzweiß, Grautonstufen oder Farbe. Sie können die Farbe vieler, jedoch nicht aller G-Objekte ändern. Terminals von Frontpanel-Objekten und Verbindungen verwenden für den Typ und die beförderten Daten Farbcodes in Blockdiagrammen, die daher nicht geändert werden können. Sie können im Schwarzweißmodus keine Farben ändern.



Um die Farbe eines Objekts oder den Hintergrund eines Fensters zu ändern, wählen Sie das Farbenwerkzeug aus der Werkzeugpalette (siehe Abbildung links), und klicken auf dem Objekt die rechte Maustaste. Die folgende Farbpalette erscheint.



Wenn Sie auf der Palette mit dem Cursor bei gedrückter Maustaste von Farbe zu Farbe gehen, wird das gewählte Objekt oder der Hintergrund mit

der Farbe gefüllt, die der Cursor gerade berührt. Sie erhalten dadurch eine Vorschau auf das Objekt mit der neuen Farbe. Um die Farbzuzuweisung abzubrechen, verschieben Sie den Cursor aus der Palette heraus, bevor Sie die Maustaste loslassen.



Wenn Sie das Feld mit dem Buchstaben “T” wählen, wird das Objekt transparent. Diese Funktion gibt Ihnen die Möglichkeit, Objekte zu staffeln. Sie können damit beispielsweise unsichtbare Bedienelemente auf Anzeigeelemente plazieren oder numerische Bedienelemente ohne den standardmäßigen dreidimensionalen Container erstellen. Transparenz beschreibt nur das Aussehen eines Objekts. Das Objekt spricht wie gewohnt auf die Bedienung der Maus und der Tasten an.

Manche Objekte besitzen sowohl einen Vordergrund als auch einen Hintergrund, der separat mit Farbe gefüllt werden kann. Die Vordergrundfarbe eines Drehknopfes ist beispielsweise die Hauptfläche des Drehreglers, und die Hintergrundfarbe ist die Grundfarbe der erhöhten Kante. Die Anzeige am unteren Rand des Farbauswahlfeldes gibt an, ob der Vordergrund, Hintergrund oder beide Flächen mit Farbe gefüllt werden. Ein schwarzer Rahmen um die Farbanzeige für Vordergrund oder Hintergrund gibt an, was gewählt ist. Bei der Standardeinstellung ist sowohl der Vordergrund als auch der Hintergrund gewählt.

Um zwischen Vordergrund und Hintergrund zu wechseln, können Sie <v> für *Vordergrund* und <h> für *Hintergrund* drücken. Durch Drücken von <a> für *beide (all)* wird sowohl der Vordergrund als auch der Hintergrund gewählt. Durch Drücken einer beliebigen anderen Taste kann ebenfalls zwischen Vordergrund und Hintergrund gewechselt werden.

Durch Wählen der Menüoption **Mehr...** in der Farbpalette erhalten Sie Zugriff auf ein Dialogfeld, in dem die Farben angepaßt werden können. Das Dialogfeld **Mehr...** ist in der folgenden Abbildung zu sehen.



Jede der drei Farbkomponenten Rot, Grün und Blau beschreibt acht Bits einer 24-Bit-Farbe. Jede Komponente weist daher einen Bereich von 0 bis 255 auf. Um den Wert einer Farbkomponente zu ändern, können Sie auf die Anzeige dieser Farbe doppelklicken und den neuen Wert eingeben. Um eine der Grundfarben zu verändern, klicken Sie in das Farbrechteck, und wählen Sie eine Farboption. Die Komponentenwerte der gewählten Farbe werden in den verschiedenen Feldern angezeigt.

Die Farbe, die Sie als letztes in der Palette auswählen, wird zur aktuellen Farbe. Wenn Sie anschließend mit dem Farbenwerkzeug auf ein Objekt klicken, nimmt dieses die aktuelle Farbe an.

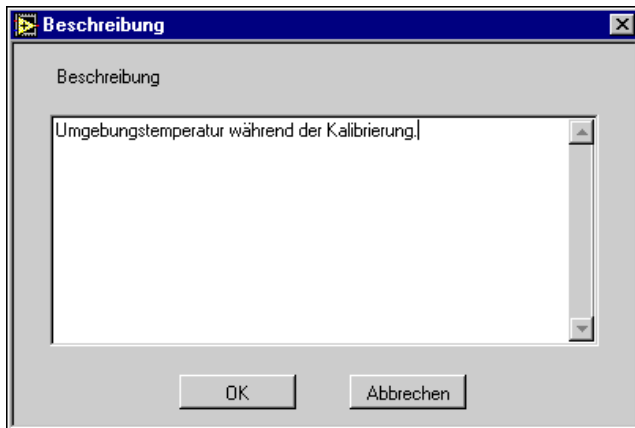
Es besteht auch die Möglichkeit, die Farbe eines Objekts zu kopieren und ohne Zuhilfenahme der **Farbpalette** auf ein anderes Objekt zu übertragen. Klicken Sie mit dem Werkzeug zum Kopieren von Farben bei gedrückter <Strg-Taste> (**Windows**), <Befehlstaste> (**Macintosh**), <Meta-Taste> (**Sun**) oder <Alt-Taste> (**HP-UX**) auf das Objekt, dessen Farbe Sie duplizieren wollen. Das Farbenwerkzeug nimmt die Gestalt einer Pipette an und übernimmt die Farbe des gewählten Objekts. Sie können nun mit dem Farbenwerkzeug auf ein anderes Objekt klicken, um dieses mit der soeben gewählten Farbe zu füllen.

Rückgängig machen

Zusätzlich zu <Strg-Z> für **Rückgängig** und <Strg-Umschalt-Z> für **Wiederherstellen** kann bei der Programmiersprache G die Anzahl der Aktionen festgelegt werden, die rückgängig gemacht bzw. wiederhergestellt werden können, und Aufrufende von VIs können automatisch aktualisiert werden, wenn sich die Schnittstelle des VIs ändert. Weitere Informationen hierzu finden Sie im Abschnitt *Rückgängig machen* von Kapitel 7, *Anpassen Ihrer Umgebung*.

Erstellen von Objektbeschreibungen

Wenn Sie eine Beschreibung eines G-Objekts, wie z.B. eines Bedien- oder Anzeigeelements, eingeben wollen, wählen Sie **Datenoperationen»Beschreibung...** aus dem Popup-Menü des Objekts. Zum Bearbeiten einer Beschreibung müssen Sie sich im Bearbeitungsmodus befinden. Geben Sie im Dialogfeld (siehe nachstehende Abbildung) eine Beschreibung ein, und klicken Sie zum Speichern auf **OK**. Die Beschreibung wird angezeigt, wenn Sie später im Popup-Menü des Objekts die Option **Beschreibung...** wählen.



Die Beschreibung des Objekts erscheint auch im Hilfefenster, wenn Sie den Cursor auf das Objekt plazieren. Die beste Methode, um für die von Ihnen erstellten VIs Hilfe einzurichten, besteht darin, für alle Bedien- und Anzeigeelemente Beschreibungen einzugeben.

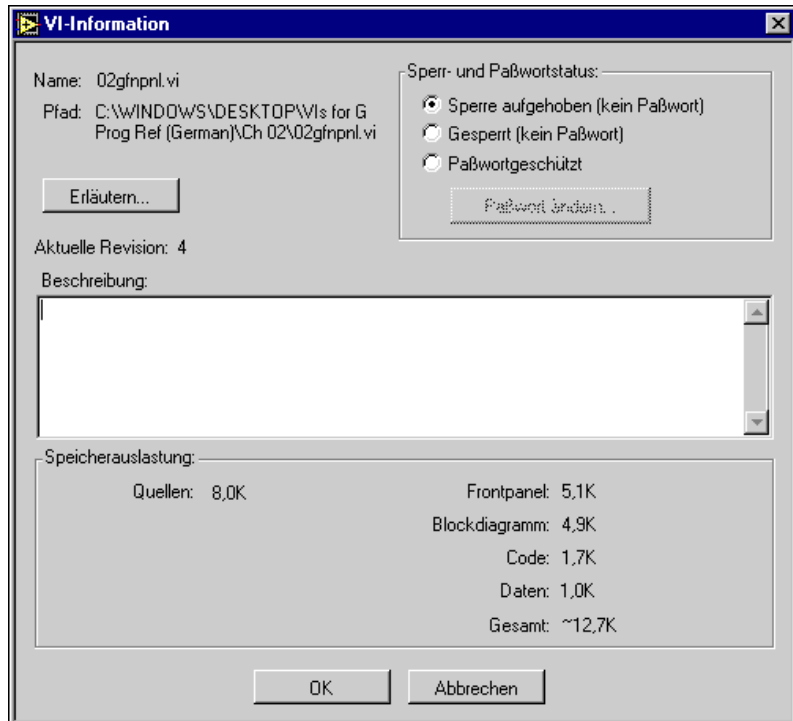


Hinweis

Sie können Beschreibungen von SubVIs nicht vom Diagramm des aufrufenden VIs aus bearbeiten. Sie können eine VI-Beschreibung jedoch über den Menüpunkt Fenster»VI-Info anzeigen... bearbeiten, wenn das Frontpanel des VIs offen ist.

Erstellen von VI-Beschreibungen

Durch Wählen von **Fenster»VI-Info anzeigen...** wird das Dialogfeld **VI Info** für das aktuelle VI angezeigt. Mit dem Dialogfeld **VI-Information** (siehe nachstehende Abbildung) können Sie die folgenden Aufgaben bewältigen:



- eine Beschreibung des VIs eingeben. Das Beschreibungsfenster besitzt einen Rollbalken zum Bearbeiten und Anzeigen längerer Beschreibungen.
- das VI fixieren oder freigeben. Ein fixiertes VI kann ausgeführt, aber nicht bearbeitet werden.
- das VI mit Paßwortschutz versehen. Weitere Informationen über Paßwortschutz finden Sie in Kapitel 27, [Verwalten Ihrer Anwendungen](#).
- die aktuelle Revisionsnummer anzeigen.
- den Pfad des VIs anzeigen.
- die Speicherauslastung durch das VI anzeigen. Im Abschnitt **Speicherauslastung** des Dialogfeldes wird der vom VI beanspruchte

Speicherplatz und Systemspeicher angezeigt. (Dieser Wert bezieht sich nur auf den vom VI beanspruchten Speicher, er beinhaltet nicht den Speicher, der von den einzelnen SubVIs in Anspruch genommen wird.)

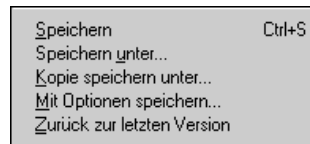
Die Speicherauslastung ist in den für Frontpanel und Blockdiagramm, VI-Code und Daten beanspruchten Speicher unterteilt. Die Speicherauslastung kann insbesondere beim Bearbeiten und Ausführen des VIs enormen Schwankungen unterworfen sein. Das Blockdiagramm erfordert in der Regel am meisten Speicher. Wenn Sie das Blockdiagramm nicht bearbeiten, sollten Sie daher das VI speichern und das Blockdiagramm schließen, um Speicher für zusätzliche VIs freizugeben. Durch Speichern und Schließen von SubVI-Frontpanels kann ebenfalls Speicher freigegeben werden.

Speichern von VIs

Sie können VIs in individuellen Dateien speichern oder mehrere VIs gruppieren und gemeinsam in einer VI-Bibliothek speichern.

Individuelle VI-Dateien

Fünf Menüpunkte im Menü **Datei** beziehen sich auf das Speichern Ihrer VIs in individuellen Dateien.



Wählen Sie den Menüpunkt **Speichern**, um Änderungen in einem bestehenden VI an dessen aktuellem Speicherort zu speichern. Ist das VI neu, erscheint ein Dateialogfeld, damit Sie den Namen und den Speicherort des VIs angeben können.

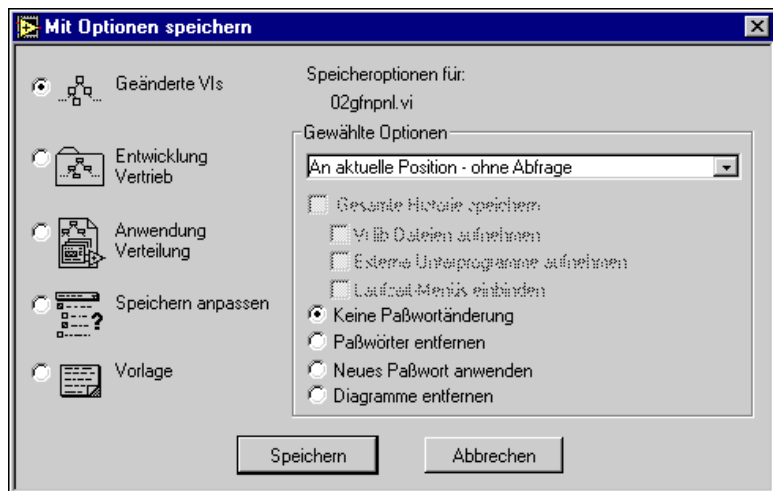
Wenn Sie ein VI unter einem neuen Namen speichern wollen, können Sie die Option **Speichern unter...**, **Kopie speichern unter...** oder **Mit Optionen speichern...** aus dem Menü **Datei** verwenden.

Wenn Sie den Menüpunkt **Speichern unter...** wählen, wird eine Kopie des im Arbeitsspeicher befindlichen VIs unter dem Namen, den Sie wählen, auf dem Speichermedium gespeichert. Nach der Speicherung ist das VI im Arbeitsspeicher mit der neuen Version verknüpft, und alle Aufrufenden des alten VIs im Arbeitsspeicher sind mit dem neuen VI verknüpft. Wenn Sie

einen neuen Namen für das VI eingeben, wird die auf dem Speichermedium gespeicherte Version des ursprünglichen VIs weder überschrieben noch gelöscht.

Wenn Sie **Kopie speichern unter...** wählen, wird eine Kopie des im Arbeitsspeicher befindlichen VIs unter dem Namen, den Sie eingeben, auf dem Speichermedium gespeichert. Dieser Vorgang hat keine Auswirkung auf den Namen des im Arbeitsspeicher befindlichen VIs.

Bei **Mit Optionen speichern...** wird ein Dialogfeld angezeigt, in dem Sie wählen können, ob eine gesamte VI-Hierarchie auf dem Speichermedium gespeichert werden soll, oder Sie können VIs mit Paßwortschutz oder ohne ihre Blockdiagramme speichern. Dies ist insbesondere nützlich, wenn Sie VIs verteilen oder Sicherungskopien anlegen.



Informationen zur Verwendung dieses Menüpunkts finden Sie in Kapitel 27, [Verwalten Ihrer Anwendungen](#).



Vorsicht

Sie können ein VI, wenn Sie es ohne ein Blockdiagramm speichern, nicht bearbeiten. Machen Sie stets eine Kopie des ursprünglichen VIs.

Mit dem Menüpunkt **Datei»Zurückkehren...** können Sie zur zuletzt gespeicherten Version des VIs zurückkehren, an dem Sie arbeiten. Ein Dialogfeld erscheint, in dem Sie bestätigen müssen, daß alle Änderungen im VI verworfen werden.

VIs, die sich seit der letzten Speicherung geändert haben, werden in der Titelleiste und der Liste der offenen VIs, die im Menü **Fenster** angezeigt wird, mit einem Sternchen markiert. Wenn Sie ein VI speichern, wird das Sternchen gelöscht und erst wieder angezeigt, wenn Sie eine neue Änderung vornehmen.

Informationen zum Speichern von Sicherungskopien finden Sie in Kapitel 27, *Verwalten Ihrer Anwendungen*.



Vorsicht

Speichern Sie die VIs nicht im Unterverzeichnis `vi.lib` der Software der G-Sprache. National Instruments aktualisiert dieses Verzeichnis bei neuen Versionsausgaben der Software der G-Sprache nach Bedarf. Durch das Plazieren von VIs in `vi.lib` besteht die Gefahr, daß bei zukünftigen Installationen ein Konflikt auftritt. Wenn Sie wollen, daß Ihre VIs in der Palette Funktionen erscheinen, schlagen Sie weitere Informationen im Abschnitt *Anpassen der Elemente- und Funktionenpaletten* in Kapitel 7, *Anpassen Ihrer Umgebung*.

VI-Bibliotheken (.LLBs)

Sie können mehrere VIs gruppieren und als VI-Bibliothek speichern. Besteht hierfür jedoch kein besonderer Grund, ist es vorzuziehen, sie in individuellen Dateien und in Verzeichnissen organisiert zu speichern. Lesen Sie die folgenden Empfehlungen, bevor Sie sich dazu entscheiden, VIs in einer Bibliothek zu speichern.

Gründe für das Speichern von VIs in Bibliotheken:

- Wenn Sie Microsoft Windows 3.1 als Betriebssystem verwenden oder beabsichtigen, Ihre Dateien auf die Windows 3.1-Plattform zu übertragen, können Sie beim Speichern der VIs in Bibliotheken (`.llb`-Dateien) Dateinamen bis zu 255 Zeichen verwenden. Andere Betriebssysteme unterstützen lange Dateinamen (31 Zeichen beim Macintosh, 255 Zeichen auf Windows 95/NT- und UNIX-Systemen.)
- Wenn Sie VIs auf andere Plattformen übertragen, ist die Übertragung einer VI-Bibliothek unter Umständen leichter als die Übertragung mehrerer individueller VIs.
- Ist Speicherplatz ein wichtiger Faktor, so lassen sich mit VI-Bibliotheken die Speicherplatzvoraussetzungen für VIs geringfügig reduzieren, da die darin enthaltenen individuellen Dateien komprimiert werden.

Gründe für das Speichern von VIs in individuellen Dateien:

- Wenn Sie die VIs in individuellen Dateien speichern, können Sie sie mit dem Dateisystem verwalten (Kopieren, Verschieben, Umbenennen, Sichern und Verwalten des Quellcodes).
- In einer VI-Bibliothek ist keine Hierarchie möglich — VI-Bibliotheken können keine Unterverzeichnisse und Ordner enthalten.
- Das Laden und Speichern von Dateien läuft über das Dateisystem schneller ab als bei VI-Bibliotheken. Während des Lade- und Speicherprozesses ist weniger Speicherplatz für temporäre Dateien erforderlich.
- Das Speichern von VIs und Bedienelementen in individuellen Dateien ist robuster als das Speichern des gesamten Projekts in einer einzigen Datei.
- VI-Bibliotheken sind nicht kompatibel mit den Steuerungen des Quellcodes im Professional Developer Toolkit.



Hinweis

Viele mit der Software der Sprache G ausgelieferten VIs sind in VI-Bibliotheken gespeichert. Dies gewährleistet einheitliche Speicherorte auf allen Plattformen.

Erstellen von VI-Bibliotheken

Gehen Sie folgendermaßen vor, um eine VI-Bibliothek zu speichern:

1. Wählen Sie **Speichern unter...**, oder **Kopie speichern unter...** aus dem Menü **Datei**.
2. **(Macintosh)** Wenn Sie gemäß der Konfiguration über **Voreinstellungen»Verschiedenes** systemeigene Dateidialogfelder verwenden, klicken Sie in dem anschließend angezeigten Dialogfeld auf **Use LLBs**.
3. Klicken Sie in dem im Anschluß an die Schritte 1 und 2 angezeigten Dialogfeld auf die Schaltfläche **Neue VI-Bibliothek**.
4. Geben Sie im nachfolgenden Dialogfeld den Namen der neuen Bibliothek ein, und klicken Sie auf die Taste **VI-Bibliothek**. Die Software der G-Programmiersprache fügt automatisch die Erweiterung `.llb` an den Namen der Bibliothek an.
5. Ein weiteres Dialogfeld wird angezeigt, in dem Sie auf Wunsch den VI-Namen angeben und in der neuen Bibliothek speichern können.

VIs in bestehenden VI-Bibliotheken speichern

Sie können neue VIs in einer VI-Bibliothek speichern, die hierbei mit einem Verzeichnis vergleichbar ist. Nach Auswahl des gewünschten Speicherpunktes aus dem Menü **Datei** erscheint der Name der Bibliotheksdatei, mit der Erweiterung `.llb`, im Dateidialogfeld. (Macintosh-Benutzer, die das systemeigene Dateidialogfeld verwenden, müssen zuerst die Schaltfläche **Use LLBs** drücken, bevor Sie eine VI-Bibliothek auswählen.) Wenn Sie auf den Namen der Bibliotheksdatei doppelklicken oder **Öffnen** wählen, bietet das Dialogfeld eine Option an, die Ihnen ermöglicht, die Datei in der Bibliothek zu speichern.

Bibliotheksinhalte bearbeiten

Mit dem Menüpunkt **Datei»VI-Bibliothek bearbeiten...** kann eine Datei aus einer VI-Bibliothek gelöscht werden. Wenn das Dialogfeld "VI-Bibliothek bearbeiten" erscheint, können Sie Dateien auch als **Höchste Ebene** markieren. Dies hat zweierlei Bedeutung. Erstens, wenn Sie mit den für die Anwendungserstellung vorgesehenen Bibliotheken eine Anwendung erstellen, gibt die Einstellung **Höchste Ebene** an, welche VIs beim Ausführen der Anwendung automatisch geöffnet werden. Zweitens, wenn Sie über das Dateisystem der Windows- oder Macintosh-Plattform auf eine bestimmte Bibliothek doppelklicken oder wenn Sie beim Aufrufen der G-Entwicklungsumgebung unter Windows oder UNIX in der Befehlszeile einen Bibliotheksnamen angeben, öffnet G automatisch alle VIs der höchsten Ebene.

VIs der höchsten Ebene werden bei der Anzeige des `.llb`-Inhalts in einem Dateidialogfeld auch durch eine Trennlinie von allen anderen VIs in der Bibliothek getrennt.

Verwendung von SubVIs

Dieses Kapitel beschreibt den Begriff des hierarchischen Aufbaus von G-Anwendungen und erläutert zwei Methoden zum Erstellen von SubVIs. Das Kapitel beschreibt außerdem zwei Dienstprogramme — das Hierarchiefenster, das die Hierarchie der VIs anzeigt, und das Suchdienstprogramm, das Ereignisse von SubVIs sowie andere Objekte oder Textstrings sucht, die Sie angeben.

Hierarchischer Aufbau

Ein Schlüssel zur Erstellung von G-Anwendungen ist das Verständnis und der Gebrauch der hierarchischen Natur eines VIs. Sobald Sie ein VI erstellt haben, können Sie es im Blockdiagramm eines VIs einer höheren Ebene als SubVI verwenden. Ein SubVI entspricht daher in C einem Unterprogramm. Wie bei C-Programmen, bei denen es keine Beschränkung der Anzahl von Unterprogrammen gibt, gibt es auch bei G keine Beschränkung der SubVIs, die in einem Programm verwendet werden können. Sie können auch ein SubVI innerhalb eines anderen SubVIs aufrufen.

Beim Erstellen einer Anwendung können Sie beim VI der höchsten Ebene beginnen und Eingaben und Ausgaben für die Anwendung definieren. Sie können dann andere VIs als SubVIs verwenden, um die erforderlichen Operationen an den Daten durchzuführen, während diese das Blockdiagramm durchlaufen. Besitzt ein Blockdiagramm eine große Anzahl von Icons, können Sie diese zu einem VI einer niedrigeren Ebene gruppieren, um die Übersichtlichkeit des Blockdiagramms beizubehalten. Dieser modulare Ansatz erleichtert Debugging, Verständnis und Pflege von Anwendungen.

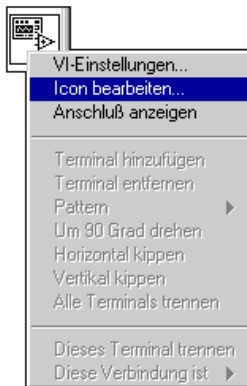
Sie können ein VI oder eine Auswahl (Teil) eines VIs in ein SubVI umwandeln. Beide Methoden werden in diesem Kapitel erläutert.

Erstellen von SubVIs aus VIs

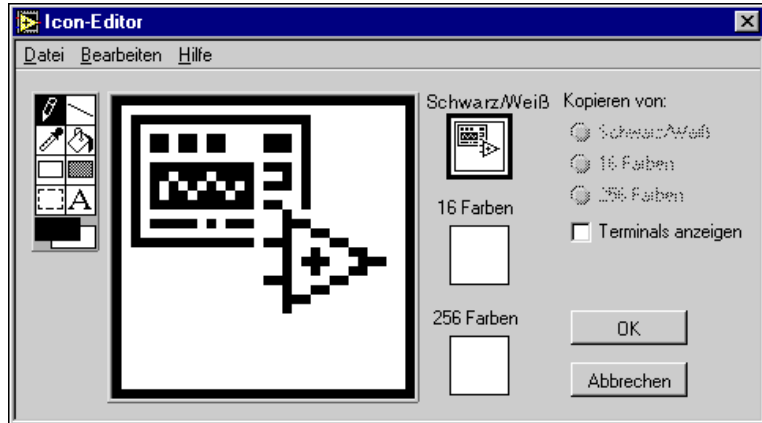
Das *Icon* ist das graphische Symbol für ein VI. Der *Anschluß* eines VIs weist Eingabe- und Ausgabeterminals Bedien- und Anzeigeelemente zu. Wenn Sie das VI vom Blockdiagramm eines anderen VIs aufrufen wollen, erstellen Sie zunächst ein Icon und einen Anschluß dafür. Dieser Abschnitt erläutert die Erstellung und Bearbeitung eines Icons und Anschlusses eines VIs.

Erstellen des Icons

Doppelklicken Sie zum Erstellen des Icons entweder auf das Symbol in der rechten oberen Ecke des Frontpanels, oder öffnen Sie das Popup-Menü des Icons, und wählen Sie **Icon bearbeiten...** (siehe Abbildung). Beachten Sie, daß Sie sich im Fenster des Frontpanels befinden müssen, um dieses Menü zu erhalten.



Sobald Sie **Icon bearbeiten...** gewählt haben, erscheint der Icon-Editor.



Verwenden Sie die Werkzeuge auf der rechten Seite des Fensters (siehe Abbildung oben), um im vergrößerten Bearbeitungsbereich das Design des Icons zu erstellen. Das Bild erscheint rechts vom Bearbeitungsbereich in seiner Normalgröße.

Je nachdem, welchen Monitortyp Sie verwenden, können Sie ein separates Icon für die Anzeige im Monochrom-, 16-Farben- und 256-Farbenmodus erstellen. Sie können jede Iconversion separat entwerfen und speichern. Standardeinstellung des Editors ist **Schwarz/Weiß**, aber Sie können auf einen anderen Farbmenüpunkt klicken, um zu wechseln. Sie können mit den Menüpunkten von **Kopieren von** auf der rechten Seite des Icon-Editors von einem Farbicon zu einem Schwarzweiß-Icon kopieren, und umgekehrt.



Hinweis

Es ist am besten, wenigstens ein Schwarzweiß-Icon zu erstellen. Wenn Sie nur ein Farbicon entwerfen, erscheint dieses nicht, wenn Sie es der Palette Funktionen hinzufügen. Außerdem kann es weder gedruckt werden, noch ist es auf einem Schwarzweiß-Monitor zu sehen. In solchen Fällen, wenn ein VI kein Schwarzweiß-Icon besitzt, verwendet die Software ein leeres Icon. Weitere Informationen hierzu finden Sie im Abschnitt [Anpassen der Elemente- und Funktionspaletten](#) in Kapitel 7 dieses Handbuchs, [Anpassen Ihrer Umgebung](#).

Die Werkzeug-Icons auf der linken Seite des Bearbeitungsbereiches erfüllen die folgenden Funktionen.



Bleistift — Zeichnet und löscht pixelweise. Benutzen Sie die <Umschalttaste>, um die Zeichenoperationen auf horizontale und vertikale Linien zu beschränken.



Linie — Zeichnet gerade Linien. Benutzen Sie die <Umschalttaste>, um die Zeichenoperationen auf horizontale, vertikale und diagonale Linien zu beschränken.



Pipette — Wählt eine Farbe von einem Element im Icon als Vordergrundfarbe. Benutzen Sie die <Umschalttaste>, um mit der Pipette die Hintergrundfarbe auszuwählen.



Füllbehälter — Füllt eine Fläche innerhalb eines Umrisses mit der Vordergrundfarbe.



Rechteck — Zeichnet einen rechteckigen Rahmen in der Vordergrundfarbe. Doppelklicken Sie auf dieses Werkzeug, um das Icon in der Vordergrundfarbe mit einem Rahmen zu versehen. Benutzen Sie die <Umschalttaste>, um ein Quadrat zu zeichnen.



Ausgefülltes Rechteck — Zeichnet ein mit der Hintergrundfarbe gefülltes Rechteck in der Vordergrundfarbe. Doppelklicken Sie auf dieses Werkzeug, um das Icon mit einem Rahmen (Vordergrundfarbe) zu versehen und mit der Hintergrundfarbe zu füllen. Benutzen Sie die <Umschalttaste>, um ein Quadrat zu zeichnen.



Auswählen — Wählt einen Bereich des Icons zum Verschieben, Kopieren oder Löschen aus. Doppelklicken Sie, um das gesamte Icon auszuwählen. Durch Drücken der <Entf>-Taste wird der gewählte Bereich des Icons gelöscht. Benutzen Sie die <Umschalttaste>, um ein Quadrat als Auswahlbereich zu definieren.



Text — Ermöglicht die Eingabe von Text in das Icon. Doppelklicken Sie auf dieses Werkzeug, um eine andere Schriftart auszuwählen. Benutzen Sie die Pfeiltasten, um den Text im Icon umherzuverschieben.



Vordergrund/Hintergrund — Zeigt die aktuellen Vordergrund- und Hintergrundfarben an. Durch Klicken erhalten Sie eine Palette, von der Sie neue Farben auswählen können.

Durch Drücken und Halten der <Strg-Taste> (**Windows**), <Optionstaste> (**Macintosh**), <Meta-Taste> (**Sun**) oder <Alt-Taste> (**HP-UX**) werden alle Werkzeuge mit Ausnahme des Auswahlwerkzeuges vorübergehend in die Pipette umgewandelt.

Die Tasten auf der rechten Seite der Bearbeitungsanzeige erfüllen die folgenden Funktionen, wenn Sie darauf klicken.

OK — Speichert die Zeichnung als VI-Icon und kehrt zum Frontpanel zurück.

Abbrechen — Kehrt zum Frontpanel zurück, ohne daß irgendwelche Änderungen gespeichert werden.



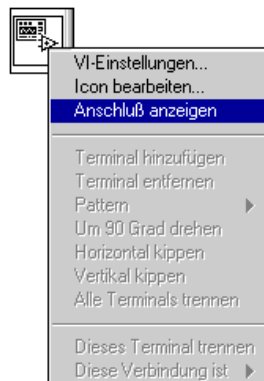
Hinweis

Mit dem Menü Bearbeiten können Sie Bilder vom Icon ausschneiden, kopieren und einfügen, und umgekehrt. Wenn Sie ein Bild einfügen und nur ein Teil des Icons ausgewählt ist, wird die Größe des Bildes an die Auswahl angepaßt.

Definieren von Anschlußterminal-Mustern

Der Datenaustausch erfolgt bei einem SubVI über die Terminals und dessen Anschlußfeld. Sie definieren Verbindungen, indem Sie die Anzahl der für das VI gewünschten Terminals wählen und jedem dieser Terminals ein Bedien- oder Anzeigeelement am Frontpanel zuweisen. Nur die Bedien- und Anzeigeelemente, die Sie mit dem Programm durch Verbinden des SubVIs anwenden, erfordern Terminals am Anschlußfeld.

Wenn der Anschluß für Ihr VI noch nicht in der rechten oberen Ecke des Frontpanels angezeigt wird, wählen Sie **Anschluß anzeigen** im Pop-up-Menü des Icon-Anschlußfeldes (siehe nachstehende Abbildung). Das Blockdiagramm besitzt kein Anschlußfeld.

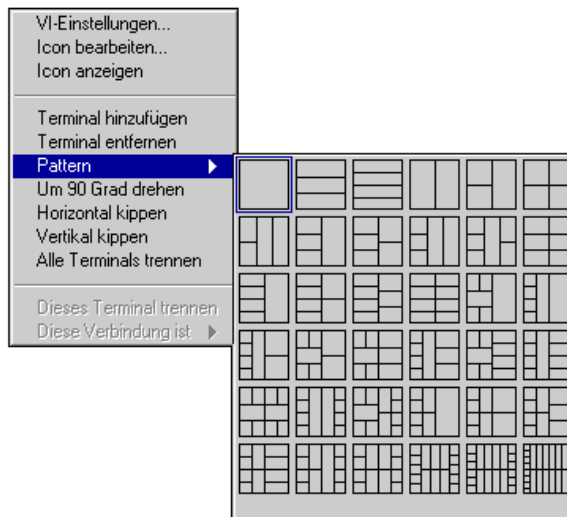


Der Anschluß ersetzt das Icon in der rechten oberen Ecke des Frontpanels. Die anfängliche Auswahl ist ein Terminal-Muster mit ebenso vielen Terminals auf der linken Seite des Anschlußfeldes, wie Bedienelemente auf dem Frontpanel zu finden sind, und ebenso vielen Terminals auf der rechten Seite des Anschlußfeldes, wie Anzeigeelemente auf dem Frontpanel zu finden sind. Ist dies nicht möglich, wählt die Software eine Annäherung an diese Anordnung.

Jedes Rechteck am Anschluß stellt einen Terminalbereich dar und kann entweder als Eingang oder Ausgang des VIs verwendet werden. Wenn Sie für das VI ein anderes Terminal-Muster verwenden wollen, können Sie ein anderes Muster auswählen.

Auswählen und Ändern von Terminal-Mustern

Um für das VI ein anderes Terminal-Muster zu verwenden, wählen Sie **Pattern** aus dem Popup-Menü des Anschlusses.



Das mit dem Icon verknüpfte Muster wird durch eine schwarze Umrandung hervorgehoben. Um das Muster zu ändern, wählen Sie ein anderes aus.

Wenn Sie dem Muster ein Terminal hinzufügen wollen, plazieren Sie den Cursor an die Stelle, an der das Terminal hinzugefügt werden soll, und wählen Sie **Terminal hinzufügen** aus dem Popup-Menü.

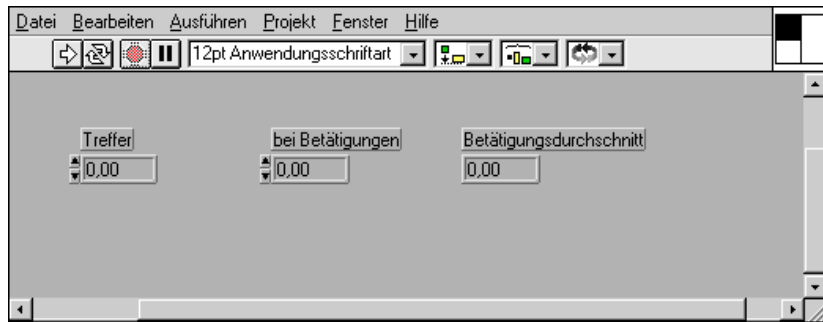
Wenn Sie ein bestehendes Terminal aus dem Muster entfernen wollen, wählen Sie **Terminal entfernen** aus dem Popup-Menü.

Wenn Sie die räumliche Verteilung der Muster von Anschlußterminals ändern wollen, wählen Sie einen der folgenden Befehle aus dem Pop-up-Menü des Anschlußfeldes: **Horizontal kippen**, **Vertikal kippen** oder **Um 90 Grad drehen**.

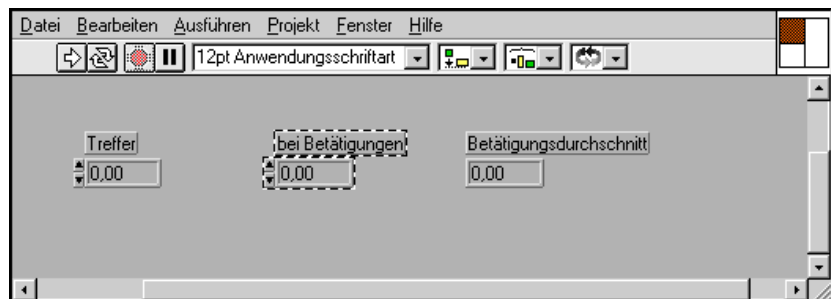
Zuweisen von Bedien- und Anzeigeelementen zu Terminals

Nachdem Sie entschieden haben, welches Terminalmuster Sie für Ihren Anschluß verwenden wollen, müssen Sie den Terminals Bedien- und Anzeigeelementen am Frontpanel zuweisen. Halten Sie sich an diese Verfahrensschritte:

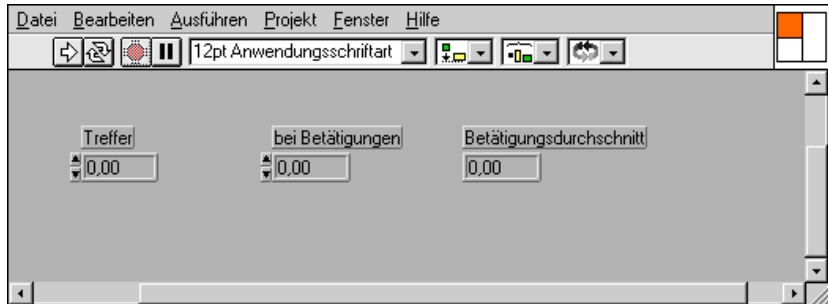
1. Klicken Sie auf das Terminal des Anschlusses. Das Werkzeug verwandelt sich automatisch in das Verbindungswerkzeug. Das Terminal wird schwarz.



2. Klicken Sie am Frontpanel auf das Bedien- oder Anzeigeelement, dem Sie das ausgewählte Terminal zuweisen wollen. Das gewählte Bedienelement wird durch ein Auswahlrechteck hervorgehoben.



3. Stellen Sie den Cursor auf eine freie Stelle am Frontpanel ein, und klicken Sie. Das Auswahlrechteck verschwindet, und das ausgewählte Terminal nimmt die Datenfarbe des verbundenen Bedienelements an, womit angezeigt wird, daß das Terminal belegt ist.



Hinweis

Wird das Anschlußterminal weiß, so ist keine Verbindung hergestellt worden. Wiederholen Sie die Schritte 1 bis 3, bis das Anschlußterminal die entsprechende Datenfarbe annimmt.

Obgleich Sie das Verbindungswerkzeug verwenden, um Bedien- und Anzeigeelementen am Frontpanel Terminals am Anschluß zuzuweisen, werden zwischen dem Anschluß und diesen Bedien- und Anzeigeelementen keine Verbindungen gezeichnet.

4. Wiederholen Sie die Schritte 1 und 2 für jedes Bedien- und Anzeigeelement, das Sie verbinden wollen.

Sie können auch zuerst das Bedien- oder Anzeigeelement und dann das Terminal auswählen. Sie können ein Muster aus mehr als den erforderlichen Terminals wählen und einige überschüssige Terminals unverbunden lassen, falls spätere Änderungen am VI absehbar sind, die einen neuen Ein- oder Ausgang erfordern. Die Verfügbarkeit dieser zusätzlichen Verbindungen bedeutet, daß der neue Ein- oder Ausgang sich nicht auf andere VIs auswirkt, die dieses VI als SubVI nutzen. Unbelegte Terminals haben keinerlei Auswirkung auf den Betrieb des VIs. Auch kann ein Frontpanel mehr Bedienelemente besitzen als Terminals.

Das Anschlußfeld besitzt maximal 28 Terminals. Wenn Ihr Frontpanel mehr als 28 Bedien- und Anzeigeelemente enthält, die Sie innerhalb von Programmen nutzen wollen, gruppieren Sie einige davon zu einem Cluster, und weisen Sie dem Cluster ein Terminal am Anschluß zu. Weitere Informationen hierzu finden Sie im Abschnitt *Cluster* in Kapitel 14 dieses Handbuchs, *Bedien- und Anzeigeelemente vom Typ Array und Cluster*.

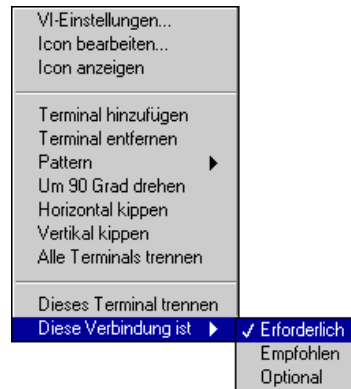
Erforderliche, empfohlene und optionale Verbindungen für SubVIs

G verfügt über eine Funktion, die Sie daran erinnern kann, Verbindungen von SubVIs zu setzen — und zwar mit Hilfe von Anzeigern für erforderliche, empfohlene und optionale Verbindungen im Anschlußfeld und denselben Anzeigern im Hilfefenster.

Ein- und Ausgänge von VIs, die in `vi.lib` enthalten sind, sind werkseitig bereits als **Erforderlich**, **Empfohlen** oder **Optional** markiert. G legt Ein- und Ausgaben, die Sie erstellen, standardmäßig als **Empfohlen** fest.

Ist ein Eingang als **Erforderlich** markiert, kann das VI ohne die richtigen Verbindungen nicht als SubVI ausgeführt werden. Ist ein Ein- oder Ausgang als “empfohlen” markiert, können Sie das VI zwar ausführen, aber das Fehlerlistenfenster führt eine Warnung an, falls Warnungen aktiviert sind.

Um festzustellen, ob Verbindungen **erforderlich**, **empfohlen** oder **optional** sind, oder um sie als einen dieser Zustände zu markieren, klicken Sie im Anschlußfeld auf ein Terminal, und wählen Sie einen Menüpunkt aus dem Untermenü **Diese Verbindung ist**. Der Status wird durch ein Häkchen angezeigt (siehe nachstehende Abbildung).

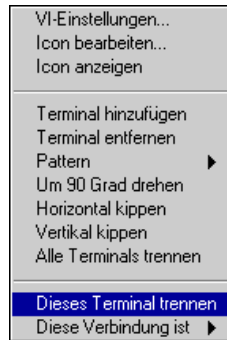


Im Hilfefenster werden Verbindungen vom Typ **Erforderlich** fettgedruckt, Verbindungen vom Typ **Empfohlen** normal und Verbindungen vom Typ **Optional** als nicht bedienbar markiert.

Löschen von Terminalverbindungen

Sie können Verbindungen zwischen Terminals und ihren Bedien- oder Anzeigeelementen einzeln oder global löschen. Um eine bestimmte Verbindung zu löschen, wählen Sie das Pop-up-Menü des Terminals,

dessen Verbindung Sie trennen wollen, und wählen Sie **Dieses Terminal trennen** aus dem Popup-Menü (siehe nachstehende Abbildung).



Das Terminal wird weiß, wodurch angezeigt wird, daß die gewählte Verbindung nicht mehr besteht. Um alle Verbindungen am Anschluß zu löschen, wählen Sie an einer beliebigen Stelle auf dem Anschluß **Alle Terminals trennen** aus dem Popup-Menü. Beachten Sie, daß sich dieser Vorgang insofern von **Terminal entfernen** unterscheidet, als **Dieses Terminal trennen** das Terminal nicht aus dem Muster entfernt.

Bestätigen von Terminalverbindungen

Um festzustellen, welches Bedien- oder Anzeigeelement einem bestimmten Terminal zugewiesen ist, klicken Sie, wenn das Anschlußfeld sichtbar ist, mit dem Verbindungswerkzeug auf ein Bedienelement, Anzeigeelement oder Terminal. G wählt dann das entsprechende zugewiesene Objekt.

Erstellen von SubVIs anhand einer VI-Auswahl

Sie können einen Teil eines VIs in ein SubVI umwandeln, das von einem anderen VI aufgerufen wird. Sie wählen einen Ausschnitt eines VIs und anschließend **Bearbeiten»SubVI erstellen**, und der Ausschnitt wird zu einem SubVI. Für das neue SubVI werden automatisch Bedien- und Anzeigeelemente erstellt, es wird automatisch in die bestehenden Verbindungen integriert, und im ursprünglichen VI wird der gewählte Ausschnitt des Blockdiagramms automatisch durch ein Icon des SubVIs ersetzt.

Mit Ausnahme der folgenden Verhaltensweisen entspricht das Erstellen eines SubVIs anhand einer Auswahl dem Entfernen der gewählten Objekte und Ersetzen durch ein SubVI:

- Keines der in der Auswahl enthaltenen Frontpanel-Terminals wird vom aufrufenden VI entfernt. Die Terminals des Frontpanels bleiben vielmehr am aufrufenden VI erhalten und werden mit dem SubVI verbunden.
- Alle in der Auswahl enthaltenen Attribute bleiben am aufrufenden VI erhalten und werden mit dem SubVI verbunden. Die gewählten Attribute werden durch Frontpanel-Terminals im SubVI ersetzt, die als Kanäle zur Übertragung eines Attributwerts in das oder vom SubVI aus fungieren.
- Wenn eine Auswahl eine lokale Variable enthält, wird sie im SubVI durch ein Frontpanel-Terminal ersetzt. Die lokale Variable bleibt am aufrufenden VI erhalten und wird mit dem SubVI verbunden. Wenn mehrere Instanzen derselben lokalen Variable gewählt werden, wird die erste Instanz zu einem Frontpanel-Terminal, und die übrigen bleiben im SubVI als lokale Variablen erhalten. Am aufrufenden VI bleibt nur eine Instanz der lokalen Variable erhalten; die übrigen gewählten Instanzen werden vom aufrufenden VI entfernt und in das SubVI verschoben.
- Da lokale Variable an einem Frontpanel-Terminal entweder lesen oder schreiben können, werden zwei Frontpanel-Objekte im SubVI erstellt, wenn Instanzen der lokalen Variablen sowohl zum Lesen als auch zum Schreiben gewählt werden — eines, um den Wert der lokalen Variable an das SubVI zu übermitteln, und ein anderes, um den Wert aus dem SubVI abzurufen. An den Namen des Frontpanel-Terminals, das zur Darstellung der lokalen Lesevariable erstellt wird, wird das Suffix “read” angefügt, und an den Namen des Frontpanel-Terminals, das zur Darstellung der lokalen Schreibvariablen erstellt wird, wird das Suffix “write” angefügt.

Regeln und Empfehlungen

Die Fähigkeit, von einer Auswahl ein SubVI zu erstellen, kann sehr nützlich sein, ist aber nicht unbedingt so leicht wie es scheint. Eine sorgfältige Planung ist dennoch erforderlich, um eine logische Hierarchie von VIs zu erstellen. Sie müssen auch in Betracht ziehen, welche Objekte in die Auswahl einzubeziehen sind, um Situationen zu vermeiden, in denen die Funktionalität eines sich ergebenden VIs geändert wird. Falls ein Problem auftritt, wird anhand der Auswahl kein SubVI erstellt; es wird jedoch ein Dialogfeld angezeigt, in dem der Grund für das Fehlschlagen der Operation erläutert wird. Liegt lediglich ein potentiell Problem vor, wird ein Dialogfeld mit einer Erläuterung des potentiellen Problems angezeigt. Sie können dann selbst entscheiden, ob Sie fortfahren wollen.



Hinweis

Beinhaltet eine Auswahl weder lokale Variable noch Attributknoten, empfiehlt es sich möglicherweise, eine Sequenzstruktur zu benutzen, um den Code, der an ein SubVI gesendet werden soll, einzukapseln und die Ergebnisse in einer Vorschau anzuzeigen.

Die folgenden Regeln und Empfehlungen können dabei helfen, diese Funktion wirkungsvoll zu nutzen.

Anzahl der Verbindungen

Vermeiden Sie sehr große Auswahlätze, die SubVIs mit mehr als 28 Eingängen und Ausgängen (der maximalen Anzahl von Verbindungen in einem Anschlußfeld) erstellen.

Beachten Sie, daß jedes Frontpanel-Terminal, jedes Attribut und bestimmte lokale Variablen in der Auswahl einen Slot im Anschlußfeld erfordern; eine Auswahl mit einer großen Anzahl dieser Objekte könnte dazu führen, daß nicht genug Slots im Anschlußfeld zur Verfügung stehen.

Um ein Überschreiten der maximalen Anzahl von Terminals zu vermeiden, wählen Sie einen kleineren Bereich aus dem Diagramm, oder gruppieren Sie Daten zu Arrays und Clustern, bevor Sie einen Bereich des Diagramms zur Umwandlung auswählen.

Zyklen

Vermeiden Sie Auswahlätze, die zu Zyklen im Diagramm führen. Zyklen entstehen, wenn ein Datenfluß von einem Ausgang des SubVIs ausgeht und am SubVI wieder als Eingang endet.

Es ist schwierig, beim Definieren von Auswahlätzen Zyklen zu identifizieren, G stellt sie jedoch für Sie fest. Wenn G einen Zyklus feststellt, wird ein Dialogfeld angezeigt, in dem Sie dazu aufgefordert werden, anhand der Auswahl entweder ein neues VI zu erstellen oder den Vorgang abzubrechen. Entscheiden Sie sich dazu, ein neues VI zu erstellen, wird anhand der Auswahl ein unbenanntes VI erstellt. Die gewählten Objekte im ursprünglichen Diagramm bleiben unberührt.

Attributknoten mit Schleifen

Nehmen Sie keine Attributknoten innerhalb einer Schleife in Ihre Auswahl auf.

Da ein Attributknoten am aufrufenden VI mit Verbindung zum SubVI erhalten bleibt, wird der Wert des Attributs bei Ausführung des SubVIs

nicht bei jeder Iteration der Schleife aktualisiert. In diesen Fällen ist kein SubVI möglich.

Unlogische Auswahlätze

Wandeln Sie an sich sinnlose Auswahlätze nicht in ein SubVI um. Es ist beispielsweise sinnlos, Auswahlätze umzuwandeln, die aus einem Objekt innerhalb einer Sequenzstruktur und einem anderen Objekt außerhalb der Sequenzstruktur bestehen, ohne die Sequenzstruktur selbst in die Auswahl miteinzubeziehen.

Bei solchen Auswahlätzen folgt eine Anzeige mit einer Erläuterung des Problems.

Lokale Variable und Frontpanel-Terminals innerhalb von Schleifen

Vermeiden Sie in Ihrer Auswahl lokale Variablen oder Frontpanel-Terminals innerhalb einer Schleife.

Da ausgewählte Frontpanel-Terminals oder lokale Variablen am aufrufenden VI mit Verbindung zum SubVI erhalten bleiben, wird der Wert des Attributs bei Ausführung des SubVIs nicht bei jeder Iteration der Schleife aktualisiert. Dies kann zu einer Änderung der Funktionalität des aufrufenden VIs führen.

In solchen Fällen werden Sie mit einer Warnungsanzeige zum Fortfahren oder Abbrechen aufgefordert.

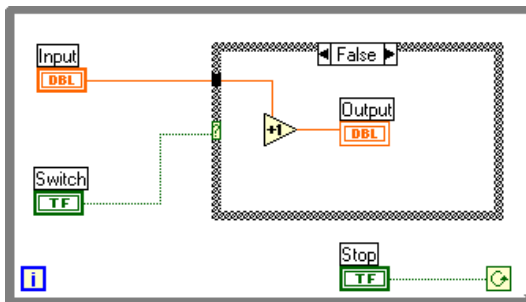
Case-Strukturen mit Attributknoten, lokalen Variablen oder Frontpanel-Terminals

Vermeiden Sie die Auswahl einer Case-Struktur mit einem Attributknoten, Frontpanel-Terminal oder einer lokalen Variablen, in die ein Wert geschrieben wird.

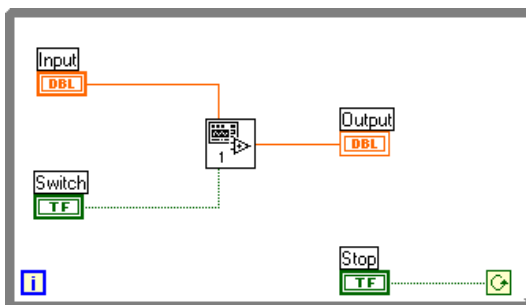
In solchen Fällen wird bei der Ausführung des SubVIs stets ein bestimmter Wert in das Objekt geschrieben, da Sie dieses mit dem SubVI verbinden. Ein Wert kann, obgleich er im ursprünglichen Blockdiagramm existiert, nur dann geschrieben werden, wenn der Teil der Case-Struktur ausgeführt wird, der entweder TRUE oder FALSE ist. Dies kann zu einer Änderung der Funktionalität des aufrufenden VIs führen.

In solchen Fällen werden Sie mit einer Warnungsanzeige zum Fortfahren oder Abbrechen aufgefordert. Wenn Sie fortfahren, müssen Sie zunächst das SubVI bearbeiten und für alle Fälle einen Wert angeben.

Betrachten Sie als Beispiel die folgende Abbildung.



Die Case-Struktur wird zur Umwandlung in ein SubVI ausgewählt. Die Ausgabe des Frontpanel-Terminals ist im ursprünglichen Diagramm weggelassen worden und wird über einen Anschluß mit dem SubVI verbunden (siehe Abbildung unten).



Im ursprünglichen Blockdiagramm hat G nur dann einen Wert in die Ausgabe geschrieben, wenn der Abschnitt FALSE des Falls innerhalb des SubVIs ausgeführt worden ist. Wenn die Auswahl jedoch in ein SubVI umgewandelt wird, schreibt G unabhängig davon, ob der Abschnitt TRUE oder FALSE innerhalb des SubVIs ausgeführt worden ist, immer einen Wert in die Ausgabe. Sie müssen das SubVI bearbeiten, um Daten für die zusätzlichen Fälle anzugeben.

Verwendung des Hierarchiefensters

Das Hierarchiefenster zeigt für alle VIs im Speicher eine grafische Darstellung der aufrufenden Hierarchie an, einschließlich der Typendefinitionen und globalen Variablen.

Viele Aspekte der Anzeige im Hierarchiefenster sind konfigurierbar. Sie können beispielsweise das Layout horizontal oder vertikal anzeigen und VIs der Bibliothek `vi.lib` sowie globale Variablen oder Typendefinition einschließen oder ausschließen.

Andere nützliche Funktionen des Hierarchiefensters sind u.a. die Fähigkeit, auf die Menüpunkte **VI-Einstellungen...**, **Icon bearbeiten...**, **VI-Info anzeigen...** und **Dokumentation drucken...** zuzugreifen, die Fähigkeit, Hierarchieknoten als SubVI in ein Blockdiagramm eines anderen VIs zu ziehen bzw. zu kopieren und einzufügen, sowie die Fähigkeit, Hierarchieknoten nach dem Namen zu suchen.

Öffnen des Hierarchiefensters

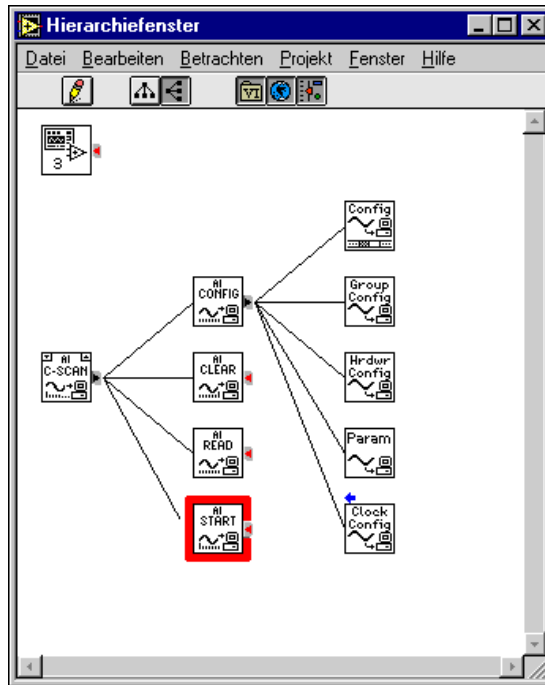
Sie können das Hierarchiefenster auf eine der folgenden Arten öffnen:

- Wählen Sie **Projekt»VI-Hierarchie**. Das Hierarchiefenster wird mit dem VI des jeweils aktiven Fensters als Fokusknoten angezeigt.
- Wählen Sie **VI-Hierarchie** aus den Popup-Menüs von SubVIs, globalen Variablen und Typendefinitionen. Das Hierarchiefenster erscheint, wobei das gewählte SubVI, die globale Variable oder Typendefinition den Fokusknoten darstellt.
- Ist das Hierarchiefenster bereits offen, können Sie es in den Vordergrund bringen, indem Sie es im Menü **Fenster** aus der Liste der offenen Fenster wählen.

Mit einem Menüpunkt im Menü **Betrachten** oder durch Drücken der Tasten **Horizontales Layout** oder **Vertikales Layout** oben im Fenster wird zwischen horizontaler und vertikaler Anzeige des Hierarchiefensters umgeschaltet.

Bei einer horizontalen Anzeige werden SubVIs rechts von den zugehörigen aufrufenden VIs angezeigt; bei einer vertikalen Anzeige werden sie unter den aufrufenden VIs angezeigt. SubVIs werden stets durch Linien mit den

zugehörigen aufrufenden VIs verbunden. Bei der folgenden Abbildung handelt es sich um die horizontale Anzeige einer Hierarchie.



Pfeiltasten und Pfeile neben Knoten zeigen an, was angezeigt bzw. verborgen wird, und zwar folgendermaßen:



- Eine rote Pfeiltaste, die auf den Knoten zeigt, weist darauf hin, daß manche oder alle SubVIs verborgen sind. Durch Klicken auf die Taste werden die unmittelbaren SubVIs des Knotens angezeigt.



- Eine schwarze Pfeiltaste, die auf die SubVIs des Knotens zeigt, weist darauf hin, daß alle unmittelbaren SubVIs angezeigt werden.



- Ein blauer Pfeil, der auf die aufrufenden Objekte des Knotens zeigt, weist darauf hin, daß der Knoten weitere aufrufende Objekte besitzt, die nicht angezeigt werden.

Besitzt ein Knoten keine SubVIs, wird weder eine rote, noch eine schwarze Pfeiltaste angezeigt.

Ein Knoten, an dem eine Operation durchgeführt wird, wird zum Fokusknoten, der von einer dicken, roten Umrandung umgeben ist. Der Fokus wechselt, wenn Sie eine Aktion an einem anderen Knoten durchführen.

Wenn Sie den Cursor über Objekte im Hierarchiefenster verschieben und der Cursor auf einem Icon pausiert, zeigt G den Namen des Knotens unterhalb des Knoten-Icons an. Wenn Sie dies vorziehen, können Sie einen Menüpunkt im Menü **Betrachten** verwenden, um statt des Namens den vollständigen Pfad anzuzeigen.

Sie können im Hierarchiefenster auf ein beliebiges Icon doppelklicken, um das zugehörige VI anzuzeigen.

Wenn Sie das Hilfefenster anzeigen und den Cursor auf ein Icon verschieben, zeigt G im Hilfefenster Informationen zu diesem VI an.

Optionen im Hierarchiefenster

Mehrere Optionen ermöglichen Ihnen, die Anzeige des Hierarchiefensters zu steuern; andere erlauben Ihnen, Aktionen an den im Fenster angezeigten Knoten durchzuführen. Optionen stehen im Menü **Betrachten**, mit Tasten am oberen Rand des Hierarchiefensters, im Popup-Menü eines Knotens und über Mausklicks zur Verfügung. Zugriff auf das Menü **Betrachten** besteht auch durch Aktivieren eines Popup-Menüs im freien Raum des Hierarchiefensters.

Optionen im Menü “Betrachten”

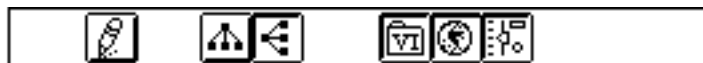
Das Menü **Betrachten** enthält die folgenden, die Anzeige des Hierarchiefensters betreffenden Menüpunkte. Viele der Menüpunkte stehen auch über Tasten am oberen Rand des Fensters zur Verfügung.

Erneut Zeichnen	Ctrl+D
Alle VIs anzeigen	Ctrl+A
<input checked="" type="checkbox"/> Vertikale Hierarchie	
<input type="checkbox"/> Horizontale Hierarchie	
<input type="checkbox"/> VIs in vi.lib aufnehmen	
<input checked="" type="checkbox"/> Globale Variable aufnehmen	
<input type="checkbox"/> Type Defs aufnehmen	
Gesamter VI-Pfad in Label	

- **Erneut Zeichnen** — Baut das Fensterlayout neu auf, um Linienüberschneidungen zu minimieren und Symmetrie zu maximieren; eine nützliche Option nach aufeinanderfolgenden Operationen an Hierarchieknoten. Existiert ein Fokusknoten, rollt das Fenster weiter, um den Knoten anzuzeigen. Existiert kein Fokusknoten, rollt das Fenster weiter, um die erste Wurzel von SubVIs anzuzeigen.
- **Alle VIs anzeigen** — Zeigt alle verborgenen VIs an. Wenn dieser Menüpunkt gewählt wird, gibt es keinen Fokusknoten mehr. Dies wirkt sich nicht auf andere Einstellungen im Menü **Betrachten** aus. In anderen Worten, VIs in `vi.lib`, globale Variable und Typdefinitionen, die nicht in die Hierarchie aufgenommen worden sind, bleiben verborgen.
- **Vertikale Hierarchie** — Ordnet die Knoten von oben nach unten an, wobei die aufrufenden VIs über den SubVIs angezeigt werden.
- **Horizontale Hierarchie** — Ordnet die Knoten von links nach rechts an, wobei die aufrufenden VIs zur Linken der SubVIs angezeigt werden.
- **VIs in `vi.lib` aufnehmen** — Schaltet die Anzeige von VIs in `vi.lib` im Hierarchiefenster ein oder aus.
- **Globale Variable aufnehmen** — Schaltet die Anzeige von globalen Variablen im Hierarchiefenster ein oder aus.
- **Type Defs aufnehmen** — Schaltet die Anzeige von Typdefinitionen im Hierarchiefenster ein oder aus.
- **Full VI Path in Label** — Schaltet die Anzeige im Hierarchiefenster entweder auf den vollständigen VI-Pfad oder lediglich den VI-Namen im Tip-Strip für jeden Hierarchieknoten um.

Tasten der Hierarchie-Werkzeugpalette

Eine Werkzeugpalette im Hierarchiefenster enthält Tasten, die die Anzeige des Fensters betreffen.



Die Tasten, die einige derselben Aktionen wie die Menüpunkte im Menü **Betrachten** durchführen, sind im folgenden beschrieben:



Layout wiederherstellen — Baut das Fensterlayout neu auf, um Linienüberschneidungen zu minimieren und Symmetrie zu maximieren; eine nützliche Option nach aufeinanderfolgenden Operationen an Hierarchieknoten. Existiert ein Fokusknoten, rollt das Fenster weiter, um den Knoten anzuzeigen. Existiert kein Fokusknoten, rollt das Fenster weiter, um die erste Wurzel von SubVIs anzuzeigen.



Vertikales Layout — Ordnet die Knoten von oben nach unten an, wobei die aufrufenden VIs über den SubVIs angezeigt werden.



Horizontales Layout — Ordnet die Knoten von links nach rechts an, wobei die aufrufenden VIs zur Linken der SubVIs angezeigt werden.



VI-Bibl. aufnehmen — Schaltet die Anzeige von VIs in `vi.lib` im Hierarchiefenster ein oder aus.



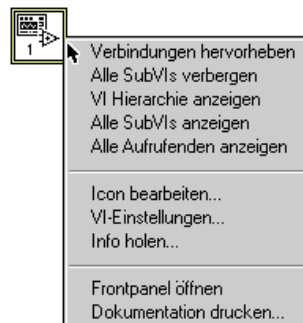
Globale Variable aufnehmen — Schaltet die Anzeige von globalen Variablen im Hierarchiefenster ein oder aus.



Typ. Def. aufnehmen — Schaltet die Anzeige von Typendefinitionen im Hierarchiefenster ein oder aus.

Popup-Menü von Hierarchieknoten

Wenn Sie das Popup-Menü eines im Hierarchiefenster angezeigten Knotens (SubVI, globale Variable oder Typendefinition) wählen, erscheint ein Menü mit Optionen zum Steuern der Anzeige oder Ausführen von Befehlen, die den gewählten Knoten betreffen.



Die Menüpunkte werden wie folgt beschrieben:

- **Verbindungen hervorheben** — Macht den gewählten Knoten zum Fokusknoten und hebt die Kanten, die die direkten Aufrufenden und SubVIs verbinden, rot hervor.
- **Direkte SubVIs anzeigen** — Sind alle oder manche SubVIs des Knotens verborgen, erweitert dieser Menüpunkt den Knoten, um alle direkten SubVIs anzuzeigen. Die Kanten, die den Knoten und die SubVIs miteinander verbinden, werden rot hervorgehoben. Der Menüpunkt wird mit **Alle SubVIs verbergen** umgeschaltet.
- **Alle SubVIs verbergen** — Werden alle direkten SubVIs angezeigt, klappt dieser Menüpunkt den Knoten zusammen, um dessen gesamte SubVI-Kette zu verbergen. Dieser Menüpunkt wird mit **Direkte SubVIs anzeigen** umgeschaltet.
- **VI-Hierarchie anzeigen** — Macht den gewählten Knoten zum Fokusknoten und zeigt die Knoten an, die zu dessen Aufrufkette und SubVI-Kette gehören. Damit nicht in Beziehung stehende Wurzeln werden zwar auch angezeigt, aber alle ihre SubVIs sind verborgen. Die Kanten, die die Aufrufkette eines Knotens und die SubVI-Kette miteinander verbinden, werden rot hervorgehoben.
- **Alle SubVIs anzeigen** — Macht den gewählten Knoten zum Fokusknoten und erweitert dessen gesamte SubVI-Kette. Die Kanten, die die SubVI-Kette des Knotens miteinander verbinden, werden rot hervorgehoben.
- **Alle Aufrufenden anzeigen** — Macht den gewählten Knoten zum Fokusknoten und erweitert dessen gesamte Aufrufkette. Die Kanten, die in der Aufrufkette die Verbindungen darstellen, werden rot hervorgehoben.
- **Icon bearbeiten...** — Zeigt den Icon-Editor an, mit dem das Icon des Knotens bearbeitet werden kann.
- **VI-Einstellungen...** — Zeigt das Dialogfeld “VI-Setup” für den Knoten an.
- **Info holen...** — Zeigt das Dialogfeld “Info holen” für den Knoten an.
- **Frontpanel öffnen** — Öffnet das Frontpanel des VIs, der globalen Variablen oder Typendefinition.
- **Dokumentation drucken...** — Ruft das Dialogfeld “Dokumentation drucken” auf, in dem Sie den VI-Abschnitt auswählen können, den Sie drucken wollen. Weitere Informationen über dieses Dialogfeld finden Sie im Abschnitt *Drucken der Dokumentation* von Kapitel 5, *Drucken und Dokumentieren von VIs*.

Mausklicksequenzen bei Hierarchieknoten

Mausklicksequenzen stehen zum Auswählen von Knoten, Kopieren oder Ziehen von Knoten und für Shortcuts zu bestimmten Popup-Menüpunkten zur Verfügung. Sofern zutreffend, werden die Sequenzen bei gewähltem Positionierwerkzeug durchgeführt.

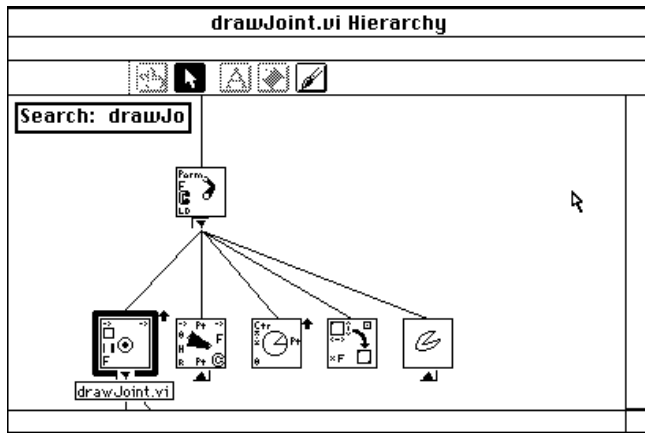
Die Shortcut-Mausklicks laufen wie folgt ab. Eine ausführlichere Beschreibung der im Popup-Menü verfügbaren Optionen finden Sie im Abschnitt *Popup-Menü von Hierarchieknoten* in diesem Kapitel.

- Durch Klicken auf die rote Pfeiltaste wird die Aktion “Direkte SubVIs anzeigen” ausgeführt.
- Durch gleichzeitiges Drücken der <Umschalttaste> und Klicken auf die rote oder schwarze Pfeiltaste wird die Aktion “Alle SubVIs anzeigen” ausgeführt.
- Durch Klicken und gleichzeitiges Drücken der <Strg-Taste> (**Windows**), <Optionstaste> (**Macintosh**) <Meta-Taste> (**Sun**) und <Alt-Taste> (**HP-UX**) auf dem Knoten wird die Aktion “VI-Hierarchie anzeigen” ausgeführt.
- Durch Doppelklicken auf den Knoten wird die Aktion “Frontpanel öffnen” ausgeführt.
- Durch einmaliges Klicken auf den Knoten wird dieser ausgewählt, um ihn zur Verwendung als SubVI in ein Blockdiagramm zu ziehen oder in die Zwischenablage zu kopieren.
- Durch Klicken und Halten der Maustaste bei gedrückter <Umschalttaste> können Sie mehrere Knoten auswählen, um sie in andere Blockdiagramme oder Frontpanels zu kopieren. Durch Ziehen eines Rechtecks um die zu wählenden Objekte können Sie auch mehrere Objekte auf einmal auswählen.
- Durch Drücken der <Tab>-Taste wird zwischen Positionswerkzeug und Bildlaufwerkzeug der Palette **Werkzeuge** umgeschaltet.

Suchen von VIs im Hierarchiefenster

Sie können die Suche einleiten, indem Sie einfach den Namen des Knotens eingeben. Ein kleines Fenster erscheint, in dem der eingegebene Text angezeigt wird. Die Suche erfolgt sofort, und ein entsprechender Knoten wird durch Anzeigen eines Tip-Strips seines Namens hervorgehoben.

In der nachfolgenden Abbildung wurde durch den Mechanismus “Hierarchieknoten suchen” der Knoten mit dem Namen `drawJoint.vi` gefunden.



Während Sie die Eingabe vornehmen, wird eine Suche durchgeführt; wenn daher kein Knoten den Zeichen entspricht, die gegenwärtig im Suchfenster angezeigt werden, ertönt ein Signalton, und es können keine weiteren Zeichen eingegeben werden. Sie können dann die <Rücktaste> oder <ENTF>-Taste benutzen, um ein oder mehrere Zeichen zu löschen, und die Eingabe fortsetzen.

Das Suchfenster wird automatisch ausgeblendet, wenn für einen bestimmten Zeitraum keine Tasten gedrückt werden. Mit Hilfe der <ESC>-Taste können Sie das Suchfenster gleich ausblenden.

Wird ein entsprechender Knoten gefunden, können Sie mit der Nach-rechts- oder Nach-unten-Pfeiltaste oder der <Eingabe>-Taste (**Windows** und **Sun**) oder der <Return>-Taste (**Macintosh** und **HP-UX**) den nächsten Knoten mit dem entsprechenden Suchstring suchen. Um den vorherigen Knoten mit der Entsprechung zu suchen, drücken Sie die Nach-links- oder nach-oben-Pfeiltaste oder <Umschalt-Eingabe> (**Windows** und **Sun**) oder <Umschalt-Return> (**Macintosh** und **HP-UX**).

Suchen von VIs, Objekten und Text

Beim Entwickeln einer Anwendung, die aus mehreren SubVIs oder auch einem einzigen, umfangreichen VI besteht, wollen Sie möglicherweise Fälle eines bestimmten Objekts oder Textstrings suchen. Der Befehl **Projekt»Suchen** kann Ihnen helfen, alle Instanzen der folgenden Objekte nach den von Ihnen angegebenen Namen zu suchen:

- VIs
- Implementierte Funktionen
- Typendefinitionen
- Globale Variable
- Lokale Variable
- Attributknoten
- Breakpoints
- Frontpanel-Terminals
- Text

Informationen über Typendefinitionen finden Sie im Abschnitt *Typendefinitionen* in Kapitel 24, *Benutzerspezifische Bedienelemente und Typendefinitionen*. Informationen über lokale und globale Variable finden Sie in Kapitel 23, *Globale und lokale Variablen*. Informationen über Attributknoten finden Sie in Kapitel 22, *Attributknoten*. Informationen über Breakpoints finden Sie im Abschnitt *Plazieren von Breakpoint-Werkzeugen* in Kapitel 4, *Ausführung und Debugging von VIs und SubVIs*.

Abgesehen vom Befehl **Suchen** verfügen viele Objekte über Popup-Menüpunkte, die Ihnen helfen können, schnell verwandte Objekte zu suchen. Wenn Sie beispielsweise das Popup-Menü eines Bedienelements wählen, hilft Ihnen ein Menüpunkt, das entsprechende Terminal sowie damit verknüpfte lokale Variable oder Attributknoten zu suchen.

Das Dialogfeld “Suchen”

Um das Dialogfeld “Suchen” zu aktivieren, wählen Sie **Projekt»Suchen...**, oder drücken Sie <Strg-f> (**Windows**), <Befehlstaste-f> (**Macintosh**), <Metataste-f> (**Sun**) oder <Alt-f> (**HP-UX**).

Als Shortcut können Sie ein Textstück oder ein Objekt auswählen, bevor Sie auf das Dialogfeld zugreifen. Im Dialogfeld wird der Text oder das Objekt dann für die Suche bereits ausgewählt.

Suchen von VIs und anderen Objekten

Um VIs oder andere Objekte zu suchen, klicken Sie im Dialogfeld “Suchen” auf die Taste **Objekte**.



Um das Objekt zu wählen, nach dem Sie suchen wollen, klicken Sie auf die Taste, die dem Ausdruck **Objekt wählen** folgt. Das Menü **Objekt wählen** erscheint.



In diesem Menü sind die folgenden Optionen verfügbar:

- **Funktionen** — Aktiviert die Palette **Funktionen**, damit Sie integrierte Funktionen, VIs oder sonstige von Ihnen angepasste Objekte, die in der Palette **Funktionen** erscheinen, auswählen können.
- **VIs** — Zeigt eine Palette aller VIs an, die sich im Arbeitsspeicher befinden, jedoch nicht in `vi.lib` gespeichert sind.
- **Type Defs** — Zeigt eine Palette aller Typendefinitionen an, die sich im Arbeitsspeicher befinden, jedoch nicht in `vi.lib` gespeichert sind.
- **Globale Variablen** — Zeigt eine Palette aller globalen VIs an, die sich im Arbeitsspeicher befinden, jedoch nicht in `vi.lib` gespeichert sind.
- **Objekte in der vi.lib** — Zeigt alle VIs, Typendefinitionen und globale Variablen an, die in `vi.lib` gespeichert sind.
- **Andere** — Zeigt ein Menü an, aus dem Sie Attributknoten, Breakpoints und Frontpanel-Terminals wählen können.
- **VIs nach Namen...** — Ruft das Dialogfeld “VIs nach Namen auswählen” auf, das alle VIs im Arbeitsspeicher in alphabetischer Reihenfolge anzeigt. Sie können den Namen des Objekts eingeben, um schnell zum gewünschten Objekt in der Liste zu springen. Mit den drei Kontrollkästchen (**VIs**, **Globale Variablen** und **Type Defs**) können Sie verschiedene Objekttypen im Listenfeld anzeigen oder von der Anzeige ausschließen.

Suchen von Text

Um einen bestimmten Text zu suchen, klicken Sie im Dialogfeld “Suchen” auf die Taste **Text**.



Optionen für Suchstrings

Sie können den zu suchenden Text eingeben und wahlweise den Geltungsbereich der Suche auf bestimmte Bereiche von VIs (Frontpanel, Blockdiagramm, angezeigte VI-Informationen oder Historieninformationen) oder Komponenten von Objekten beschränken, wie z.B. das Label oder die Beschreibung.

Zur Angabe des Suchstrings stehen die folgenden Menüpunkte zur Verfügung:

- **Groß-/Kleinschreibung beachten** — Sucht nur Textstrings, deren Schreibweise (Groß-/Kleinschreibung) genau der der eingegebenen Zeichen entspricht.
- **Ganzes Wort vergleichen** — Sucht Textstrings nur dann, wenn ihnen ein nicht alphanumerisches Zeichen, wie z.B. ein Leerzeichen oder Pluszeichen (+) vorausgeht und folgt oder wenn sie am Zeilenanfang oder -ende stehen. Wird dieser Menüpunkt nicht gewählt, vergleicht der Befehl jeden String unabhängig davon, ob es sich um ein Bruchteil eines größeren Strings handelt oder nicht.
- **Gültiger Ausdruck** — Behandelt den Zeichen-String als einen gültigen Ausdruck. Der gültige Ausdruck besitzt dieselben Spezifikationen wie die Funktion "Pattern vergleichen" (Spezifikationen siehe **Online-Hilfe** » **Funktion und VI-Referenz**).

Textsuchoptionen

Die Taste **Mehr Optionen...** ruft das Dialogfeld "Optionen Textsuche" auf, das Menüpunkte zum Konfigurieren des Geltungsbereichs der Textsuche für jedes VI enthält.



Die Menüpunkte zeigen an, ob und wo sichtbarer oder verborgener Text zu suchen ist. Sie lassen sich wie folgt beschreiben:

- **In VIs suchen**— Legt fest, ob auf dem Frontpanel eines VIs, in einem Blockdiagramm, im Bereich “Infotext holen” oder im Text des Historie-Fensters zu suchen ist. Mindestens eine dieser Optionen muß gewählt sein.
- **Im Label des Objekts suchen** — Legt fest, ob in sichtbaren oder verborgenen Namen-Labels von Objekten gesucht werden soll.
- **In Objektdaten & -elementen suchen** — Legt fest, ob in sichtbaren oder verborgenen Daten und Teilen zu suchen ist. Die Daten und Teile eines Objekts bestehen aus allen zu einem Objekte gehörenden Elementen mit Ausnahme des Namen-Labels des Objekts. Es besteht auch die Möglichkeit, nach der Beschreibung verschiedener Objekte zu suchen. Mindestens einer der Menüpunkte unter **Im Label des Objekts suchen** und **In Objektdaten und -elementen suchen** muß gewählt sein.



Hinweis

Die Textsuchoptionen werden nach jeder Suche gespeichert; wenn Sie daher einen Text-String nicht finden, dessen Existenz zu erwarten ist, vergewissern Sie sich, daß die Textsuchoptionen richtig eingestellt sind.

Eingrenzen des Suchbereichs

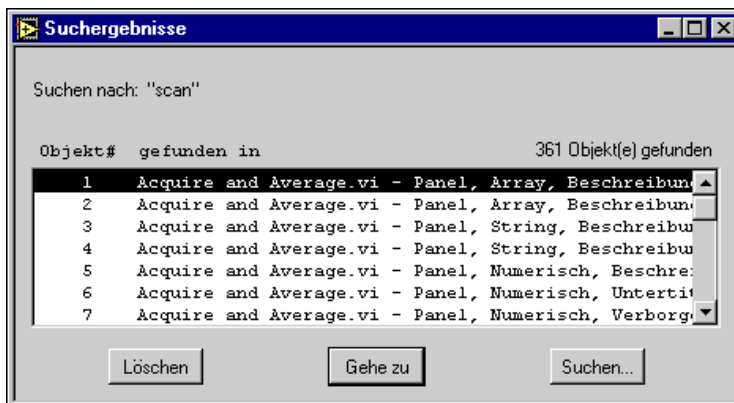
Im Ringmenü **Suchbereich** werden die zu durchsuchenden VIs durch drei Optionen festgelegt:

- **Alle VIs im Speicher** — Legt den Suchbereich auf alle gegenwärtig gewählten VIs fest. Zwei Kontrollkästchen stehen zur Verfügung, mit denen VIs in `vi.lib` und im Hierarchiefenster in die Suche miteinbezogen oder davon ausgeschlossen werden können.
- **Ausgewählte VIs** — Wählt einen benutzerspezifischen Satz von VIs zur Suche aus. Das Hierarchiefenster kann hierbei in den Suchbereich miteinbezogen oder davon ausgeschlossen werden. Die Taste **Auswählen...** ruft den Menüpunkt **VIs auswählen** im Suchdialogfeld auf.
- **Name des VI** — Sucht nur in einem bestimmten VI. Der VI-Name in dieser Ringoption hängt davon ab, welches VI beim Aufrufen des Suchdialogfeldes aktiv war.

Klicken Sie auf die Taste **Suchen**, um die Suche nach dem gewählten Text zu beginnen.

Das Fenster “Suchergebnisse”

Nach Abschluß einer Suche werden alle Ergebnisse der Suche (sofern mehrere Suchergebnisse vorliegen) im Fenster “Suchergebnisse” angezeigt (siehe Abbildung unten). Wird nur ein Ergebnis gefunden, wird dieses unter Umgehung des Fensters “Suchergebnisse” sofort hervorgehoben.



Dieses Fenster wird auch durch Auswählen von **Projekt» Suchergebnisse...** angezeigt.

Im Fenster “Suchergebnisse” stehen die folgenden Menüpunkte zur Verfügung:

- **Löschen** — Löscht und gibt Arbeitsspeicher frei, der zur Speicherung von Suchergebnissen benutzt wird.
- **Gehe zu** — Hebt das gegenwärtig gewählte Suchergebnis hervor. Dies ist auch durch Doppelklicken auf die Option möglich. Bereits hervorgehobene Optionen werden überprüft.
- **Suchen...** — Ruft das Dialogfeld “Suchen” auf.
- **Stopp** — Ersetzt die Aufschrift **Suchen...** der Taste während einer Suche. Bei einer langen Suche in vielen VIs werden die Suchergebnisse nach und nach aktualisiert, und der Status der Suche wird angezeigt. Mit der Taste **Stopp** kann die Suche unterbrochen werden.

Wenn Sie ein Objekt oder VI, das Bestandteil eines Suchergebnisses ist, löschen, wird der zugehörige Eintrag im Ergebnisfenster deaktiviert. Beachten Sie jedoch, daß die Suchergebnisse bei Textänderungen nicht aktualisiert werden. Es ist daher denkbar, daß Sie ein Textsuchergebnis hervorheben, das nicht mehr die Suchkriterien erfüllt. Führen Sie in diesem Fall eine neue Suche durch, um die Suchergebnisse zu aktualisieren.

Suchen des nächsten und vorherigen Suchobjekts

Wählen Sie **Projekt»Weitersuchen**, um das nächste bzw. vorherige Ergebnis in der Suchliste hervorzuheben. Bei verborgenem Text werden alle Objekte, die zum Anzeigen des Texts erforderlich sind, kurzfristig als nicht bedienbar markiert, und der Text wird ausgewählt. Wenn Sie die Maustaste oder eine Taste drücken, werden die Objekte wieder verborgen.

Popup-Menü “Suchen” für globale und lokale Variablen und Attributknoten

Bei Bedien- und Anzeigeelementen, globalen Variablen, lokalen Variablen und Attributknoten von Frontpanels ist ein Popup-Menüpunkt “Suchen” verfügbar. Sie können das Popup-Menü einer lokalen Variable oder eines Attributknoten aufrufen, um das zugehörige Bedienelement, das Frontpanel-Terminal oder andere lokale Variablen und Attributknoten zu suchen. Umgekehrt besitzen Bedien- und Anzeigeelemente von Frontpanels Popup-Menüs, um alle entsprechenden lokalen Variablen und Attributknoten zu suchen. Von einem globalen Knoten aus können Sie dessen entsprechende globale Definition oder alle anderen globalen Knoten suchen. Umgekehrt besitzen globale Bedienelemente ein Popup-Menü, um alle entsprechenden globalen Bezüge im Arbeitsspeicher zu suchen. Werden mehrere Ergebnisse ermittelt, wird das Fenster “Suchergebnisse” angezeigt.

Ausführung und Debugging von VIs und SubVIs

In diesem Kapitel werden Bedienung und Debugging von VIs beschrieben und die Einrichtung von VIs und SubVIs für besondere Ausführungsmodi erklärt.

Ausführung von VIs

Ausführen von VIs



Ausführen

Ein VI kann ausgeführt werden, indem Sie **Ausführen»Ausführen** auswählen oder auf die **Ausführen**-Taste klicken. G kompiliert das VI gegebenenfalls.



VI-Ausführung
auf höchster Ebene

Während der VI-Ausführung auf höchster Ebene (d.h. es hat keine Aufrufer und ist daher kein SubVI) ändert sich die **Ausführen**-Taste, so daß sie folgendermaßen aussieht.



VI-Aufrufer
wird ausgeführt

Wenn das VI als SubVI ausgeführt wird, ändert sich die **Ausführen**-Taste, so daß sie folgendermaßen aussieht.



Wiederholt
Ausführen

Zur erneuten ständig wiederholten Ausführung des VIs die Taste **Wiederholt ausführen** drücken, bis der Vorgang abgebrochen oder die Ausführung angehalten wird.



Abbrechen

Durch das Drücken der **Abbrechen**-Taste wird die Ausführung des VIs der *höchsten Ebene* abgebrochen. Wenn ein VI von mehr als einem ausgeführten VI der höchsten Ebene verwendet wird, ist die Taste als nicht bedienbar markiert.



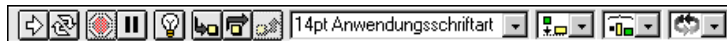
Anhalten

Durch das Drücken dieser Taste wird die Ausführung angehalten. Die Taste kann jederzeit erneut gedrückt werden, um mit der Ausführung fortzufahren.

Es können mehrere VIs zur gleichen Zeit ausgeführt werden. Nachdem das erste VI gestartet wurde, zum Frontpanel oder Blockdiagramm des nächsten umschalten und dieses VI wie zuvor beschrieben starten. Beachten Sie, daß bei der Ausführung eines SubVIs als VI der höchsten Ebene alle Aufrufer-VIs so lange nicht ausführbar sind, bis das SubVI abgeschlossen wird. Ein SubVI kann nicht zur gleichen Zeit als VI der höchsten Ebene und als SubVI ausgeführt werden.

Wenn das VI ausgeführt wird, jedoch nicht richtig funktioniert, siehe die Abschnitte *Debugging von nicht ausführbaren VIs* und *Debugging von ausführbaren VIs* weiter hinten in diesem Kapitel.

Wenn ein VI bearbeitet wird, enthält die Werkzeugpalette mehrere Werkzeuge, die zum Bearbeiten von VIs eingesetzt werden, darunter den Fontring, den Ausrichtungsring, den Verteilungsring und den Neuordnungsring.

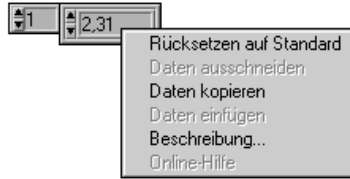


Wenn ein VI ausgeführt wird, werden diese Werkzeuge durch Debugging-Werkzeuge ersetzt (siehe nachstehende Abbildung).

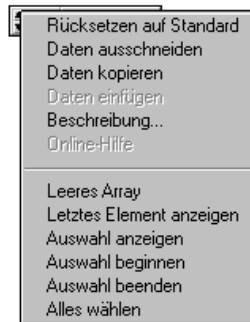


Ausführen

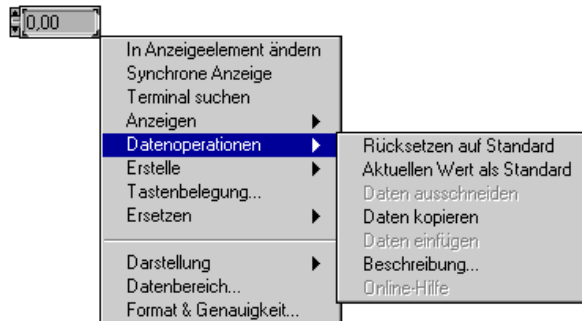
Wenn Sie während der Ausführung eines VIs das Popup-Menü eines Bedienelements aufrufen, können Sie feststellen, daß alle Objekte über einen wesentlich einfacheren Satz von Menüpunkten verfügen. Dabei handelt es sich um dieselben Menüpunkte wie im Untermenü **Datenoperationen**, das bei der Bearbeitung von VIs in Popup-Menüs erscheint.



Mit den Menüpunkten in diesem Menü können Sie den Inhalt der Bedienelemente ausschneiden, kopieren oder einfügen, das Bedienelement auf seinen Standardwert einstellen und die Beschreibung des Bedienelements lesen. Einige der komplizierteren Bedienelemente verfügen über zusätzliche Optionen. Ein Array hat z.B. Menüpunkte zum Kopieren eines Wertebereiches oder zum Vorrücken zum letzten Element im Array.



Bei der Bearbeitung eines VIs enthält das Popup-Menü eines Objektes ein Bearbeitungsменю und ein Untermenü **Datenoperationen** (siehe nachstehende Abbildung). Dieses Untermenü enthält dieselben Menüpunkte wie bei der Ausführung eines VIs und darüber hinaus den zusätzlichen Menüpunkt **Aktuellen Wert als Standard**.



Das Menü **Ausführen** auf der Menüleiste enthält Befehle zur Ausführung des aktuellen VIs. In den folgenden Abschnitten werden mehrere wichtige Tasks behandelt, die sich auf die Ausführung beziehen.

Anhalten von VIs



Abbrechen

Unter normalen Umständen wird ein VI ausgeführt, bis es abgeschlossen ist. Wenn Sie jedoch die Ausführung sofort anhalten müssen, können Sie auf die **Abbrechen**-Taste klicken oder **Ausführen»Stopp** auswählen. Mit dem **Abbrechen**-Befehl wird das VI der höchsten Ebene zum frühestmöglichen Zeitpunkt angehalten. Das angehaltene VI hat mit großer Wahrscheinlichkeit seine Aufgabe noch nicht abgeschlossen; Sie können daher keine der von ihm erzeugten Daten verwenden. G schließt zwar alle Dateien, die zu diesem Zeitpunkt offen sind, und hält alle Datenerfassungsvorgänge an, die eventuell ablaufen; vermeiden Sie jedoch die Verwendung von VIs, die auf die **Abbrechen**-Taste oder den Menüpunkt **Abbrechen** angewiesen sind. Wenn Sie z.B. ein VI erstellen, das innerhalb einer While-Schleife unendlich ausgeführt wird, bis der Bediener es anhält, verbinden Sie den Bedingungsanschluß der Schleife mit einem Booleschen Schalter auf dem Frontpanel.

Wenn der Bediener daran gehindert werden soll, das VI versehentlich durch Klicken auf die **Abbrechen**-Taste abzubrechen, kann die Taste verborgen werden, indem das Kontrollkästchen **VI Einstellungen» Fensteroptionen»Taste "Abbrechen" anzeigen** im Popup-Menü des Icon-Feldes auf dem Frontpanel des VIs ausgewählt wird. Weitere Informationen finden Sie im Abschnitt *Fensteroptionen* in Kapitel 6, *Einrichten von VIs und SubVIs*.

Wiederholte Ausführung von VIs



Wiederholt Ausführen

Zur wiederholten Ausführung eines VIs auf die Taste **Wiederholt ausführen** klicken. Das VI beginnt sofort mit der Ausführung, und die Taste **Wiederholt ausführen** ändert sich während der Ausführung des VIs von Pfeilumrissen auf gefüllte Pfeile. Erneut auf die Taste **Wiederholt ausführen** klicken, um das VI anzuhalten. Das VI hält an, wenn es normal abgeschlossen wird.



VI in wiederholter Ausführung

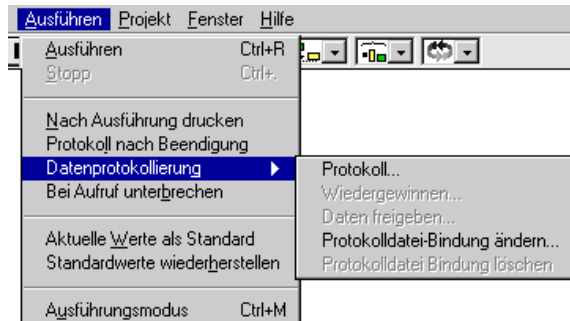
Das Verhalten des VIs und der Zustand der Werkzeugpalette während der wiederholten Ausführung ist mit dem Verhalten und Zustand während einer Einzelausführung zu vergleichen, die mit der **Ausführen**-Taste oder dem Befehl **Ausführen»Ausführen** gestartet wurde.

Datenprotokollierung auf dem Frontpanel

Die Frontpanel-Datenprotokollierung speichert die Daten in allen Frontpanel-Bedienelementen eines VIs mit einem Zeitstempel in einer Datenprotokolldatei. Sie können mehrere Dateien mit Protokolldaten von unterschiedlichen Tests anlegen. Diese Daten können mit Hilfe des Interactive Retrieval-Modus über das VI, mit Hilfe von programmatischen Datenabfragemethoden oder standardmäßigen Datei-I/O-Funktionen aufgerufen werden.

Jedes VI behält eine Bindung zur Protokolldatei bei, wobei die Speicherstelle derjenigen Datenprotokolldatei angezeigt wird, in der Frontpanel-Daten aufgezeichnet werden. Falls die Bindung bei aktivierter Datenprotokollierung leer ist, d.h. daß die Speicherstelle nicht festgelegt ist, fordert das VI Sie zur Eingabe der Speicherstelle der Datei auf.

Die Datenprotokollierung über das Frontpanel kann konfiguriert und gesteuert werden, indem das Menü **Ausführen** und das dazugehörige Untermenü **Datenprotokollierung** ausgewählt wird (siehe nachstehende Abbildung).



Wenn für ein VI die automatische Datenprotokollierung aktiviert ist, protokolliert das Frontpanel des VIs jedesmal, wenn das VI die Ausführung abschließt. Sie können feststellen, ob die automatische Datenprotokollierung für ein VI aktiviert ist, indem Sie den Menüpunkt **Ausführen»Protokoll nach Beendigung** aufrufen. Durch die Auswahl von **Ausführen»Protokoll nach Beendigung** wird die automatische Datenprotokollierung für ein VI aktiviert, wenn sie deaktiviert ist und umgekehrt.

Zum interaktiven Protokollieren von Frontpanel-Daten für ein VI den Menüpunkt **Ausführen»Datenprotokollierung»Protokoll...** auswählen.



Hinweis

Ein Kurvenformdiagramm liefert mit der Frontpanel-Datenprotokollierung nur einen Datenpunkt auf einmal. Wenn Sie ein Array mit der Diagrammanzeige verbinden, enthält das Datenprotokoll das im Diagramm angezeigte Array-Subset für diesen Datensatz.

Die Protokoll-Dateibindung eines VIs kann mit dem Menüpunkt **Ausführen»Datenprotokollierung»Log-Dateibindung ändern...** geändert werden.



Enter-Taste

Ein bestimmter Datensatz kann durch Eingabe der Datensatznummer und anschließendem Klicken auf die **Enter**-Taste oder Drücken der **<Eingabe>**-Taste auf dem numerischen Tastenfeld ausgewählt werden.



Löschen

Sie können den ausgewählten Datensatz zum Löschen auswählen, indem Sie auf die Taste **Löschen** klicken. Wenn der ausgewählte Datensatz zum Löschen markiert ist, wird aus der Taste **Löschen** ein voller Papierkorb. Durch Klicken auf den vollen Papierkorb wird der zum Löschen markierte Datensatz wieder abgewählt. Ausgewählte Datensätze werden erst dann gelöscht, wenn Sie **Ausführen»Datenprotokollierung»Daten freigeben...** auswählen oder wenn Sie vom Datenabfragemodus zurückschalten; dies geschieht entweder durch Klicken auf die **OK**-Taste oder durch erneutes Wählen von **Wiedergewinnen...** Wenn nach Zurückschalten aus dem Datenabfragemodus irgendeine Datensätze noch zum Löschen markiert sind, werden Sie gefragt, ob Sie die markierten Datensätze löschen möchten.

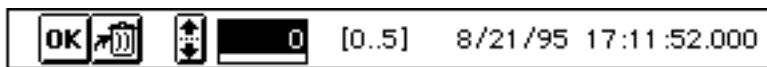


Zum Löschen markierter Datensatz

Zur Rückkehr zum VI, dessen Datenprotokoll Sie betrachten haben, die **OK**-Taste drücken.

Durch die Auswahl von **Ausführen»Bindung zu Log-Datei löschen** wird die Verknüpfung des VIs mit assoziierten Log-Dateien gelöst. Wenn Sie das nächste Mal Daten des VIs vom Frontpanel protokollieren, werden Sie zur Angabe der Log-Datei aufgefordert.

Zur interaktiven Ansicht der protokollierten Daten den Menüpunkt **Ausführen»Wiedergewinnen...** auswählen. Die Werkzeugpalette wird zur Datenabfrage-Werkzeugpalette (siehe nachstehende Abbildung).



Die hervorgehobene Zahl gibt an, welcher Datensatz momentan zur Ansicht ausgewählt ist. Die Zahlen in den eckigen Klammern geben den Bereich der protokollierten Datensätze an. Datum und Zeit geben an, wann der ausgewählte Datensatz protokolliert wurde. Der nächste bzw. vorherige

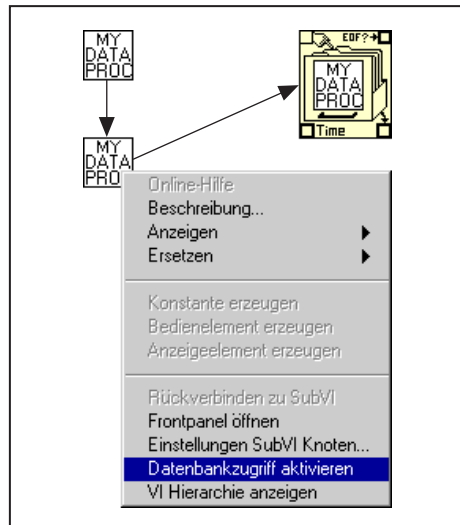
Datensatz kann betrachtet werden, indem Sie auf die Nach-Oben- bzw. Nach-Unten-Pfeiltaste klicken. Außerdem können Sie die Nach-Oben- oder Nach-Unten-Pfeiltasten auf Ihrer Tastatur verwenden.

Programmatische Datenabfrage

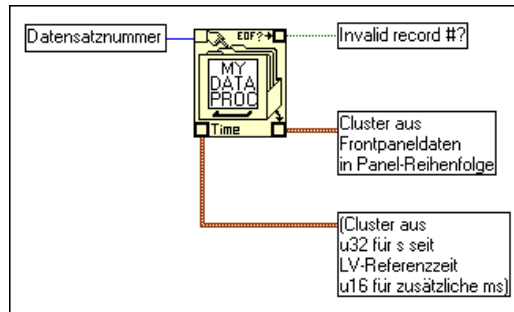
Von einem VI protokollierte Daten können wiedergewonnen werden, indem Sie den Menüpunkt **Datenbankzugriff aktivieren** oder standardmäßige Datei-I/O-Funktionen verwenden, um die Datei als Datenprotokoll-Datei zu lesen.

Zugreifen auf Datenbanken

Die Datenabfrage erfolgt über den Menüpunkt **Datenbankzugriff aktivieren**. Dies geschieht vom Popup-Menü eines SubVIs aus, das Frontpanel-Daten in der assoziierten Datenprotokoll-Datei des VIs protokolliert hat (siehe nachstehende Abbildung). Ein *Halo*, das wie ein Aktenschrank aussieht, erscheint um die Datenprotokoll-Datei. Dieses Halo verfügt über Terminals für den Zugriff auf Daten von der Datenprotokoll-Datei. Wenn das aufrufende Diagramm ausgeführt wird, kommt das SubVI nicht zur Ausführung. Statt dessen rufen Sie Daten von einem bestimmten Datensatz der Datenprotokoll-Datei ab. Außerdem wird die Zeit ausgegeben, zu der die Daten protokolliert wurden, sowie ein Boolescher Wert, der angibt, ob die angezeigte Datensatznummer ungültig ist.



Folgende Abbildung zeigt die Halo-Terminals zum Zugreifen auf die protokollierten Daten.



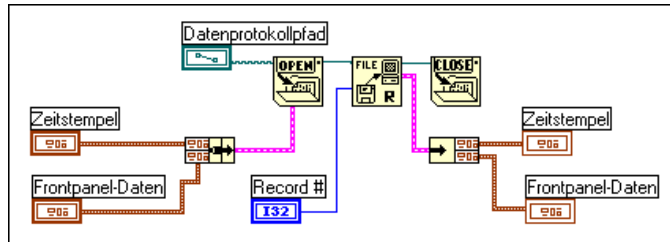
Wenn das SubVI über n protokollierte Datensätze verfügt, können Sie eine beliebige Anzahl von $-n$ bis $n - 1$ mit dem Terminal der Datensatznummer verbinden. Sie können mit Hilfe von nicht-negativen Datensatznummern auf Datensätze zugreifen, die relativ zum *ersten* protokollierten Datensatz sind. 0 stellt den ersten Datensatz dar, 1 stellt den zweiten Datensatz dar und so weiter, bis zu $n - 1$, was den letzten Datensatz repräsentiert.

Mit Hilfe von negativen Nummern können Sie auf Datensätze zugreifen, die relativ zum *letzten* protokollierten Datensatz sind. Denken Sie daran, daß -1 den letzten Datensatz darstellt, -2 den zweitletzten und so weiter, bis zu $-n$, was den ersten Datensatz repräsentiert. Wenn Sie eine Nummer außerhalb des Bereichs $-n$ bis $n - 1$ mit dem Datensatznummer-Terminal verbinden, wird der Ausgang Invalid record #? auf TRUE gesetzt, und es werden keine Daten abgerufen.

Abrufen von Daten mit Datei-I/O-Funktionen

Daten, die vom Frontpanel eines VIs protokolliert wurden, können auch mit Hilfe der Datei-I/O-Funktionen wiedergewonnen werden, die in G integriert sind. Jeder Datensatz in der Datenprotokoll-Datei, der durch die Datenprotokollierung vom Frontpanel erstellt wurde, ist ein Cluster mit einem Zeitstempel und den Frontpanel-Daten. Der Cluster mit dem Zeitstempel besteht aus einem vorzeichenlosen 32-Bit-Integer-Wert, der Sekunden darstellt, und einem vorzeichenlosen 16-Bit-Integer-Wert, der Millisekunden darstellt, die seit der G-Referenzzeit verstrichen sind. Auf diesen Cluster folgt ein weiterer Cluster der Frontpanel-Daten in Panelreihenfolge.

Auf die Datensätze der Datenprotokoll-Dateien, die mit Hilfe der Datenprotokollierung durch das Frontpanel erstellt wurden, kann mit denselben I/O-Funktionen der G-Datei zugegriffen werden, die für den Zugriff auf durch Programmsteuerung erstellte Datenprotokoll-Dateien verwendet werden. Wie zuvor beschrieben, den entsprechenden Typ als Eingabetyp für die Funktion "Datei öffnen" eingeben, wie im folgenden Beispiel dargestellt.



Debugging von VIs

In diesem Abschnitt werden eine Reihe von Lösungen für Fehler und Probleme in virtuellen Instrumenten behandelt.

Debugging von nicht ausführbaren VIs

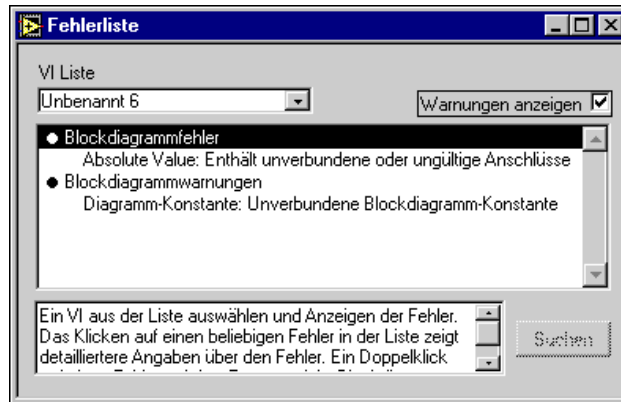
Ein VI kann nicht kompiliert oder ausgeführt werden, wenn es Störungen aufweist. Die Taste **VI Ausführen** erscheint normalerweise unterbrochen, während Sie das VI erstellen oder bearbeiten, bis alle Icons im Diagramm verbunden sind. Wenn sie nach Abschluß der Verbindungen der Icons immer noch unterbrochen erscheint, wählen Sie **Bearbeiten»Ungültige Verbindungen entfernen** oder <Strg-b> (**Windows**); <Befehlstaste-b> (**Macintosh**); <Meta-b> (**Sun**) oder <Alt-b> (**HP-UX**). Damit kann ein gestörtes VI häufig wiederhergestellt werden.



Gestörte
Ausführung

Um festzustellen, warum ein VI gestört ist, auf die Taste **Gestörte Ausführung** klicken. Es erscheint ein Fehlerlisten-Fenster, in dem alle

Fehler aufgeführt sind. Dieses Fenster wird in folgender Abbildung dargestellt.



Warnung

Auf dieses Feld kann auch durch Klicken auf die **Warnungs**-Taste eines VIs, oder durch Auswahl von **Fenster»Fehlerliste anzeigen** zugegriffen werden. Die **Warnungs**-Taste für ein VI ist nur dann sichtbar, wenn das VI über eine Warnung verfügt (wie für überlappende Objekte) und Sie **Warnungen anzeigen** im Fehlerlisten-Fenster markieren. Wenn G so konfiguriert werden soll, daß standardmäßig Warnungen angezeigt werden, verwenden Sie das Dialogfeld **Voreinstellungen**.

Zum Auffinden eines Fehlers auf den Text klicken, der ihn beschreibt. Der Fehler wird angezeigt, indem das relevante Fenster in der Vordergrund geholt und das Objekt, das den Fehler verursacht, hervorgehoben wird. Fehler und Warnungen für andere VIs können betrachtet werden, indem Sie den Namen des VIs aus dem Popup-Menü **VI Liste** wählen.

Die folgende Liste enthält einige der häufigsten Ursachen für Störungen eines VIs während der Bearbeitung.

- Ein Funktionsterminal, das einen Eingang erfordert, ist unverbunden. Sie müssen z.B. alle Eingänge mit arithmetischen Funktionen verbinden. Es dürfen keine Funktionen im Diagramm unverbunden bleiben, während das VI zum Testen verschiedener Designs ausgeführt wird.
- Das Blockdiagramm enthält eine unterbrochene Verbindung, da ein Konflikt der Datentypen vorliegt oder ein Ende lose oder unverbunden ist. Unterbrochene Verbindungen einschließlich der Verbindungsstümpfe, die nicht sichtbar sind, mit dem Menüpunkt **Bearbeiten»Ungültige Verbindungen entfernen** entfernen.

- Ein SubVI ist gestört oder der Anschluß wurde bearbeitet, nachdem das entsprechende Icon in das Diagramm plazierte wurde.
- Es liegt eventuell ein Problem mit einem Objekt vor, das mit einem Attributknoten unsichtbar gemacht, deaktiviert oder anderweitig abgeändert wurde. Das Objekt, wenn möglich, mit dem Attributknoten wiederherstellen, um das Problem zu beheben.

Auslegen von Fehlermeldungen

Folgende Liste enthält einige mögliche Fehler in einer Form, die der im Fehlerlistenfenster gleicht.

Tabelle 4-1. Fehlermeldungen

Fehlermeldungen	Beschreibung
Function Name: contains unwired or bad terminal.	Ein erforderlicher Eingang ist nicht verbunden oder der Typ ist ungültig. Die erforderlichen Eingänge mit den richtigen Datentypen verbinden.
Code Interface Node: object code not loaded.	Objektcode des CIN wurde nicht ordnungsgemäß verknüpft. Das Popup-Menü des Knotens aufrufen und den Code erneut laden.
Control: control does not match its type definition.	Bedienelement entsprechend bearbeiten oder das Popup-Menü des Bedienelements aufrufen und es aktualisieren oder das VI mit der Typendefinition vom Bedienelement trennen.
Enumeration has duplicate entries.	Alle Elemente in einem Aufzählungs-Bedienelement müssen eindeutig und ausgeprägt sein.
More Errors....	Das Fehlerfenster zeigt nur 100 Fehler zur gleichen Zeit. Nicht in der Liste enthaltenen Fehler erscheinen, nachdem die aufgeführten Fehler behoben wurden.
For Loop: N is unwired and there are no input indexing tunnels.	Nur wenn N verbunden oder ein Array am Eingang indiziert ist, kann die Schleife feststellen, wieviele Iterationen ausgeführt werden sollen.

Tabelle 4-1. Fehlermeldungen (Fortsetzung)

Fehlermeldungen	Beschreibung
Global or Local Variable: named component doesn't exist.	Der Name einer Variable hat sich seit dem letzten Laden des VIs geändert, und der Name in der globalen oder lokalen Variable stimmt nicht mehr überein. Die Variable bei festgehaltener Befehlstaste anklicken und einen anderen Namen auswählen.
Global Variable: subVI is missing.	Beim Laden des aufrufenden VIs wurde das globale SubVI nicht gefunden, weil Sie vielleicht den Namen geändert haben. Das ungültige globale SubVI mit einem gültigen SubVI ersetzen.
Node: A subroutine priority VI cannot contain an asynchronous node.	Eine asynchrone Funktion wie z.B. Warten kann nicht in einem VI mit einer Prioritätsstufe verwendet werden, die einer Subroutine entspricht.
Right Shift Register: type is undefined.	Unbenutzte Schieberegister müssen entfernt werden.
Right Shift Register: some but not all left sides are wired.	Ein Schieberegister muß Eingänge entweder für alle linken Seiten oder für keine haben.
Sequence: One or more sequence locals were never assigned.	Sie haben vergessen, eine Verbindung zu einer lokalen Variablen der Sequenzstruktur herzustellen. Unbenutzte Sequenzvariablen entfernen.
SubVI name: bad linkage to subVI.	Das Anschlußmuster des SubVIs hat sich seit dem letzten Laden des SubVIs geändert, oder Sie haben vergessen, Bedien- oder Anzeigeelemente des SubVIs den Terminals zuzuweisen. Trifft letzteres zu, den Befehl Rückverbinden im Popup-Menü des SubVIs verwenden.
SubVI name: recursive references (dispose it).	Der Name des aufrufenden VIs wurde geändert, so daß er dem Namen eines seiner SubVIs entspricht, und nun registriert G diese Änderung in der Weise, daß dieses VI sich selbst rekursiv aufruft. (Das VI und SubVI befinden sich wahrscheinlich in unterschiedlichen Verzeichnissen.) G verhindert die Rekursion, wenn Sie versuchen, ein VI auf das eigene Blockdiagramm zu plazieren.
SubVI name: LV Subroutine link error.	Bei der Verknüpfung mit einer CIN-Routine ist ein Problem aufgetreten.

Tabelle 4-1. Fehlermeldungen (Fortsetzung)

Fehlermeldungen	Beschreibung
SubVI name: A Subroutine priority VI cannot call a non-subroutine priority subVI.	Das SubVI muß mit Subroutinenpriorität ausgeführt werden, oder das aufrufende VI muß eine andere Priorität als Subroutinenpriorität erhalten.
SubVI name: subVI is already running.	Ein VI kann nicht ausgeführt werden, wenn eines seiner SubVIs bereits als SubVI eines anderen VIs ausgeführt wird.
SubVI name: subVI is in either panel order or cluster order mode.	Ein VI kann nicht ausgeführt werden, wenn die Panel- oder Clusterreihenfolge eines seiner SubVIs geändert wird.
SubVI name: subVI is in interactive retrieval mode.	Ein VI kann nicht ausgeführt werden, wenn eines seiner SubVIs sich im interaktiven Abrufmodus befindet.
SubVI name: subVI is missing.	Beim Laden des aufrufenden VIs wurde das SubVI nicht gefunden, weil Sie vielleicht den Namen geändert haben. Das ungültige SubVI durch ein gültiges ersetzen oder das fehlende SubVI öffnen, wenn Sie es finden können.
SubVI name: subVI is not executable.	Das SubVI ist gestört. Das Sub-IV öffnen und die Fehler beheben.
Terminal: The associated array or cluster on the front panel has no elements; its type is undefined.	Sie müssen ein Bedien- oder Anzeigeelement in das Array oder den Cluster plazieren.
Type Definition: can't find valid type definition.	Das VI zur Typendefinition wurde beim Laden des aufrufenden VIs nicht gefunden, weil Sie vielleicht den Namen geändert haben. Das fehlende VI zur Typendefinition öffnen, wenn Sie es finden, das Popup-Menü des ungültigen Bedienelements aufrufen und es von der Typendefinition trennen oder es durch ein gültiges Bedienelement ersetzen.

Tabelle 4-1. Fehlermeldungen (Fortsetzung)

Fehlermeldungen	Beschreibung
(Un) bundle By Name: Empty cluster, or some components are unnamed.	Der Eingangscluster, der mit der Funktion Nach Namen oder Nach Namen aufschlüsseln verbunden ist, ist entweder leer, oder einige Komponenten sind nicht beschriftet.
Unit: bad unit syntax.	Der Text im Knoten ist kein gültiger Einheitsausdruck. Ein ? wird unmittelbar vor das nicht erkennbare Zeichen gesetzt.

Wenn Sie Meldungen erhalten, die nicht selbsterklärend und nicht in diesem Kapitel aufgeführt sind, wenden Sie sich an National Instruments.

Debugging von ausführbaren VIs

Wenn Ihr Programm ausgeführt wird, jedoch nicht die erwarteten Ergebnisse liefert, können die Probleme häufig durch folgende Maßnahmen behoben werden.

- Eliminieren Sie alle VI-Warnungen. Wenn Sie **Warnungen anzeigen** im Fehlerlistenfenster ausgewählt haben, werden die Warnungen für Ihr VI im Listefeld identifiziert. Stellen Sie die Ursachen fest und eliminieren Sie sie aus Ihrem VI.
- Überprüfen Sie die Verbindungspfade, um sicherzustellen, daß die Verbindungen an die richtigen Terminals angeschlossen sind. Durch dreimaliges Klicken auf die Verbindung mit dem Bedienwerkzeug wird der gesamte Pfad hervorgehoben. Es ist möglich, daß eine Verbindung, die von einem Terminal zu stammen scheint, in Wirklichkeit von einem anderen stammt. Achten Sie genau darauf, wo das Ende der Verbindung am Knoten angeschlossen ist.
- Mit dem Hilfe-Fenster (**Hilfe»Hilfe anzeigen**) können Sie sicherstellen, daß Funktionen richtig verbunden sind.
- Überprüfen Sie, ob der Standardwert für unverbundene Eingänge von Funktionen oder SubVIs den Anforderungen entspricht.
- Verwenden Sie Breakpoints, Highlight-Funktionen und Einzelschrittausführung zur Feststellung, ob ein VI wie geplant ausgeführt wird.

- Mit der Probe-Funktion, die im Abschnitt *Verwendung des Probe-Werkzeugs* dieses Kapitels beschrieben ist, die unmittelbaren Datenwerte beobachten. Außerdem die Fehlerausgabe von Funktionen und SubVIs überprüfen, besonders diejenigen, die I/O-Funktionen ausüben.
- Beobachten Sie das Verhalten des VIs oder SubVIs mit verschiedenen Eingabewerten. Für numerische Bedienelemente mit Fließkomma die Werte NaN und $\pm\text{Inf}$ zusätzlich zu den normalen Werten eingeben.
- Wenn Ihr VI langsamer als erwartet ausgeführt wird, sicherstellen, daß Highlight-Funktionen in SubVIs ausgeschaltet ist. Außerdem die SubVI-Fenster schließen, wenn sie nicht benötigt werden.
- Überprüfen Sie die Darstellung Ihrer Bedien- und Anzeigeelemente, um festzustellen, ob Sie einen Überlauf erhalten, weil Sie eine Fließkommazahl in einen Integer-Wert oder einen Integer-Wert in einen kleineren Integer-Wert umgewandelt haben. Siehe auch den Abschnitt *Erkennen undefinierter Daten* in diesem Kapitel.
- Überprüfen Sie den Datenbereich und das Bereichsfehlerverhalten von Bedien- und Anzeigeelementen. Sie weisen unter Umständen nicht das gewünschte Fehlerverhalten auf.
- Stellen Sie fest, ob irgendwelche FOR-Schleifen unbeabsichtigterweise Null-Iterationen ausführen und leere Arrays produzieren. Siehe auch die Abschnitte *Highlight-Funktion* und *Erkennen undefinierter Daten* in diesem Kapitel.
- Überprüfen Sie, ob Sie Schieberegister ordnungsgemäß initialisiert haben, es sei denn, sie sollen absichtlich Daten von einer Ausführung der Schleife zur nächsten speichern.
- Überprüfen Sie die Reihenfolge von Clusterelementen an Quell- und Zielpunkten. G stellt zwar Konflikte zwischen Datentyp und Clustergröße zum Bearbeitungszeitpunkt fest, es erkennt jedoch keine Konflikte zwischen Elementen desselben Typs. Mit dem Menüpunkt **Cluster-Anordnung ...** auf dem Popup-Menü der Cluster-Shell die Clusterreihenfolge überprüfen.
- Überprüfen Sie die Knotenausführungsreihenfolge. Knoten, die nicht durch eine Verbindung angeschlossen sind, können in beliebiger Reihenfolge ausgeführt werden. Die räumliche Anordnung dieser Knoten hat keinen Einfluß auf die Reihenfolge. Das heißt, daß unverbundene Knoten *nicht* von links nach rechts und von oben nach unten im Diagramm ausgeführt werden, wie es bei Anweisungen in textbasierten Sprachen der Fall ist.

- Suchen Sie eventuell vorhandene externe VIs. Im Gegensatz zu Funktionen erzeugen unverbundene SubVIs nicht immer Fehler (es sei denn, ein Eingang wird wie erfordert oder empfohlen entsprechend konfiguriert). Wenn Sie versehentlich ein unverbundenes SubVI auf das Blockdiagramm plazieren, wird es bei der Diagrammausführung ebenfalls ausgeführt; infolgedessen kommt es in Ihrem Programm eventuell zur Durchführung zusätzlicher Vorgänge.
- Stellen Sie sicher, daß keine verborgenen VIs vorhanden sind. SubVIs können versehentlich auf drei Arten versteckt werden—durch Ablegen eines SubVIs direkt auf einem anderen Knoten; durch Verkleinern der Größe einer Struktur, ohne das SubVI in der Ansicht zu belassen; oder durch Plazieren eines SubVIs außerhalb des Hauptdiagrammbereichs. Falls letzteres zutrifft, das Blockdiagrammfenster an die äußersten Grenzen rollen. Außerdem das im VI verwendete Inventar der SubVIs gegen die Menüpunkte **SubVIs des VIs** und **Ungeöffnete SubVIs** im Menü **Projekt** prüfen, um festzustellen, ob irgendwelche externen SubVIs vorhanden sind. Rufen Sie auch das Hierarchiefenster auf, um die SubVIs für ein VI zu betrachten, oder öffnen Sie das Fehlerlistenfenster, um eine Liste mit versteckten Objekten zu betrachten (solange das Kontrollkästchen **Warnungen anzeigen** ausgewählt ist). Damit falsche Ergebnisse aufgrund von versteckten VIs vermieden werden, können Sie angeben, daß bei VIs Eingänge erforderlich sind; weitere Informationen siehe *Erforderliche, empfohlene und optionale Verbindungen für SubVIs* in Kapitel 3, *Verwendung von SubVIs*.

Warnungen

Wenn ein Objekt vollständig durch ein anderes Objekt verdeckt ist, erzeugt G eine Warnung, um anzuzeigen, daß nicht alle Objekte sichtbar sind. Wenn z.B. ein Terminal unter der Kante einer Struktur verborgen ist oder Tunnel übereinander angeordnet sind, erscheint eine Warnmeldung im Fehlerlistenfenster. Warnungen hindern Sie nicht an der Ausführung eines VIs; sie dienen zur Hilfestellung beim Debugging von potentiellen Problemen in Ihren Programmen.

Erkennen undefinierter Daten

Zwei symbolische Werte können in Digitalanzeigen mit Fließkommata erscheinen, um fehlerhafte Berechnungen oder bedeutungslose Ergebnisse anzuzeigen. *NaN* (keine Zahl) ist das Symbol, das einen bestimmten Fließkommawert darstellt, der durch Vorgänge wie das Ziehen der Quadratwurzel aus einer negativen Zahl erzielt werden kann. *Inf* ist ein weiterer besonderer Fließkommawert, der z.B. erzeugt wird, indem Eins durch Null geteilt wird.

Undefinierte Daten können alle folgenden Vorgänge beschädigen. Durch Fließkomma-Vorgänge pflanzen NaN und \pm Inf sich fort, wodurch sie bei der expliziten oder impliziten Umwandlung in Integer- oder Boolesche Werte zu bedeutungslosen Werten werden. Das Dividieren von Eins durch Null erzeugt z.B. den Inf-Wert; die Umwandlung dieses Wertes in einen Word-Integer-Wert jedoch erzeugt 32.767, was ein normaler Wert zu sein scheint. Vor der Umwandlung in Integer-Typen die mittelwerte der Fließkommawerte auf Gültigkeit überprüfen, es sei denn, Sie sind sicher, daß dieser Fehlertyp in Ihrem VI nicht vorkommt.

Die nullmalige Ausführung einer FOR-Schleife kann zu unerwarteten Werten führen. Die Ausgabe eines initialisierten Schieberegisters z.B. ist der anfängliche Wert. Wenn Sie das Schieberegister jedoch nicht initialisieren, ist die Ausgabe entweder der Standardwert für den Datentyp—0, FALSE, oder leerer String—oder die Ausgabe ist der letzte Wert, der bei der letzten Ausführung des Diagramms in das Schieberegister geladen wurde.

Die Indizierung über die Grenzen eines Arrays hinaus erzeugt den Standardwert für den Datentyp des Array-Elements. Dies kann auf mehrere Arten unbeabsichtigt geschehen: z.B. durch Indizierung eines Arrays über das letzte Element hinaus bei Verwendung einer While-Schleife, durch Vorgabe eines zu großen Wertes an die Index-Eingabe einer Index-Array-Funktion oder durch Vorgabe eines leeren Arrays an eine Index-Array-Funktion.

Wenn Sie VIs entwerfen, die undefinierte Ausgabewerte für bestimmte Eingabewerte erzeugen können, verlassen Sie sich nicht auf besondere Werte wie NaN oder leere Arrays zur Identifizierung des Problems. Es muß statt dessen sichergestellt werden, daß Ihr VI entweder eine Fehlermeldung erzeugt, die das Problem bezeichnet, oder daß nur definierte Standarddaten erstellt werden.

Wenn Sie z.B. ein VI mit einem ankommendem Array zur Auto-Indizierung einer For-Schleife erstellen, muß die Funktion des VIs überprüft werden, wenn das Eingabearray leer ist. Dann entweder einen Ausgabefehlercode erstellen oder die durch die Schleife erzeugten Werte durch vordefinierte Werte ersetzen.



Hinweis

Ein Fließkomma-Anzeige- oder Bedienelement mit einem Bereich von – Unendlich bis +Unendlich kann trotzdem einen Bereichsfehler erzeugen, wenn NaN an es gesendet wird.

Korrigieren von VI-Bereichsfehlern

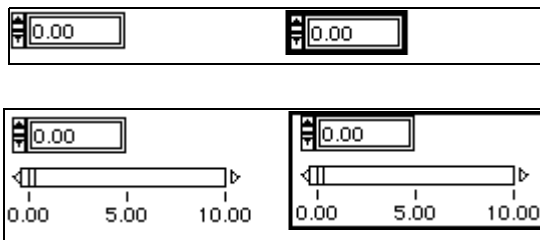


Bereichsfehler-
Anzeigeelement

Ein Bereichsfehler-Anzeigeelement erscheint anstelle der **Ausführen**-Taste unter folgenden Umständen.

- Ein Bedienelement auf einem SubVI wurde so konfiguriert, daß die Ausführung gestoppt wird, wenn es einen Wert außerhalb des Bereichs erhält (über den Menüpunkt **Datenbereich...** des Bedienelements), und dieses Ereignis ist eingetreten.
- Ein Anzeigeelement auf einem SubVI wurde so konfiguriert, daß die Ausführung gestoppt wird, wenn der Versuch unternommen wird, einen Wert außerhalb des Bereichs an ein aufrufendes VI zurückzugeben, und das Anzeigeelement hat versucht, einen solchen Wert zurückzugeben.
- Ein Bediener gibt einen Wert außerhalb des Bereichs in ein Bedienelement ein, das von Ihnen so eingestellt wurde, bei Auftreten eines Fehlers die Ausführung anzuhalten, solange das VI zu diesem Zeitpunkt nicht ausgeführt wird.

Wenn ein Bedien- oder Anzeigeelement außerhalb des Bereichs liegt, ist es von einem dick gezeichneten roten Rechteck umgeben (siehe Abbildung rechts).



Weitere Informationen finden Sie unter [Bereichsoptionen für numerische Bedien- und Anzeigeelemente](#) in Kapitel 9, [Numerische Bedien- und Anzeigeelemente](#).



Hinweis

Ein VI oder SubVI kann Werte außerhalb des Bereichs an eines seiner Anzeigeelemente übergeben, ohne daß die Ausführung angehalten werden muß. Der Fehlerzustand besteht nur dann, wenn ein SubVI versucht, einen solchen Wert zurückzugeben. Außerdem kann ein Bediener einen Wert außerhalb des Bereichs in ein Bedienelement eingeben, während das zugehörige VI ausgeführt wird. Der Fehlerzustand "Außerhalb Bereich" hindert das VI nur am Starten, nicht jedoch an der Fortführung, nachdem es gestartet wurde.

Debugging-Funktionen

Dieser Abschnitt behandelt folgende Debugging-Funktionen.

- Einzelschrittausführung eines VIs zum Beobachten jedes Ausführungsschrittes
- Highlight-Funktion zur Beobachtung des Datenflusses
- Verwendung einer Probe zur Anzeige von Daten
- Einstellung von Breakpoints zum Anhalten der Ausführung, so daß Sie Probe-Verbindungen einzeln durchlaufen können, um die entsprechenden Daten zu betrachten oder die Anzeigeelemente zu bearbeiten
- Unterdrücken der Ausführung, um Anzeigeelemente zu bearbeiten, die Anzahl der Ausführungen zu steuern oder um zum Beginn des VIs zurückzugehen

Einzelschrittausführung der VIs

Für Debugging-Zwecke sollten Sie ein Blockdiagramm Knoten um Knoten ausführen. Diese Form der Ausführung wird als Einzelschrittausführung bezeichnet. Vor der Ausführung eines VIs die Einzelschrittausführung starten, indem Sie die **Anhalten**-Taste drücken. Die Ausführung eines VIs kann außerdem jederzeit angehalten werden, indem Sie auf die **Anhalten**-Taste klicken. Sie können jederzeit zur normalen Ausführung zurückkehren, indem Sie die **Anhalten**-Taste loslassen. Durch das Klicken auf die **Ausführen**-Taste, wenn ein VI angehalten ist, wird die Ausführung wiederaufgenommen, aber VI-Anhalten ist noch eingestellt. Wenn das VI wieder aufgerufen wird, wird die Ausführung angehalten.



Anhalten



Überspringen



Hineinspringen



Herausspringen

Bei der Ausführung des VIs im Einzelschritt-Modus auf eine der drei aktiven Schritttasten drücken, um zum nächsten Schritt vorzurücken. Die Schritttaste, die gedrückt wird, bestimmt den nächsten ausgeführten Schritt.

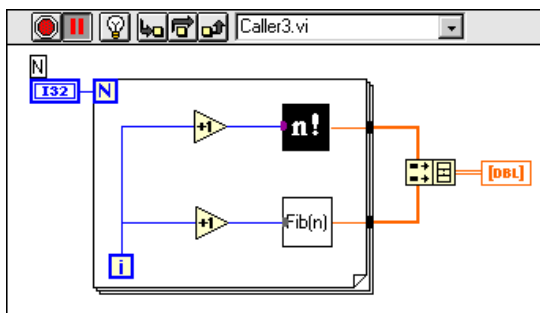
Wenn der Cursor über einer Schritttaste verweilt, erscheint ein Tip Strip mit einer Beschreibung des nächsten Schritts, wenn diese Taste gedrückt wird.

Bei der Einzelschrittausführung eines VIs sollten Sie eventuell die Highlight-Funktion verwenden, um den Datenfluß durch die Knoten zu verfolgen. Siehe den Abschnitt [Highlight-Funktion](#) in diesem Kapitel.

Ein Beispiel der Einzelschrittausführung eines VIs

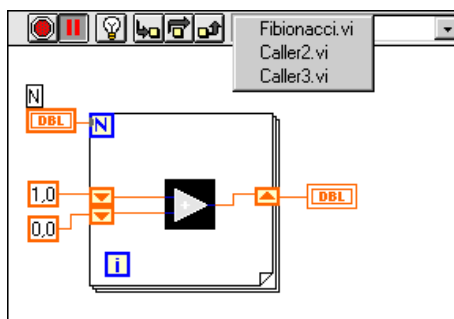
Folgende Abbildung zeigt ein Beispiel eines VIs, das im Einzelschritt-Modus ausgeführt wird (Caller 1.vi). Dieses VI wird momentan für das VI Caller 3.vi ausgeführt. Das SubVI Fibonacci.vi wird momentan

ausgeführt (wird durch das Pfeilsymbol auf der **Ausführen**-Taste angezeigt). Das VI `Factorial.vi` ist der nächste Knoten, der ausgeführt wird. Durch Klicken der Taste **Hineinspringen** wird das dazugehörige Blockdiagramm geöffnet und mit der Einzelschrittausführung begonnen. Durch Klicken der Taste **Überspringen** wird das SubVI ausgeführt und dann angehalten. Durch Klicken der Taste **Herauspringen** wird die Ausführung des aktuellen Rahmens der Schleife beendet und angehalten.



Nach dem Hineinspringen in das SubVI `Fibonacci.vi` und Hineinspringen in die Schleife auf die Taste **Herauspringen** klicken und die Maus eine Sekunde lang niederhalten; es erscheint ein Menü. Wählen Sie, wie weit das VI ausgeführt werden soll, bevor es angehalten wird. Durch die Wahl von **Blockdiagramm** wird das VI ausgeführt, bis alle Knoten auf dem Blockdiagramm ausgeführt werden und die Ausführung angehalten wird.

Durch die Wahl eines VIs im aufrufenden Stapel wird dieses VI ausgeführt, bis es abgeschlossen ist. An diesem Punkt hält der Aufrufer des ausgewählten VIs an.



Verwendung der Schritttasten

Die Schritttasten werden wie folgt verwendet:

Die Taste **Überspringen** drücken, um eine Struktur (Sequenz, Schleife etc.) oder ein SubVI auszuführen und anschließend am nächsten Knoten anzuhalten. Die Tastenkombination dafür ist die <Strg>- (**Windows**); <Befehl>- (**Macintosh**); <Meta>- (**Sun**) oder <Alt>- (**HP-UX**) Taste, gefolgt von der Nach-Rechts-Taste.



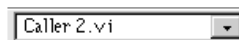
Hineinspringen

Die Taste **Hineinspringen** drücken, um den ersten Schritt eines SubVIs bzw. einer Struktur (Sequenz, Schleife etc.) auszuführen und dann am nächsten Schritt des SubVIs bzw. der Struktur anzuhalten. Die Tastenkombination dafür ist die <Strg>- (**Windows**); <Befehl>- (**Macintosh**); <Meta>- (**Sun**) oder <Alt>- (**HP-UX**) Taste, gefolgt von der Nach-Unten-Taste. Die Schritttasten wirken sich bei einem VI oder SubVI nur im Einzelschritt-Modus aus. Wenn ein VI im Einzelschritt-Modus über ein SubVI verfügt, das sich ebenfalls im Einzelschritt-Modus befindet und ein SubVI, das sich im normalen Ausführungsmodus befindet, wird das erste SubVI bei Aufruf im Einzelschritt-Modus ausgeführt, das zweite wird jedoch bei Aufruf normal ausgeführt. Die Taste **Herauspringen** drücken, um die Ausführung des aktuellen Blockdiagramms, der Struktur bzw. des VIs zu beenden und anzuhalten. Die Tastenkombination ist die <Strg>- (**Windows**); <Befehl>- (**Macintosh**); <Meta>- (**Sun**) oder <Alt>- (**HP-UX**) Taste, gefolgt von der Nach-Oben-Taste.

Wenn das VI die Ausführung beendet hat, sind die Schritttasten als nicht bedienbar markiert.

Lesen von Aufrufketten

Wenn ein SubVI angehalten wird, erscheint ein **Aufrufkettenring**. In diesem Menü wird die Kette der Aufrufer vom VI der höchsten Ebene bis hinunter zu diesem SubVI aufgeführt. Denken Sie daran, daß es sich hierbei nicht um denselben Menüpunkt wie **Projekt>Aufrufer des VIs** handelt, in dem alle aufrufenden VIs aufgeführt sind, unabhängig davon, ob sie momentan ausgeführt werden. Wenn Sie ein VI vom Aufrufkettenring auswählen, wird das dazugehörige Blockdiagramm geöffnet, und das VI, welches das aktuelle SubVI aufruft, wird ausgewählt. Dadurch können Sie die aktuelle Instanz des SubVIs bestimmen, wenn das Blockdiagramm mehr als eine Instanz enthält.



Aufrufkettenring

Zusätzlich zu den Tastenkombinationen, die im Abschnitt *Verwendung der Schritttasten* beschrieben sind, stehen folgende Befehle zur Verfügung:

- Bei der Einzelschrittausführung eines SubVIs bzw. einer Struktur kann durch Klicken auf die Taste **Herausspringen** bei gleichzeitigem Niederhalten der Maus ein Menü aufgerufen werden. Wählen Sie aus, wie weit das VI vor dem Anhalten ausgeführt werden soll.
- Durch Doppelklicken auf Bedien- und Anzeigeelemente, lokale und globale Variablen des Frontpanels erscheint das entsprechende Blockdiagramm-Objekt und wird hervorgehoben.
- Durch das Drücken der <Strg>- (**Windows**); <Option>- (**Macintosh**); <Meta>- (**Sun**) oder <Alt>- (**HP-UX**) Taste bei gleichzeitigem Doppelklicken auf ein SubVI wird das Blockdiagramm in den Vordergrund geholt.

Highlight-Funktion

Für Debugging-Zwecke ist es eventuell hilfreich, eine Animation der Ausführung des VI-Blockdiagramms zu betrachten.



aus

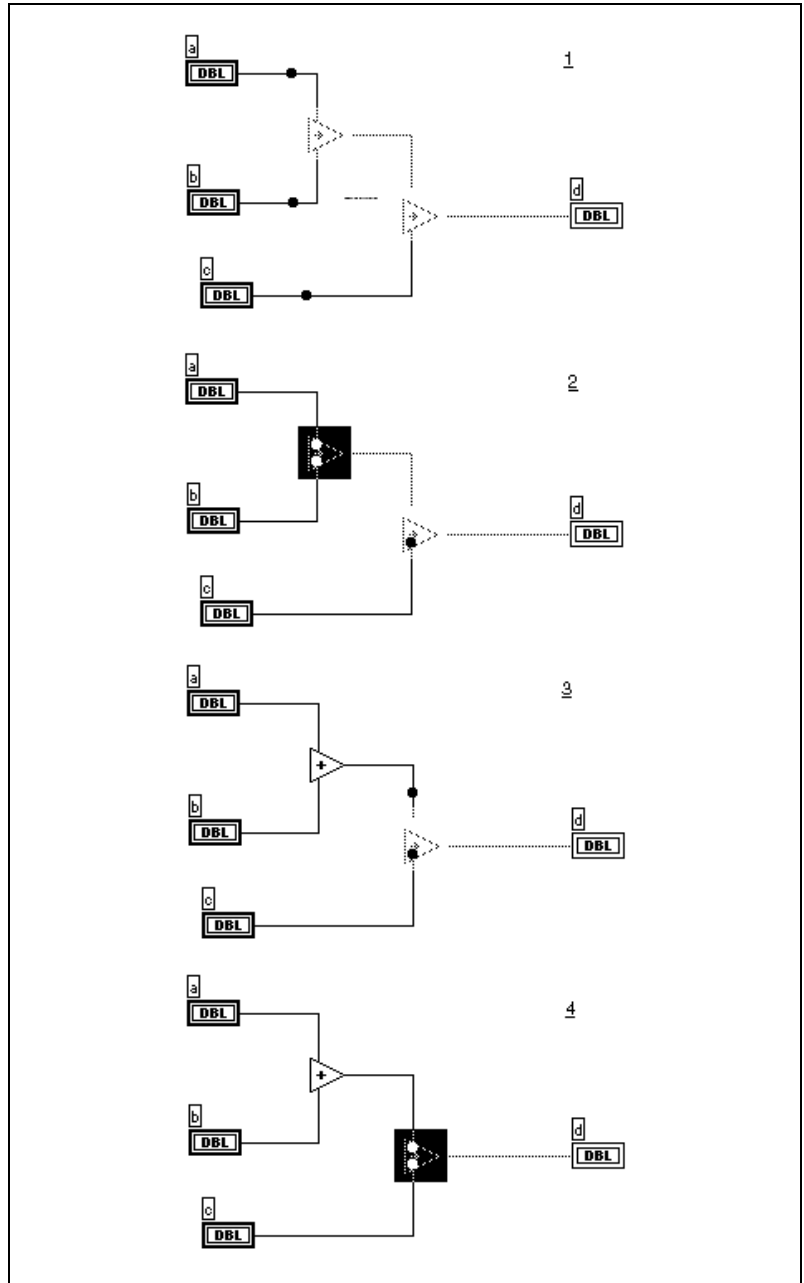


ein

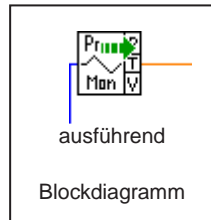
Highlight-Funktion

Zur Verwendung dieser Funktion auf die Taste **Highlight-Funktion** klicken. Die Taste ändert ihr Aussehen. Zur Rückkehr zum normalen Ansichtsmodus können Sie jederzeit diese Taste klicken. Die Highlight-Funktion wird im allgemeinen in Verbindung mit dem Einzelschritt-Modus verwendet, um einen Einblick in den Datenfluß durch die Knoten zu gewinnen. Mit der Highlight-Funktion wird die Leistung eines VIs ganz erheblich gemindert.

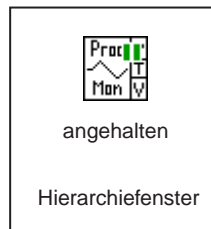
Durch die Highlight-Funktion wird die Bewegung von Daten von einem Knoten zu einem anderen Knoten durch Bubbles markiert, die sich entlang der Verbindung bewegen. Außerdem blinkt im Einzelschritt-Modus der nächste Knoten in rascher Folge (siehe nachstehende Abbildungssequenz).



Wenn Sie ein SubVI bei eingeschalteter **Highlight-Funktion**-Taste im Einzelschritt-Modus ausführen, zeigt ein Ausführungssymbol auf dem SubVI-Icon des Blockdiagramms an, welche VIs ausgeführt werden und welche auf ihre Ausführung warten. Der Pfeil in folgender Abbildung zeigt an, daß ein SubVI auf seine Ausführung wartet.



Im Hierarchiefenster wird angezeigt, ob ein VI angehalten ist und/oder unterdrückt wird. (Weitere Informationen über das Hierarchiefenster finden Sie unter im Abschnitt *Verwendung des Hierarchiefensters* von Kapitel 3, *Verwendung von SubVIs*) Das Anhalten-Symbol zeigt an, daß ein SubVI angehalten wurde und/oder unterdrückt wird. Ein grünes Anhalten-Symbol (oder ein hohles Symbol in Schwarz-Weiß angezeigt) gibt an, daß dieses SubVI beim Aufrufen angehalten wird. Ein rotes Anhalten-Symbol (oder ein durchgezogenes Symbol in Schwarz-Weiß) gibt an, daß dieses SubVI momentan angehalten ist.



Ausgewähltes Probe-Werkzeug

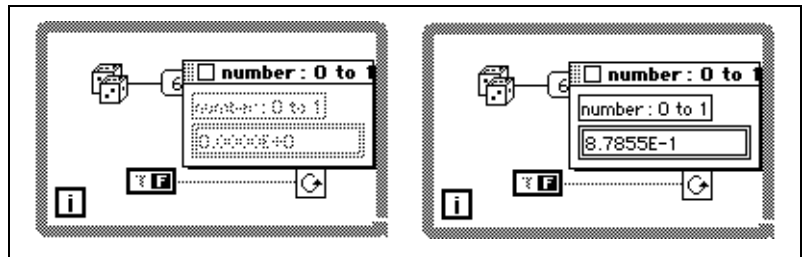
Verwendung des Probe-Werkzeugs

Sie können das Probe-Werkzeug zum Prüfen von Zwischenwerten in einem VI verwenden, das ausgeführt wird, jedoch fragwürdige oder unerwartete Ergebnisse erzeugt. Sie haben z.B. ein kompliziertes Blockdiagramm mit einer Reihe von Operationen, von denen eine eventuell falsche Daten zurückgibt.

Eine Methode zum Auffinden der Quelle der fragwürdigen Ergebnisse ist das Verbinden eines Anzeigeelements mit der Ausgangsverbindung einer der Operationen zur Anzeige der Zwischenergebnisse. Das Plazieren eines Anzeigeelements auf das Frontpanel und die Verbindung des

dazugehörigen Terminals mit dem Blockdiagramm ist jedoch keine praktische Debugging-Methode. Es ist zeitraubend und führt zu unerwünschten Elementen auf Ihrem Frontpanel und Blockdiagramm, die Sie später wieder löschen müssen.

Das Probe-Werkzeug auswählen und den Cursor auf eine Verbindung plazieren oder das Popup-Menü der die Verbindung aufrufen und **Probe** auswählen. Wie in folgender Abbildung dargestellt, erscheint das floatende Probe-Fenster ohne Anzeige von Daten. Sobald das VI ausgeführt wird, zeigt das Probe-Fenster Daten, die über die Verbindung weitergeleitet werden.



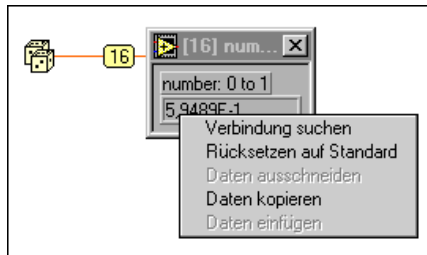
Jede Probe wird automatisch und eindeutig numeriert. Die Verbindung, an die sie angeschlossen ist, ist mit derselben Nummer markiert. Dadurch kann darüber Buch geführt werden, welche Probe mit welcher Verbindung assoziiert ist. Falls jedoch der Name des Frontpanel-Bedienelementes so lang oder länger als das Probe-Fenster ist, ist die Nummer nicht sichtbar.

Die Probe kann im Zusammenhang mit Highlight-Funktionen, Einzelschrittausführung oder Breakpoints verwendet werden, damit Ansichtswerte leichter berechnet werden können. Während einer Einzelschrittausführung oder dem Anhalten eines Breakpoints werden verfügbare Daten sofort aktualisiert. Prüfen Sie die Eingänge, wenn die Ausführung an einem Knoten angehalten wird.

Die Probe kann vor der Ausführung des VIs zur Ansicht der Daten eingefügt werden. Für die Einzelschrittausführung eine Probe für eine ausgeführte Verbindung erstellen. Die Probe wird aktualisiert und zeigt den Inhalt der Verbindung an. Diese Funktion ist nützlich, wenn Sie einen Breakpoint auf einem Knoten eingestellt haben und Sie dessen Inhalt prüfen möchten.

Daten können mit der Probe nicht geändert werden, und die Probe hat keine Auswirkung auf die Ausführung des VIs.

Wenn Sie das Popup-Menü des Anzeigeelements des Probe-Fensters aufrufen, mit der Funktion **Verbindung suchen** die dazugehörige Verbindung suchen (siehe nachstehende Abbildung).



Wenn Sie den Menüpunkt **Verbindung suchen** auswählen, wird das Blockdiagramm mit dieser Verbindung in den Vordergrund geholt, und die Verbindung wird hervorgehoben.

Mit der Option **Rücksetzen auf Standard** kann der im Probe-Fenster angezeigte Wert wieder auf den Standardwert zurückgesetzt werden.

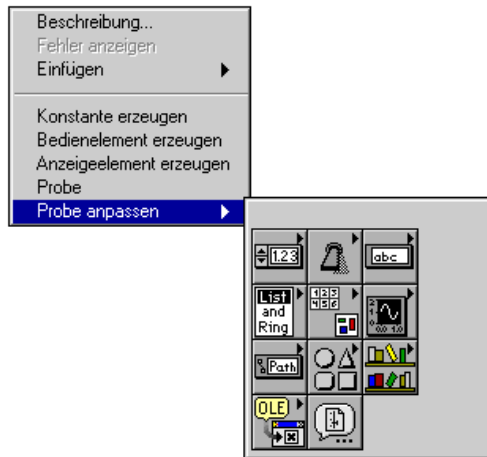
Mit dem Menüpunkt **Daten kopieren** können die Daten in andere numerische Bedienelemente im selben VI oder in anderen VIs kopiert werden.

Erstellen von Proben

Wenn Sie eine Probe erstellen, dann erstellen Sie eine Probe im Standardstil, die an den Datentyp der Verbindung angepaßt ist. Bei numerischen Datentypen zum Beispiel werden die Daten einer Verbindung mit Hilfe eines Digital-Anzeigeelements geprüft.

Wenn gewünscht, können Sie ein Bedienelement aus den eingebauten Bedienelementen für die Probe aussuchen; oder Sie können Bedienelemente verwenden, die bereits als benutzerspezifische Bedienelemente oder Typendefinitionen gesichert wurden. Dazu das Popup-Menü der Verbindung aufrufen, **Probe anpassen** auswählen und ein Bedienelement von der Palette **Probe anpassen** aussuchen, die rechts erscheint. Wenn Sie **Element auswählen...** wählen, mit dem Dateialogfeld beliebige benutzerspezifische Bedienelemente oder Typendefinitionen auswählen, die im Dateisystem gespeichert wurden. Typendefinitionen werden wie standardmäßige benutzerspezifische

Bedienelemente behandelt, wenn sie zur Prüfung von Daten eingesetzt werden.



Nur die Teile auf der **Probe anpassen**-Palette, die eventuell zum Datentyp der Verbindung passen, sind aktiviert. Wenn Sie ein Bedienelement auswählen, das nicht über denselben Datentyp wie die Verbindung verfügen kann, wird dieser Konflikt durch einen Piepton signalisiert, und die Probe wird nicht erstellt. Bei Arrays und Clustern z.B. können die Array- und Cluster-Shells von der **Array & Cluster**-Palette nicht verwendet werden, da es sich dabei nicht um komplementäre Datentypen handelt.

Sie können am Ende der **Probe anpassen**-Palette Ihre eigenen Paletten anfügen, und zwar genauso, wie sie bei der **Elemente**-Palette angefügt werden. Weitere Informationen darüber finden Sie unter [Anpassen Ihrer Umgebung](#), Kapitel 7.

Plazieren von Breakpoint-Werkzeugen

Sie können einen Breakpoint auf ein VI, einen Knoten oder eine Verbindung setzen, um ein Anhalten während der Ausführung einzustellen. Knoten enthalten SubVIs, Funktionen, Strukturen, Code Interface Nodes (CINs), Formelknoten und Attributknoten. Wenn ein Blockdiagramm über einen Breakpoint verfügt, wird die Ausführung angehalten, nachdem alle Knoten auf dem Diagramm ausgeführt wurden. Wenn eine Verbindung einen Breakpoint aufweist, wird die Ausführung angehalten, nachdem Daten durch die Verbindung gelaufen sind.

Wenn Sie während der Ausführung einen Breakpoint erreichen, können Sie mit Hilfe der Schritttasten zur Einzelschrittausführung wechseln, Verbindungen zur Ansicht der dazugehörigen Daten prüfen, Werte der Frontpanel-Bedienelemente ändern oder die Ausführung bis zum nächsten Breakpoint oder bis Abschluß der Ausführung weiterlaufen lassen. Durch Klicken auf die **Ausführen**-Taste kann die Ausführung leichter bis zum nächsten Breakpoint oder bis das SubVI wieder aufgerufen wird, fortgesetzt werden.



Breakpoint-Werkzeug



Breakpoint löschen

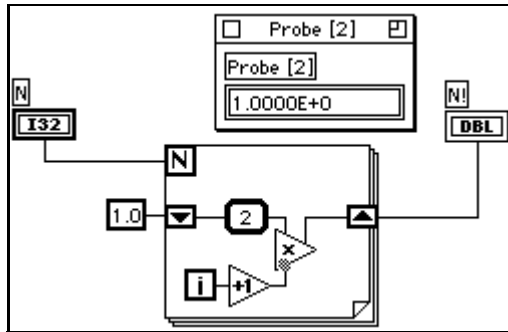
Zum Einstellen eines Breakpoints das Breakpoint-Werkzeug in der **Werkzeuge**-Palette auswählen und anschließend auf das Blockdiagramm, den Knoten oder die Verbindung klicken. Zum Entfernen des Breakpoints können Sie mit dem Werkzeug jederzeit auf dasselbe Objekt klicken. Die Erscheinung des Werkzeugs zeigt an, ob ein Breakpoint eingestellt oder gelöscht ist (siehe Darstellung links).

Folgende Tabelle zeigt, wie Breakpoints angezeigt werden und wann die Ausführung angehalten wird, und zwar im Verhältnis zur Platzierung der Breakpoints.

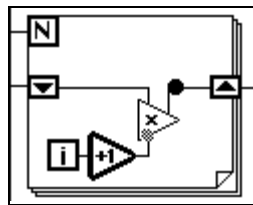
Tabelle 4-2. Breakpoint-Plazierung

Position des Breakpoints	Wie ein Breakpoint hervorgehoben wird	Wenn Anhalten auftritt
Blockdiagramm	Rote Umrandung um Blockdiagramm. Wenn das Diagramm sich innerhalb einer Struktur befindet, liegt die rote Umrandung ebenfalls innerhalb der Struktur.	Wenn alle Knoten innerhalb Blockdiagramms die Ausführung beendet haben. Im Fall von Schleifen tritt das Anhalten nach jeder Schleifenwiederholung auf.
Knoten	Rote Umrandung um den Knoten.	Unmittelbar vor der Knotenausführung. Zu diesem Zeitpunkt können alle Eingangssignale in den Knoten mit dem Probe-Werkzeug geprüft werden.
Verbindung	Rotes Aufzählungszeichen in der Mitte der Verbindung. Wenn eine Probe an der Verbindung angeschlossen ist, hat die Probe eine rote Umrandung.	Nach Durchlauf der Daten durch die Verbindung.

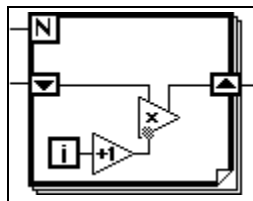
Im folgenden Beispiel hat die Verbindung mit der angeschlossenen Probe einen Breakpoint. Das VI hält an, nachdem die Probe Daten anzeigt.



Im nächsten Beispiel hat der Inkrement-Knoten einen Breakpoint. Das VI hält an, bevor das Inkrement ausgeführt wird. Eine Verbindung hat ebenfalls einen Breakpoint. Das VI hält erneut an, nachdem Multiplizieren ausgeführt wird.



Im folgenden Beispiel hat die FOR-Schleife einen Breakpoint. Das VI hält an, nachdem Multiplizieren ausgeführt wird.



Wenn ein VI aufgrund eines Breakpoints anhält, wird das dazugehörige Blockdiagramm in den Vordergrund geholt (wenn das Frontpanel geschlossen war, wird es außerdem geöffnet), und der Knoten bzw. die Verbindung, die das Anhalten auslöst, wird mit einem Auswahlrechteck hervorgehoben.

Breakpoints werden zusammen mit dem jeweiligen VI gespeichert; sie werden jedoch nur während der Ausführung aktiviert.

Unterdrücken der Ausführung

Bedien- und Anzeigeelemente können vor der Rückkehr zum Aufrufer so oft wie Sie wollen bearbeitet, und das SubVI kann so oft wie gewünscht ausgeführt werden, oder Sie können zum Beginn des SubVIs zurückkehren, indem die Ausführung eines SubVIs unterdrückt wird.

Zum Einstellen eines SubVIs im Unterdrückungsmodus das SubVI öffnen und den Menüpunkt **Ausführen»Bei Aufruf anhalten** prüfen. Diese Auswahl kann außerdem aufgerufen werden, indem Sie das Popup-Menü des Anschlußfeldes auf dem Frontpanel aufrufen, während Sie sich im Ausführungsmodus befinden, und den Menüpunkt **Einstellung»Ausführungsoptionen»Bei Aufruf anhalten...** auswählen. Das SubVI wird bei Aufruf automatisch unterdrückt. Wenn Sie bei der Einzelschrittausführung diesen Menüpunkt markieren, schaltet das VI nicht sofort in den Unterdrückungsmodus.

Wenn Sie die Ausführung bei einem bestimmten Aufruf an ein SubVI unterdrücken möchten, verwenden Sie den Menüpunkt **Einstellungen SubVI...** vom Popup-Menü des SubVI-Knotens anstelle von **Bei Aufruf anhalten**. Mit **Einstellungen SubVI...** wird nur die Ausführung dieser besonderen Instanz des SubVIs unterdrückt.

Erkennen automatischer Unterdrückung

Ein VI wird automatisch unterdrückt, wenn während der Ausführung ein Bedien- oder Anzeigeelement außerhalb des Bereichs liegt. Es wird das Terminal bzw. die lokale Variable ausgewählt, der bzw. die den Fehler "Außerhalb Bereich" verursacht.

Ein SubVI, dessen Bedien-oder Anzeigeelemente so eingestellt sind, daß bei einem Bereichsfehler angehalten wird, verfügt über einen bedingten Breakpoint. Wenn ein Bereichsfehler auftritt, hält das SubVI so an, als ob ein Breakpoint vorliegt. Wenn kein Bereichsfehler auftritt, wird das SubVI normal ausgeführt.

Wenn Daten, die an ein SubVI übergeben werden, zu einem Bereichsfehler führen, wird das Frontpanel des SubVIs geöffnet bzw. es wird in den Vordergrund geholt, und das SubVI verbleibt im Unterdrückungsmodus. Zu diesem Zeitpunkt sind die Werte auf den SubVI-Bedienelementen die Eingaben, die durch das aufrufende VI übergeben wurden, und Sie können die Werte gegebenenfalls ändern. Zur Ausführung des SubVIs müssen sie sogar geändert werden, wenn das Bereichsfehler-Anzeigeelement eingeschaltet ist. Die Anzeigeelemente des SubVIs zeigen entweder

Standardwerte oder Werte von der letzten Ausführung des SubVIs an, während derer das Frontpanel geöffnet war.

Verwendung von Werkzeugpaletten-Tasten bei Unterdrückung von SubVIs



Ausführen

Wenn das aktuelle SubVI vor Rückkehr zum Aufrufer ausgeführt werden soll, die **Ausführen**-Taste drücken (oder den Befehl **Ausführen» Ausführen** wählen), während Sie sich im Unterdrückungsmodus befinden. Die Ausführung beliebig oft wiederholen.

Wenn das SubVI abgeschlossen ist, zeigen die Anzeigeelemente die Ergebnisse der Ausführung des SubVIs an. Sie können die Werte des Anzeigeelements jedoch ändern, wenn Sie andere Werte an das aufrufende VI zurückgeben möchten. Der Unterdrückungsmodus ist sogar die einzige Möglichkeit, Werte an Anzeigeelementen einzustellen. Erneut die Taste **Zurück zu Aufrufendem** klicken, wenn Sie die Werte des Anzeigeelements des SubVIs an das aufrufende VI zurückgeben möchten.

Springe zum
Anfang

Wenn Sie zum Beginn des VIs zurückkehren möchten, auf die Taste **Springe zum Anfang** klicken.

Zurück zu
Aufrufendem

Die Taste **Zurück zu Aufrufendem** erscheint, wenn ein unterdrücktes SubVI nicht ausgeführt wird. Zur Rückkehr zum aufrufenden VI diese Taste klicken. Beachten Sie, daß Sie zum Aufrufer zurückkehren können, ohne das aktuelle VI auszuführen. Wenn Sie es ausführen möchten, müssen Sie zuerst die **Ausführen**-Taste drücken, bevor Sie zum Aufrufer zurückkehren.

Ansicht der Hierarchiefenster während der Unterdrückung



Im Hierarchiefenster erscheint ein Ausrufungszeichen-Symbol, das auf ein unterdrücktes SubVI hinweist. Folgende Abbildung zeigt ein SubVI im Hierarchiefenster, während die Option **Bei Aufruf anhalten** eingeschaltet ist.

Deaktivieren von Debugging-Funktionen

Zur Verringerung der Speichererfordernisse und zum Erzielen einer geringfügigen Leistungssteigerung können Sie ein VI ohne die verschiedenen Debugging-Funktionen kompilieren. Dazu vom Anschlußfeld-Popup-Menü auf dem Frontpanel aus den Menüpunkt **VI-Einstellungen» Fensteroptionen** aufrufen und das Kontrollkästchen **Debugging zulassen** abwählen.

Ausschließen von Diagrammabschnitten

In einigen Fällen möchten Sie eventuell ein VI ausführen, während ein Abschnitt des Blockdiagramms deaktiviert ist. Am leichtesten können Sie dies tun, indem der zu deaktivierende Diagrammabschnitt in einer Case-Struktur eingeschlossen und gleichzeitig ein konstanter Boolescher Wert mit dem Wähler verbunden wird. Das zu deaktivierende Diagramm in den Rahmen der Case-Struktur plazieren, die nicht ausgeführt wird.

Wenn die Case-Struktur irgendwelche Datenwerte erzeugt, müssen Konstanten für die Ausgabetunnel erstellt werden.

Drucken und Dokumentieren von VIs

Dieses Kapitel beschreibt eine Vielzahl von Themen im Zusammenhang mit dem Drucken und Dokumentieren von VIs.

Drucken

In G gibt es vier Hauptmethoden zum Drucken von VIs.

- Mit dem Menüpunkt **Fenster drucken...** können Sie einen schnellen Ausdruck des Inhalts des aktuellen Fensters anfertigen.
- Einen umfassenderen Ausdruck eines VIs einschließlich Informationen über das Frontpanel, Blockdiagramm, SubVIs, Bedienelemente, VI-Historie usw. können Sie durch Auswahl von **Dokumentation drucken...** erstellen.
- Mit der programmatischen Druckfunktion können VIs so eingestellt werden, daß die dazugehörigen Frontpanels unter dem Bedienelement Ihrer Anwendung automatisch gedruckt werden.
- Mit dem VI-Server kann ein VI-Fenster bzw. eine VI-Dokumentation programmatisch gedruckt werden.

Außerdem kann mit seriellen Schnittstellen-VIs Text an den Drucker geschickt werden, wenn Ihr Drucker an eine serielle oder parallele Schnittstelle angeschlossen ist; dabei werden im allgemeinen einige Kenntnisse der Befehlssprache des Druckers vorausgesetzt.

In Windows und UNIX ist eine weitere Option für das Drucken von Text die Verwendung vom "System Exec-VI", das sich unter **Funktionen»Kommunikation** befindet. Auf dem Macintosh das "AESend Print Document-VI" verwenden, das sich unter **Funktionen»Kommunikation»AppleEvent** befindet. In Windows können Sie ActiveX-Automatisierung verwenden, um eine andere Anwendung zum Drucken von Daten aufzufordern.

Druckkonfiguration

Es gibt zwei Dialogfelder, die sich auf die Erscheinung aller Ausdrucke auswirkt, und zwar unabhängig von der zum Drucken verwendeten Methode: das Dialogfeld Voreinstellungen und das Dialogfeld Einstellungen.

Die Dialogseite Drucken des Dialogfelds “Voreinstellungen” wird zur Konfiguration von Druckoptionen verwendet.

Zur Konfiguration von druckerspezifischen Informationen und Formatierungen den Menüpunkt **Druckereinstellungen...** (**Seite einrichten** auf dem Macintosh) im **Datei**-Menü verwenden. Bei den meisten Druckern wird z.B. die Ausrichtung der Ausdrucke (Querformat im Vergleich zu Hochformat) mit diesem Dialogfeld geändert. Zahlreiche Drucker verfügen außerdem über Optionen für Schriftarten-Ersetzung, Papiergröße und andere druckerspezifische Einstellungen. **Druckereinstellungen...** (**Seite einrichten** für den Macintosh)-Einstellungen werden mit Ihrem VI gespeichert.

PostScript-Drucken

Wenn Sie Zugriff auf einen PostScript-Drucker haben, könnten Sie auch den Menüpunkt **Bearbeiten»Voreinstellungen»Drucken»PostScript drucken** auswählen, anstatt andere Ausgabeoptionen zu verwenden.

Ein PostScript-Drucker bietet folgende Vorteile.

- PostScript-Druck gibt das auf dem Bildschirm gezeigte Bild genauer wieder.
- PostScript führt zu besseren Ergebnissen bei graphischen Darstellungen mit hoher Auflösung.
- PostScript gibt Muster, Linienarten und Schriftarten genauer wieder.

Das Dialogfeld Voreinstellungen wird dazu verwendet, anzuzeigen, ob Sie PostScript-Drucken verwenden möchten. Weitere Informationen zum Dialogfeld “Voreinstellungen” befinden sich in Kapitel 7, [Anpassen Ihrer Umgebung](#).

Drucken des aktiven Fensters

Mit dem Menüpunkt **Fenster drucken...** kann der Inhalt des momentan aktiven Fensters (Frontpanel oder Blockdiagramm) gedruckt werden. Mit diesem Menüpunkt können Sie einen schnellen Ausdruck mit der

geringsten Zahl von Aufforderungszeichen erstellen. Für umfassendere Ausdrücke von VIs verwenden Sie **Dokumentation drucken....**

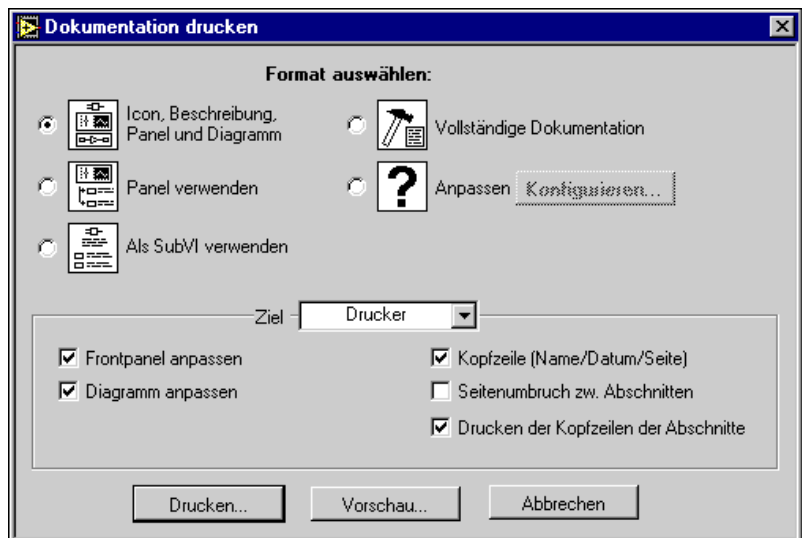
Fenster drucken... formatiert den Ausdruck so, daß er genauso aussieht, als wenn das VI für den programmatischen Druck eingestellt wird. Der Abschnitt **Ausführungsoptionen** des Dialogfeldes VI-Einstellung enthält einige Optionen, die eine bessere Kontrolle des Druckbildes beim programmatischen Drucken oder unter Verwendung des Menüpunkts **Fenster drucken....** ermöglichen. Informationen zu diesen Optionen befinden sich im Abschnitt *Einstellen des Seitenlayouts* weiter hinten in diesem Kapitel.

Dokumentieren von VIs

Dieser Abschnitt beschreibt das Drucken Ihrer VI-Dokumentation auf Papier oder als RTF- und HTML-Dateien; außerdem enthält es entsprechende Informationen über die Formatierung von Grafik-Dateien.

Drucken der Dokumentation

Wenn ein ausführlicher Ausdruck des Inhalts eines VIs gewünscht ist, den Menüpunkt **Dokumentation drucken...** verwenden (siehe nachstehende Abbildung). Mit diesem Dialogfeld kann aus mehreren VI-Formaten gewählt werden. Außerdem können Sie ein benutzerspezifisches Format erstellen.



Einstellen von Ausdruckformaten

Das Dialogfeld “Dokumentation drucken” verfügt über fünf unterschiedliche Druckformate sowie eine benutzerspezifische Option zur Erstellung Ihres eigenen Formates. Das Icon für jedes dieser Formate ist links von der Beschreibung abgebildet.



Das Format **Icon, Beschreibung, Panel und Diagramm** (der Standard) druckt das Icon, die VI-Beschreibung, das Frontpanel und das Blockdiagramm.



Das Format **Panel verwenden** druckt das Frontpanel, die VI-Beschreibung und die Namen der Bedienelemente und ihre Beschreibungen.



Das Format **Als SubVI verwenden** druckt das Icon, den Anschluß, die VI-Beschreibung und die Terminals (Datentypen), Namen und Beschreibungen der angeschlossenen Bedienelemente. Dieses Format gleicht dem G-Funktionsreferenz-Format.



Das Format **Vollständige Dokumentation** druckt alles einschließlich dem Icon, dem Anschluß, der Beschreibung, dem Frontpanel, der Informationen über alle Bedienelemente des Frontpanels, dem Blockdiagramm und einer Liste der Namen des SubVIs.



Das Format **Anpassen** wird unter Verwendung der aktuellen benutzerspezifischen Einstellungen gedruckt. Die benutzerspezifischen Einstellungen durch die Wahl von **Konfigurieren...** ändern, nachdem Sie den Menüpunkt benutzerspezifisch ausgewählt haben. Weitere Informationen über dieses Dialogfeld finden Sie im Abschnitt [Erstellen von benutzerspezifischen Druckeinstellungen](#) in diesem Kapitel.

Das Auswahlfeld **Ziel** bietet eine Auswahl zum Senden der Daten an den Drucker, die HTML-Datei, die RTF-Datei oder eine Standardtext-Datei. Weitere Informationen über die Funktionen von **Ziel** befinden sich im Abschnitt [Drucken/Exportieren von Bedienelement- und VI-Beschreibungen in eine RTF- oder HTML-Datei](#) in diesem Kapitel.

Einstellen anderer Druckoptionen

Das Dialogfeld **Dokumentation drucken** enthält mehrere Seitenlayoutoptionen zur Steuerung von Skalierung, Seitenumbrüchen und Kopfzeilen für den Ausdruck. Diese werden wie folgt beschrieben.

Frontpanel anpassen und **Blockdiagramm anpassen**— Wenn diese Option über diesen Menüpunkt eingestellt ist, kann das Frontpanel und/oder das Blockdiagramm auf ein Viertel der Originalgröße herunterskaliert werden, damit es besser in die Seiten paßt.

Kopfzeile drucken— Druckt auf jeder Seite oben eine Kopfzeile. Diese Kopfzeile umfaßt eine Seitenzahl, den Namen des VIs und das letzte Änderungsdatum des VIs.

Seitenumbruch zwischen Abschnitten— Fügt einen Seitenumbruch zwischen folgenden Abschnitten ein:

- Anschluß-Icon und Beschreibung
- Frontpanel
- Liste der Bedienelement-Details des Frontpanels
- Blockdiagramm
- Blockdiagramm-Details
- VI-Hierarchie
- Liste der SubVIs

Drucken der Kopfzeilen der Abschnitte— Druckt eine Kopfzeile für jeden Abschnitt (z.B. eine Kopfzeile wie eine Liste der SubVIs vor den SubVI-Informationen). Beim Drucken unter Verwendung von **Dokumentation drucken** wird diese Option automatisch für jedes Format eingestellt.

Erstellen von benutzerspezifischen Druckeinstellungen

Zum Aufrufen des Dialogfeldes Benutzerspezifische Druckeinstellungen die Taste **Anpassen** im Dialogfeld Dokumentation drucken auswählen. Nach der Wahl von **Anpassen** wird die Taste **Konfigurieren** aktiviert.

Diese Taste klicken, um das in folgender Abbildung dargestellte Dialogfeld aufzurufen.



In diesem Dialogfeld werden die Kategorien der Elemente angezeigt, die gedruckt werden können. Diese Kategorien sind in der Reihenfolge aufgeführt, in der sie im Ausdruck erscheinen. Außerdem werden mit den Seitenlayoutoptionen in diesem Dialogfeld Skalierung, Seitenumbrüche und Kopfzeilen für den Ausdruck gesteuert.

Das Icon für jeden Menüpunkt der benutzerspezifischen Druckeinstellungen ist links von der Beschreibung abgebildet.



Icon und Beschreibung

VI-Anschluß und Icon—Druckt ein Bild des VI-Icons sowie die dazugehörigen Ein- und Ausgänge.

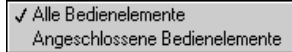
VI-Beschreibung—Druckt die VI-Beschreibung.



Frontpanel—Druckt das Frontpanel.



Elemente—Druckt eine Liste der Namen der Bedien- und Anzeigeelemente. Wenn ein Array, Cluster oder eine Refnum aufgefunden wird, werden die untergeordneten Bedienelemente gedruckt. Folgende Menüpunkte vom Popup-Menü auswählen.



Beschreibungen—Druckt die Beschreibungen neben den Namen der Bedienelemente.

Datentypinformationen aufnehmen—Wenn über einen Drucker ausgegeben wird, wird das Terminal für jedes Bedienelement links vom Namen des Bedienelements gedruckt. Wenn die Daten in einer Textdatei gespeichert werden, befindet sich der Datentyp nach der Beschreibung.



Blockdiagramm—Druckt das Blockdiagramm.

Verborgene Rahmen—Druckt die nicht sichtbaren Rahmen der Case-Strukturen und Sequenzstrukturen.

Rahmen höherer Ebene wiederholen—Beim Drucken von nicht sichtbaren Rahmen werden die sichtbaren Rahmen erneut in der entsprechenden Reihenfolge gedruckt.



VI-Hierarchie—Druckt eine Beschreibung der aktuellen VI-Hierarchie im Speicher, wobei die Anschlüsse zwischen VIs und ihren SubVIs durch Linien dargestellt sind. Das aktuelle VI ist mit einem Kästchen hervorgehoben.



Liste der SubVIs—Druckt das Icon, den Namen und Pfad aller verwendeten SubVIs.



VI-Historie—Druckt die Historieninformationen des aktuellen VIs, falls vorhanden.

Programmatisches Drucken

Wenn Sie anstatt mit der interaktiven Methode mit dem Druckfenster und dem Dialogfeld "Dokumentation drucken" unter dem Bedienelement Ihres VI etwas drucken möchten, können Sie dies auf zweierlei Arten tun.

- Unter Verwendung des programmatischen Druckens können Sie ein VI so einstellen, daß das Panel automatisch nach Abschluß jeder Ausführung gedruckt wird.
- Mit dem VI-Server können Sie das Panel oder die Dokumentation eines beliebigen VIs drucken.

Zur Anwendung des programmatischen Druckens den Menüpunkt **Ausführen»Drucken nach Abschluß** auswählen.

Wenn diese Option eingeschaltet ist, druckt das VI den Inhalt des Frontpanels jedesmal dann, wenn die Ausführung eines VIs abgeschlossen ist. Wenn es sich bei dem VI um ein SubVI handelt, wird das Panel gedruckt, wenn die Ausführung des SubVIs abgeschlossen ist und bevor es zum Aufrufer zurückkehrt.

Steuern des Ausdruckzeitpunkts

In einigen Fällen soll ein VI eventuell nicht jedesmal ausgedruckt werden. Sie möchten es vielleicht so einstellen, daß es nur dann ausgedruckt wird, wenn der Benutzer eine Taste drückt oder wenn irgendein Zustand auftritt, wie z.B. ein Testfehler. Vielleicht möchten Sie auch das Format des Ausdrucks besser steuern können. Oder Sie möchten eventuell nur ein Subset der Bedienelemente oder vielleicht nur ein spezifisches Bedienelement ausdrucken.

Sie können ein SubVI erstellen, dessen Panel genauso formatiert ist, wie das VI aussehen soll. Anstatt den Menüpunkt **Ausführen»Drucken nach Abschluß** auf Ihrem VI zu wählen, können Sie es vom SubVI auswählen. Wenn ein Ausdruck gewünscht ist, kann das SubVI aufgerufen werden, und die zu druckenden Daten werden an dieses SubVI übergeben.

Durch diesen Aufbau ist es möglich, das Aussehen des Ausdrucks präzise zu steuern; mit G-Bedienelementen und -Werkzeugen können ganz einfache oder komplexe Ausdrücke erstellt werden.

Eine alternative Methode der Steuerung des Druckvorgangs ist die Verwendung der VI-Server-Druckmethoden. Mit diesen Optionen können Sie jederzeit jedes beliebige VI drucken. Weitere Informationen finden Sie in Kapitel 21, [VI-Server](#).

Verbesserung der Ausdrucke

Unter Verwendung von Transparenz und dem Farbwerkzeug können Sie die Teile der Bedienelemente verbergen, die in Ihren Ausdrucken nicht sichtbar sein sollen. Verwenden Sie z.B. Transparenz, um das Aussehen der Ränder der Bedienelemente zu vereinfachen.

In der Palette **Gestaltungselemente** können graphische Objekte zum Hervorheben von Ausschnitten des Ausdrucks verwendet werden. Sie können z.B. einen Ausschnitt der Bedienelemente mit einem Kästchen umranden, um diesen Teil vom restlichen Panel optisch abzusetzen.

Außerdem können zur benutzerspezifischen Einstellung des Ausdrucks Bitmap-Grafiken verwendet werden, indem Elemente wie z.B. das Firmenzeichen dem Bericht hinzugefügt werden.

Einstellen des Seitenlayouts

Die Einstellungen unter **Bearbeiten»Voreinstellungen...** und **Datei»Druckereinstellungen... (Seite einrichten** für den Macintosh und UNIX) wirken sich auf das Aussehen Ihrer Ausdrucke aus. Mit **VI-Einstellungen» Ausführungsoptionen** können einige Layoutoptionen aktiviert oder deaktiviert werden, die das Aussehen Ihres Ausdrucks beeinflussen. Diese Seitenlayoutoptionen, die in folgender Abbildung dargestellt sind, wirken sich ebenfalls auf das Verhalten des **Fenster drucken...** aus.

The image shows a dialog box titled "Seitenränder" (Margins). It contains the following options and settings:

- Fenster nach Ausführung des VI drucken
- Kopfzeile (Name, Datum, Seite)
- Beim Drucken Größe anpassen
- Beim Drucken Frontpanel umranden
- Seitenränder
 - Zoll
 - mm
 - Oben: 0,00
 - Links: 0,00
 - Rechts: 0,00
 - Unten: 0,00

Der Menüpunkt **Panel drucken, wenn VI Ausführung beendet** ist eine andere Möglichkeit, den programmatischen Druck einzuschalten.

Wenn **Kopfzeile drucken** gewählt wurde, erscheint eine Kopfzeile zusammen mit dem Namen des VIs, dem letzten Änderungsdatum und der Seitenzahl oben auf jeder Seite.

Wenn **Skalierung anpassen** gewählt wird, und das Panel größer als eine einzelne Seite ist, wird der Ausdruck bis zu einem Viertel der Originalgröße herabskaliert, damit das Panel auf so wenig Seiten wie möglich paßt.

Wenn **Panel mit Rahmen umgeben** gewählt wird, wird eine Umrandung um das Panel herum gedruckt.

Wenn **Seitenränder** gewählt wird, können absolute Ränder in Zoll oder Millimeter für die Ausdrücke eingestellt werden. Alle vier Ränder können separat angegeben werden. Ränder sind durch die physischen Parameter des Druckers begrenzt. Wenn Sie versuchen, kleinere Ränder einzustellen, als es mit dem Drucker möglich ist, werden faktisch die Ränder verwendet, die beim Drucker minimal möglich sind.

Verwendung von alternativen Druckmethoden

Vielleicht stellt sich heraus, daß die zuvor erwähnte Druckmethode für Ihre Anwendung nicht geeignet ist. Es gibt einige zusätzliche Techniken, mit denen verschiedene Druckprobleme angesprochen werden.

Die zuvor behandelte Druckmethode ist z.B. auf das Drucken von ganzen Datenseiten ausgerichtet. In einigen Anwendungen ist es jedoch vielleicht besser, Daten Zeile um Zeile zu drucken. Wenn ein auf Zeilen basierender Drucker an die serielle oder parallele Schnittstelle angeschlossen ist, können Sie mit seriellen Schnittstellen-VIs Text an den Drucker senden. Dabei werden im allgemeinen einige Kenntnisse der Befehlssprache des Druckers vorausgesetzt; diese Methode funktioniert jedoch sehr gut bei einer Reihe von Anwendungen, die von G-Benutzern entwickelt wurden.

Wenn die Druckergebnisse nicht Ihren Vorstellungen entsprechen, können Sie eine andere Anwendung zum Drucken Ihrer Daten verwenden, indem die Daten in einer Datei gespeichert und dann in dieser Anwendung gedruckt werden.

Weitere Einzelheiten über alternative Druckmethoden befinden sich unter [Wie kann ich einen String drucken?](#) im Anhang B, [Häufig gestellte Fragen zu G](#).

Drucken/Exportieren von Bedienelement- und VI-Beschreibungen in eine RTF- oder HTML-Datei

VI-Beschreibungen und Bedienelemente können in Rich Text-Format (RTF)- und Hyper Text Markup Language (HTML)-Dateiformate exportiert und dann gedruckt werden. Dies ist deshalb nützlich, weil Sie

RTF in die meisten Desktop-Publishing-Programme importieren können. RTF-Dateien sind auch die Quelle für Online-Hilfedateien. Das HTML-Format wird für Online-Dokumente verwendet, besonders solche, die im World Wide Web veröffentlicht werden sollen.

Wenn eine Dokumentation in eine RTF-Datei gedruckt wird, können Sie festlegen, ob eine Datei zur Verwendung als Online-Hilfedatei (Online-Referenz) oder für die Textverarbeitung erstellt wird. Im Textverarbeitungsformat werden Graphiken in das Dokument plaziert. Wenn Sie Ihre Dokumentation im RTF-Format für Online-Hilfedateien drucken, werden die Graphiken in externen Bitmap (.BMP)-Dateien gespeichert. Bei HTML-Dokumentationen werden alle Graphiken extern im JPEG (.JPG)- oder Portable Network Graphics (.PNG)-Dateiformat gespeichert.

Zum Drucken Ihrer Dokumentation im RTF- oder HTML-Format den Menüpunkt **Datei»Dokumentation drucken...** auswählen. Das folgende Dialogfeld erscheint.

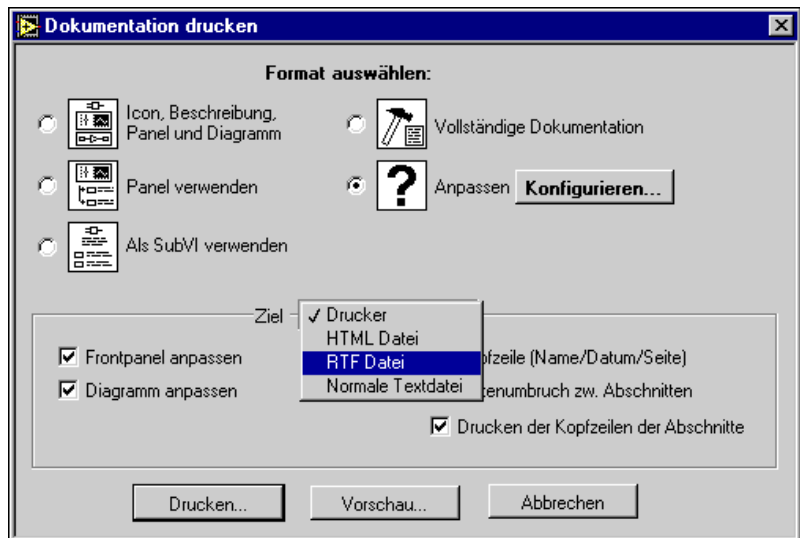


Abbildung 5-1. Dialogfeld Dokumentation drucken

Die **Ziel**-Datei solle als RTF-Datei oder HTML-Datei gespeichert werden.

Wenn Sie RTF-Datei wählen, zeigt das Dialogfeld die folgenden zusätzlichen Menüpunkte. Ihre Datei kann im Online-Hilfeformat gespeichert werden, indem Sie den Menüpunkt **Quelle für Hilfe-Compiler**

(Bilder werden extern geschrieben) auswählen, mit dem die Graphiken in einer externen Datei gespeichert werden. **Ziel** und **Farbtiefe** wählen.

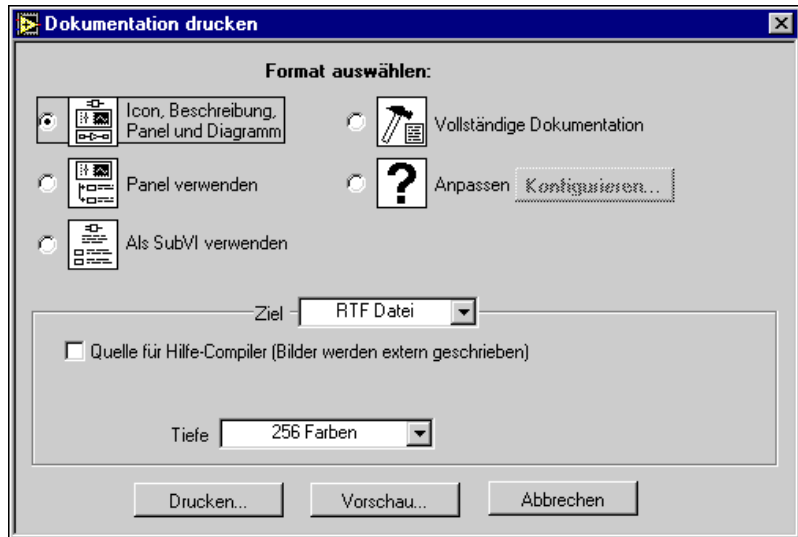


Abbildung 5-2. Dialogfeld Dokumentation drucken, RTF-Datei

Wenn Sie HTML-Datei als Format für Ihre Bedienelement- und VI-Dokumentation auswählen, erscheint das Dialogfeld wie unten dargestellt.



Abbildung 5-3. Dialogfeld Dokumentation drucken, HTML-Datei

Hier kann das Format der Graphik-Dateien sowie die Farbtiefe gewählt werden. Das JPEG-Format verfügt über gute Komprimierung, es können jedoch einige Bilddetails verloren gehen. Dieses Format funktioniert am besten bei Photos. Bei Strichvorlagen, Panels und Diagrammen kann die JPEG-Komprimierung zu verschwommenen Bildern und ungleichmäßigen Farben führen. Bilder in JPEG sind stets 24-Bit-Bilder. Wenn Sie bei Verwendung des JPEG-Formates eine geringere Farbtiefe wählen, z.B. Schwarz-Weiß, werden Bilder mit der erforderlichen Tiefe erfaßt, aber das Ergebnis ist trotzdem ein 24-Bit-Bild.

Das PNG-Format verfügt ebenfalls über gute Komprimierung, auch wenn sie nicht so gut wie beim JPEG-Format ist. Die PNG-Komprimierung führt jedoch nicht zu einem Detailverlust. Außerdem unterstützt sie 1-Bit-, 4-Bit-, 8-Bit- und 24-Bit-Bilder. Für eine geringere Bit-Tiefe läßt sich das resultierende Bild sehr viel besser als das JPEG-Format komprimieren. Das PNG-Format wurde als Ersatz für das GIF-Format entwickelt. Es bietet zwar eine Reihe von Vorteilen im Vergleich zu JPEG und GIF, wird jedoch nicht in so hohem Maße unterstützt.

Wenn die Graphiken in Ihren HTML-Dateien im GIF-Format gespeichert werden sollen, können Sie sie mit Hilfe eines von mehreren Shareware-Graphikformat-Umwandlern konvertiert werden. In diesem Fall muß HTML modifiziert werden, damit auf die GIF-Bilder mit der richtigen Erweiterung Bezug genommen werden kann. Wenn Sie GIF-Bilder auf diese Weise erstellen möchten, wird empfohlen, mit PNG-Bildern zu beginnen, da es sich dabei um verlustlose Reproduktionen der Original-Graphiken handelt. Diese können dann in GIF konvertiert werden. Aus lizenzrechtlichen Gründen wird GIF nicht direkt unterstützt; dies kann jedoch in Zukunft der Fall sein.

Wenn die Dokumentation für das VI gespeichert wird, erscheint ein Dialogfeld und fordert Sie dazu auf, den Namen des Dokuments einzugeben. Der Standardname einer jeden Dokumentation, die in RTF oder HTML gespeichert wird, enthält den Namen VI entweder mit einer `.rtf`- oder einer `.html`-Erweiterung im Dateinamen (wie z.B. `AI Clear.rtf`). (Alle Dokumentation, die auf der Plattform Windows 3.1 in HTML gespeichert wird, enthält den VI-Namen mit einer `.htm`-Erweiterung.)

Folgende Tabelle basiert auf einem Beispiel und führt die VI-Parameter und die dazugehörigen JPEG-Dateinamen auf. Angenommen Sie wählen beim Speichern eines VIs mit der Bezeichnung "Test" ein JPEG-Graphikformat; folgende JPEG-Dateien erscheinen dann als Ergebnis.

Tabelle 5-1. Resultierende JPEG-Dateien

VI-Parameter	JPEG-Name
VI-Icon	testi.jpg
VI-Anschluß	testc.jpg
VI-Frontpanel	testp.jpg
VI-Blockdiagramm	testd.jpg
VI- Hierarchie-Fenster	testh.jpg
SubVI-Icons	<i>Name des SubVIs + i.jpg</i>

Bilder für Terminals von Bedien- und Anzeigeelementen werden mit konsistenten Namen in Bilddateien gespeichert. Wenn Ihr VI daher über mehrere Eingänge desselben Typs verfügt, haben Sie letztendlich eine einzige Bilddatei für jedes der Terminals. Wenn Ihr VI z.B. über drei int32-Eingänge verfügt, wird eine einzelne ci32.jpg-Datei erstellt.

Das spezifische Benennungsschema funktioniert folgendermaßen: Der erste Buchstabe ist ein c oder ein i, und zwar je nachdem, ob das Terminal zu einem Bedien- oder Anzeigeelement gehört. Wenn das Terminal ein Array darstellt, folgt die Dimensionalität des Terminals (wie z.B. 2d) auf den Buchstaben. Dann folgt ein kurzer Name, der den Datentyp darstellt. Nachfolgend finden Sie einige Beispiele von Bilddateinamen.

Tabelle 5-2. Beispiele für Bildbenennungsschemata

Name	Terminal
ci32.jpg	Bedienelement, Int32
i2di32.jpg	Anzeigeelement, 2D-Array mit Int32s
c3dstr.jpg	Bedienelement, 3D-Array mit Strings

Programmatische RTF- und HTML-Dateien

Informationen über die programmatische Erstellung von RTF- und HTML-Dateien siehe Kapitel 21, *VI-Server*. Dieses Kapitel enthält außerdem Informationen über den Export und Import von Strings für mehrere VIs.

Lokalisierungsthemen

Durch Lokalisierung der Strings auf dem Frontpanel eines VIs können Sie auch die VI-Ausdrucke lokalisieren. Weitere Informationen über die Lokalisierung von VI-Strings finden Sie im Abschnitt *Lokalisierungsthemen* in Kapitel 29, *Portierbarkeit und Lokalisierungsthemen*.

Erstellen Ihrer eigenen Hilfe-Dateien

Mit den richtigen Entwicklungs-Tools können Sie Ihre eigenen Online-Hilfe- oder Referenz-Dokumente erstellen. Hilfe-Dokumente basieren auf formatierten Textdokumenten, und Sie können Themen in diese Dokumente eingeben, um Verknüpfungen mit Ihren VIs zu erstellen.

Die Quelldateien für alle Plattformen müssen im Windows-Hilfeformat sein. Es ist möglich, Hilfedokumente für mehrere Plattformen zu erstellen.

Wenn Sie Quelldokumente erstellen, können Sie einen Hilfe-Compiler verwenden, um ein Hilfedokument zu erstellen. Wenn Sie Hilfedateien auf mehreren Plattformen wünschen, müssen Sie den Hilfe-Compiler für die spezifische Plattform verwenden, auf der die Datei verwendet werden soll. Sie können alle nachfolgend aufgeführten Compiler dazu verwenden. Die Windows-Compiler enthalten außerdem Tools zum Erstellen von Hilfedokumenten.

- **(Windows)** RoboHelp von Blue Sky Software, 800 677 4946; für internationale Kunden 619 459 6365.
- **(Windows)** Doc-To-Help von WexTech Systems, Inc., 800 939 8324.
- **(Macintosh)** QuickView von Altura Software, 408 655 8005.
- **(UNIX)** HyperHelp von Bristol Technologies, 203 438 6969.

Wenn die Hilfedateien erstellt und kompiliert sind, können sie direkt mit einem VI verknüpft werden. Das Popup-Menü des VI-Anschlußfelds aufrufen, das mit einer Hilfedatei verknüpft werden soll, und den Menüpunkt **VI-Einstellungen»Dokumentation** auswählen. Das Feld **Hilfe Tag** wählen und die Themenverknüpfung in die Hilfedatei eingeben. Die Hilfedatei kann durch Klicken auf die Taste **Durchsuchen...** gewählt werden. Der Dateipfad erscheint im Feld **Hilfefad**.

Einrichten von VIs und SubVIs

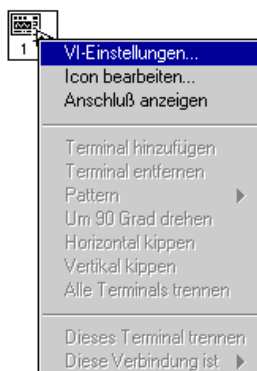
Dieses Kapitel beschreibt die Anpassungsmöglichkeiten des VI-Verhaltens mit Hilfe der Dialogfelder **VI-Einstellungen** und **Einstellungen SubVI**.

Erstellen von Popup-Panels

Ein einziges Frontpanel ist für die Darstellung verschiedener Optionen oder Anzeigen manchmal zu einschränkend. Um dieses Problem zu beseitigen, organisieren Sie Ihre VIs so, daß das VI der höchsten Ebene Optionen hoher Ebenen darstellt und SubVIs dementsprechende Optionen darstellen.

Wenn G ein SubVI aufruft, erfolgt dessen Ausführung gewöhnlich ohne Öffnen des Frontpanels. Mit Hilfe des Dialogfelds **VI-Einstellungen** oder **Einstellungen SubVI** können Sie ein SubVI dazu veranlassen, sein Frontpanel zu öffnen, wenn das SubVI aufgerufen wird, und wieder zu schließen, sobald die Ausführung des SubVIs abgeschlossen ist.

Zugriff auf das Dialogfeld **VI-Einstellungen** besteht über das Popup-Menü des VI-Icons rechts oben auf einem Frontpanel, indem Sie **VI-Einstellungen...** wählen (siehe Abbildung unten). Sie müssen sich im Bearbeitungsmodus befinden.



Dieses Dialogfeld besitzt mehrere Optionen, mit denen das Aussehen und die Verhaltensweise des VIs angepaßt werden kann. Benutzen Sie den Ring

oben im Dialogfeld, um Zugriff auf vier verschiedene Optionskategorien zu erhalten — **Ausführungsoptionen**, **Fensteroptionen** und **Dokumentation**.

VI-Einrichtungsoptionen

Dieser Abschnitt beschreibt Optionen der Ausführung, des visuellen Kontextes, der Dokumentation und Anpassung von Menüpunkten in Ihren VIs.

Ausführungsoptionen

Das Dialogfeld **VI-Einstellungen** zeigt zu Anfang Ausführungsoptionen für das aktuelle VI an. Die Parameter für **Ausführungsoptionen** sind in der folgenden Abbildung zu sehen.



Wenn Sie wollen, daß ein SubVI-Panel geöffnet wird, sobald das VI der höchsten Ebene geladen wird, aktivieren Sie die Option **Frontpanel nach dem Laden anzeigen**.

Die beiden nützlichsten Optionen dieser Seite sind **Frontpanel nach dem Aufruf anzeigen** und **Nach Abarbeitung schließen**. Wenn Sie diese Optionen für ein SubVI aktivieren, wird das Frontpanel des SubVIs automatisch geöffnet, sobald das SubVI aufgerufen wird, und automatisch geschlossen, falls es ursprünglich geschlossen war.

Die Option **Nach Öffnen ausführen** richtet ein VI so ein, daß dieses automatisch gestartet wird, sobald es geöffnet wird.

Das Setzen der Option **Unterdrückt wenn aufgerufen** entspricht der Ausführung von **Ausführen»Bei Aufruf anhalten**. Diese Debug-Option wird im Abschnitt *Debugging-Funktionen* von Kapitel 4 beschrieben, *Einrichten von VIs und SubVIs*.

Ablaufinvariante Ausführbarkeit, Priorität und Bevorzugtes Ausführungssystem sind einigermaßen fortgeschrittene Funktionen, die die Art der VI-Ausführung beeinflussen; sie werden nur in Sonderanwendungen benötigt. Diese Optionen werden jeweils in den Abschnitten *Ablaufinvariante Ausführung* und *VI-Einstellungen und Priorität* von Kapitel 26, *Überblick über das G-Ausführungssystem*, beschrieben. Eine andere fortgeschrittene Ausführungsfunktion ist *Multithreading*, wo die Ausführung mehrerer VIs gleichzeitig abläuft, während die VIs weiterhin auf Maus- und Tastatureingaben reagieren und die CPU-Zeit gleichmäßig unter den Ausführungs-Threads freigegeben ist. Diese Funktion wird auch in Kapitel 26, *Überblick über das G-Ausführungssystem*, beschrieben.

Mit den übrigen Optionen kann das VI bei der Ausführung gedrückt werden und das Druckbild angepaßt werden. Einzelheiten des programmgesteuerten Drucks werden im Abschnitt *Programmatisches Drucken* von Kapitel 5, *Drucken und Dokumentieren von VIs*, behandelt.

Fensteroptionen

Die Fensteroptionen gelten für das VI bei dessen Ausführung, nicht im Bearbeitungsmodus. Mit diesen Optionen wird die Fähigkeit eines Benutzers gesteuert, mit dem Programm zu interagieren; dies geschieht durch die Beschränkung des Zugriffs auf G-Funktionen und durch Ändern des Aussehens und der Verhaltensweise des Fensters. Sie können Aussehen und Funktionsweise eines VIs wie ein Dialogfeld konfigurieren, da der Benutzer nicht mit anderen Fenstern interagieren kann, wenn dieses Fenster offen ist. Sie können auch die Rollbalken und die Werkzeugpalette ausblenden und ein Fenster zentrieren oder automatisch an die Bildschirmgröße anpassen.

Sie können den VI-Fenstertitel anpassen, um ihm eine deskriptive Bezeichnung als dem VI-Dateinamen zu geben. Diese Funktion ist insbesondere bei VIs von Bedeutung, wenn ein VI-Fenstertitel ins Deutsche übersetzt wird. Um den VI-Fenstertitel zu ändern, während Sie das VI bearbeiten, wählen **VI-Einstellungen** und **Fensteroptionen** aus dem oberen Ring (siehe nachfolgende Abbildung). Deaktivieren Sie die

Option **Gleich wie VI Name**, und geben Sie den gewünschten VI-Fenstertitel ein.



Die Option **Dialogfeld** hält den Benutzer, wie bei einem Systemdialogfeld, von der Interaktion mit anderen Fenstern ab, während dieses offen ist.

Beachten Sie, daß dieser Menüpunkt Sie unter UNIX nicht daran hindert, Fenster anderer Anwendungen in den Vordergrund zu bringen. Es ist nicht möglich, ein Fenster vor allen anderen Fenstern zu behalten.

Run-Time Popup-Menü zulassen bestimmt, ob Objekte auf diesem Frontpanel im Ausführungsmodus ein Popup-Menü Datenoperationen anzeigen.

Wenn Sie den Menüpunkt **<Return> Boolesch hervorheben** wählen, wird ein beliebiges Boolesches Objekt hervorgehoben, das gegenwärtig mit der <Eingabe>- (**Windows** und **HP-UX**) oder <Return>-Taste (**Macintosh** und **Sun**) verknüpft ist. Die Tastenbelegung mit Bedienelementen erfolgt über das Dialogfeld "Tastenbelegung", das im Abschnitt *Optionen der Tastenbelegung für Bedienelemente* von Kapitel 8, *Einführung in Frontpanel-Objekte*, beschrieben ist.

Mit dem Menüpunkt **Auf Fenstergröße anpassen** wird ein VI-Panel automatisch in die Bildschirmanzeige eingepaßt, wenn das VI ausgeführt wird. Die ursprüngliche Größe und Position wird nirgends gespeichert, so daß das VI an derselben Stelle bleibt, wenn Sie zum Bearbeitungsmodus zurückwechseln.

Automatisch Zentrieren zentriert das Frontpanel automatisch auf dem Computerbildschirm.

Andere Menüpunkte schalten verschiedene Fensterfunktionen ein oder aus. Durch Aufheben der Auswahl verbergen Sie einzelne Tasten, und durch Aufheben der Auswahl des Menüpunkts **Werkzeugpalette anzeigen** verbergen Sie die gesamte Werkzeugpalette.

Protokoll aktivieren/Drucken nach Abschluß aktiviert oder deaktiviert die automatische Datenprotokollierung (siehe Abschnitt *Datenprotokollierung auf dem Frontpanel* in Kapitel 4, *Ausführung und Debugging von VIs und SubVIs*) und den programmgesteuerten Druck (siehe Abschnitt *Programmatisches Drucken* in Kapitel 5, *Drucken und Dokumentieren von VIs*).

Um den VI-Fenstertitel programmatisch zu ändern, ziehen Sie Kapitel 21, *VI-Server*, zu Rate.

Durch Aufheben der Auswahl **Beim Start automatische Verwaltung der Menüs** wird die Run-Time-Menüleiste deaktiviert, bis Sie dazu bereit sind, die Menüauswahl mit Hilfe der Auswahlfunktion "Menüauswahl bekommen" zu handhaben. Weitere Informationen hierzu finden Sie unter *Handhabung der Menüauswahl* weiter unten in diesem Kapitel.



Vorsicht

Wenn Sie die Menüleiste und die Werkzeugpalette verbergen, besteht keine visuelle Möglichkeit, vom Ausführungsmodus zum Bearbeitungsmodus zurückzuwechseln. Benutzen Sie die Zugriffstasten <Strg-m> (Windows), <Befehlstaste-m> (Macintosh), <Metataste-m> (Sun) oder <Alt-m> (HP-UX), die dem Menüpunkt Ausführen»In Bearbeitungsmodus wechseln entspricht, um zum Bearbeitungsmodus zurückzuwechseln, wo die Menüleiste und die Werkzeugpalette sichtbar sind.

Dokumentationsoptionen

Das Dialogfeld **VI-Einstellungen»Dokumentation**, das in der folgenden Abbildung zu sehen ist, enthält Menüpunkte, die Einträge im Historie-Fenster betreffen, welches die Entwicklungshistorie des VIs anzeigt.

Weitere Informationen über das Historie-Fenster finden Sie im Abschnitt *VI-Historiefenster* von Kapitel 27, *Verwalten Ihrer Anwendungen*.



Um Historie-Optionen im Menü **Dokumentation** zu benutzen, muß die Auswahl von **Standard zur Historie verwenden (im Dialog “Voreinstellungen”)** aufgehoben werden. Wenn Sie diesen Menüpunkt wählen, werden statt dessen die Historie-Voreinstellungen verwendet. Die Historie-Voreinstellungen sind identisch, mit der Ausnahme, daß sie die Standardwerte festlegen, mit denen ein neues VI erstellt wird, obgleich die Menüpunkte für Dokumentation unter “VI-Einstellungen” nur für das aktuelle VI gelten.

Wenn Sie **Eintrag bei jedem Speichern hinzufügen** aktivieren, wird jedesmal, wenn Sie das VI speichern, ein Eintrag in der VI-Historie vorgenommen. Wenn Sie im Feld **Kommentar** des Historie-Fensters keine Kommentare eingeben, wird nur eine Kopfzeile in die Historie eingetragen. Die Kopfzeile enthält die Revisionsnummer, Uhrzeit und Datum und den VI-Namen, wenn der Menüpunkt im Dialogfeld **Voreinstellungen»Historie** aktiviert ist.

Wenn Sie die Option **Nach dem Schließen nach Kommentar fragen** aktivieren, wird das Historie-Fenster angezeigt. Sie können einen Kommentar eingeben, wenn Sie ein VI schließen, das sich geändert hat, seitdem Sie es geladen haben, selbst wenn Sie die Änderungen bereits gespeichert haben.

Wenn Sie die Option **Nach dem Speichern nach Kommentar fragen** aktivieren, wird das Historie-Fenster beim Speichern angezeigt. Dies ist nützlich, wenn Sie es vorziehen, Ihre Änderungen nachträglich mit Kommentaren zu versehen, anstatt sie während der Bearbeitung einzugeben. Wenn Sie diese Option nicht setzen, können Sie die VI-Historie erst dann ändern, wenn die Speicherung nach Auswahl von **Speichern** abgeschlossen ist.

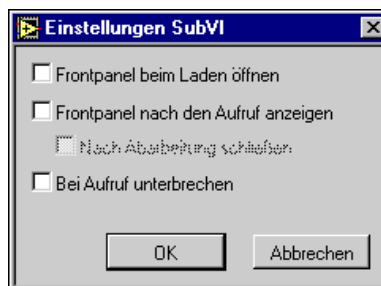
**Hinweis**

Wenn Sie lediglich Änderungen in der Historie vorgenommen haben, werden Sie beim Speichern oder Schließen eines VIs nicht aufgefordert, einen Kommentar einzugeben.

Die Menüpunkte auf der rechten Seite des Dialogfeldes **VI-Einstellungen»Dokumentation** legen den Zugriff auf Online-Hilfdateien fest, die Sie erstellt haben. Klicken Sie in das Feld **Hilfe Tag**, und geben Sie ein bei diesem VI anzuzeigendes Thema ein. Geben Sie anschließend den Pfad der Hilfdatei in das Feld **Hilfepfad** ein, oder klicken Sie auf die Taste **Durchsuchen...**, um die Datei zu suchen. Der Name und Pfad der Datei werden im Feld **Hilfepfad** angezeigt. Sobald Sie diese Menüpunkte gesetzt haben, klicken Sie auf das Icon der Online-Hilfe am unteren Rand des Dialogfeldes “Hilfe”, um auf die gewählte Hilfdatei zuzugreifen. Informationen zum Erstellen eigener Hilfdateien finden Sie in Kapitel 5, *Drucken und Dokumentieren von VIs*.

Das Dialogfeld “Einstellungen SubVI”

Das folgende Dialogfeld erscheint, wenn Sie im Blockdiagramm **Einstellungen SubVI...** aus dem Popup-Menü eines SubVI-Knotens wählen.



Bei diesen Menüpunkten handelt es sich um ein Subset der Optionen im Dialogfeld “VI-Einstellungen”. Der Unterschied zwischen ihnen besteht darin, daß mit dem Dialogfeld “Einstellungen SubVI” Optionen angegeben werden, die sich auf einen bestimmten Aufruf eines SubVIs beziehen. Sie

können jedoch mit Hilfe der SubVI-Ausführungsoptionen unter “VI-Einstellungen” das Verhalten aller an dieses VI ergehenden Aufrufe festlegen.

Anpassen der Menüleiste

Sie können die Menüleiste für jedes VI, das Sie erstellen, anpassen, obgleich Sie die angepaßten Menüleisten nur bei der VI-Ausführung sehen können.

Die Anpassung von Menüs umfaßt zwei Teile – die Erstellung und die Handhabung von Menüs.

Sie können angepaßte Menüs auf zweierlei Art erstellen – statisch zum Zeitpunkt der Bearbeitung oder dynamisch bei der Ausführung. Mit dem Menü-Editor können Sie eine angepaßte Menüvorlage statisch erstellen und in einer Datei, die als Run-Time-Menüdatei (RTM) bezeichnet wird, speichern (weitere Informationen hierzu finden Sie im Abschnitt über den *Menü-Editor* in diesem Kapitel). Der Menü-Editor läßt Sie auch eine RTM-Datei mit dem fraglichen VI verknüpfen. Wenn das VI ausgeführt wird, lädt es das Menü aus der entsprechenden RTM-Datei. Sie können Menüpunkte programmatisch, bei der Ausführung und mit Hilfe von G-Funktionen über das Diagramm (weitere Informationen hierzu finden Sie im Abschnitt *Dynamische Menüfunktionen* in diesem Kapitel) einfügen, löschen und ändern.

Die Handhabung der angepaßten Menüpunkte erfolgt durch das Diagramm. Jeder Menüpunkt besitzt eine eindeutige Stringkennung, bei der nicht nach Groß- und Kleinschreibung unterschieden wird und die als “Tag” bezeichnet wird. Wenn Sie einen Menüpunkt wählen, können Sie das Tag des gewählten Menüpunkts programmatisch mit Hilfe der G-Funktion **Menüauswahl bekommen** abrufen (weitere Informationen hierzu finden Sie im Abschnitt *Handhabung der Menüauswahl* in diesem Kapitel). Je nach dem Tag-Wert können Sie für jeden Menüpunkt im Diagramm einen Handler festlegen.



Hinweis

Angepaßte Menüleisten werden auch als Ausführungsmenüs (oder Run-Time-Menüs) bezeichnet.

Menü-Editor

Der Menü-Editor bietet eine Schnittstelle zur Erstellung und Bearbeitung von RTM-Dateien und ihrer Verknüpfung mit dem betreffenden VI. Den Menü-Editor eines VIs rufen Sie durch Wählen von **Bearbeiten» Menü bearbeiten...** auf. Gegenwärtig können Sie 3 Arten von Menüs mit einem VI verknüpfen, und zwar die Typen **Standard**, **Minimal** und **Anpassen** (siehe hierzu Abbildung 6-1). Die Option "Standard" liefert das Standardmenü, die Option "Minimal" entfernt Optionen aus dem Standardmenü, die weniger häufig gebraucht werden. Und die Option "Anpassen" ermöglicht Ihnen die Verknüpfung einer RTM-Datei mit dem VI. Die Optionen "Standard" und "Minimal" können nicht bearbeitet werden.

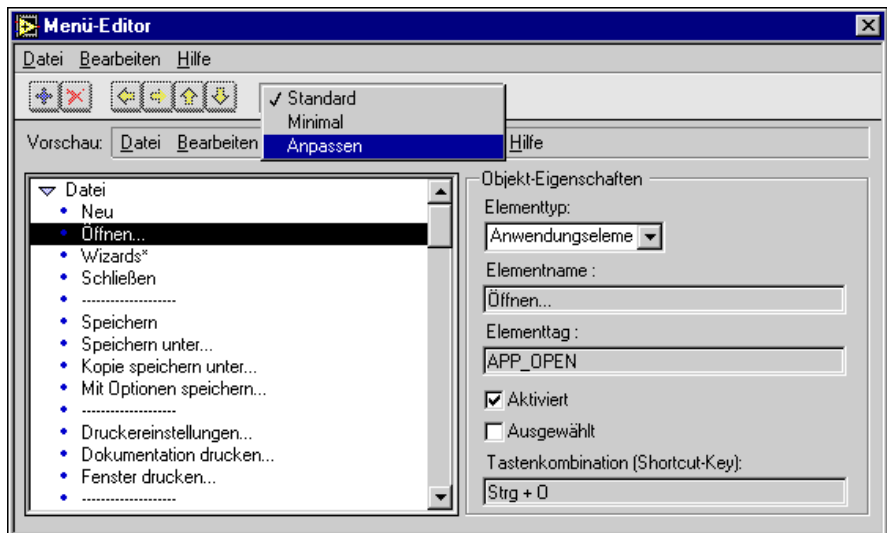


Abbildung 6-1. Menü-Editor






Der Menü-Editor zeigt auf der linken Seite die Menühierarchie und auf der rechten Seite die Eigenschaften des in der Menühierarchie gewählten Menüpunkts an. Im Bereich **Vorschau** können Sie das Menü anzeigen, wie es bei der Ausführung erscheinen wird. Durch Klicken auf die links abgebildete Taste der Symbolleiste oder Auswahl der Einfügeoptionen aus dem Menü **Bearbeiten** können Sie einen neuen Menüpunkt hinzufügen.

Es gibt drei Arten von Menüpunkten: *Angepaßtes Element*, *Anwendungsselement* oder *Trennzeichen*. Der Typ eines Menüpunkts kann durch Auswahl aus dem Ring **Elementtyp** geändert werden.

Ein Menüpunkt vom Typ “**Angepaßtes Element**” läßt sich programmatisch im Diagramm handhaben. Er besitzt einen Namen, und zwar das String, das im eigentlichen Menü erscheint, und ein Tag, bei dem es sich um eine eindeutige Stringkennung handelt, bei der nicht nach Groß- und Kleinschreibung unterschieden wird. Ein Tag identifiziert den angepaßten Menüpunkt im Diagramm. Wenn Sie einen Namen eingeben, wird dieser zur Erleichterung der Bearbeitung als Tag dupliziert. Sie können das Tag jedoch jederzeit ändern, um es vom Namen selber zu unterscheiden. Damit ein Menüpunkt gültig ist, muß dessen Tag einen eindeutigen Wert haben. Ungültige Menüpunkte werden als ??? angezeigt. Der Menü-Editor stellt stets sicher, daß die Tags in einer bestimmten Menühierarchie eindeutig sind, indem der Wert falls notwendig hochgezählt wird.

Ein angepaßter Menüpunkt kann durch Setzen der jeweiligen Attribute aktiviert oder deaktiviert bzw. ausgewählt oder nicht ausgewählt werden. Sie können durch die Eingabe einer entsprechenden Tastenkombination einen Shortcut (Zugriffstaste) für einen angepaßten Menüpunkt festlegen. Die Tastenkombination hat das Format `<Menütaste-[Umschalttaste]-Taste>`.

-  **Hinweis** *PC-Benutzer können in Namen in Menüpunkten Unterstriche verwenden, um die Menüauswahl mit Hilfe der ALT-Taste zu erleichtern. Um beispielsweise einen Menüpunkt namens “Datei/Beenden” mit ALT+D+B auszuwählen, geben Sie die Namen der Menüpunkte mit “_Datei” und “_Beenden” an.*
-  **Hinweis** *Umschalttasten und Funktionstasten in Kurzbefehlen sind unter dem Betriebssystem Macintosh OS 7 oder einer älteren Version nicht sichtbar. Um Menüs auf Macintosh OS 7 portierbar zu gestalten, vermeiden Sie den Gebrauch der Umschalttaste oder von Funktionstasten in Kurzbefehlen.*
-  **Hinweis** *Auf der Macintosh-Plattform müssen alle Menüpunkte auf der Menüleiste Pull-down-Menüs sein. Benutzer von Windows 95/NT und UNIX können jedoch in der Menüleiste auswählbare Menüpunkte erstellen. Vermeiden Sie daher auf dem Macintosh auswählbare Menüpunkte in Menüleisten.*

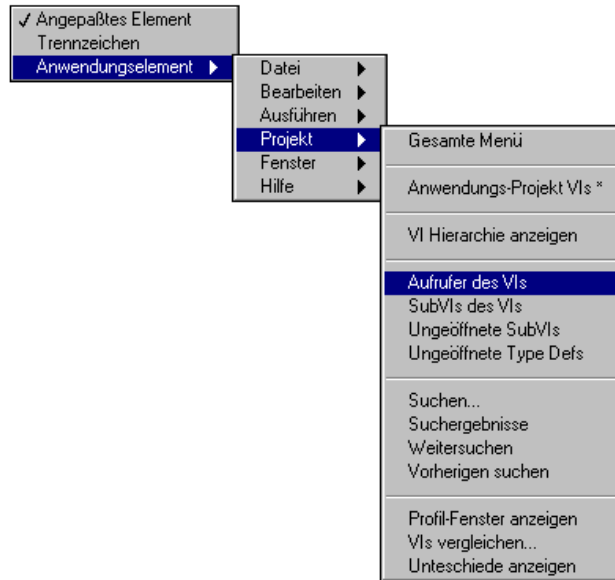


Abbildung 6-2. Auswahl eines Anwendungselements

Eine Option vom Typ **Anwendungselement** wird von der Software der G-Sprache zur Verfügung gestellt. Diese Menüpunkte sind Bestandteil des Standardmenüs. Um einen bestimmten Punkt auszuwählen, wählen Sie **Elementtyp»Anwendungselement**, und folgen Sie der in der Abbildung gezeigten Hierarchie. Mit diesem Prozeß können Sie einzelne Punkte oder ganze Untermenüs hinzufügen. Anwendungsmenüpunkte werden implizit von der Software gehandhabt. Die Tags dieser Punkte erscheinen nicht in Diagrammen. Weder Name, noch Tag oder Attribute eines Anwendungsmenüpunkts können geändert werden. Die Software reserviert Tags, die mit APP_ anfangen, für Anwendungsmenüpunkte. Die Tags der Anwendungsmenüpunkte sind in Tabelle 6-1 am Ende dieses Kapitels einzeln aufgelistet.

Ein Menüpunkt vom Typ **“Trennzeichen”** fügt eine Trennlinie in das Menü ein. Für eine Trennlinie können keine Attribute gesetzt werden.

Durch Klicken auf die Pfeiltasten in der Symbolleiste, durch Verwendung der Optionen zur Manipulation der Hierarchie im Menü **Bearbeiten** oder einfach durch Ziehen und Ablegen lassen sich Menüpunkte in der Menühierarchie verschieben. Hierarchien können durch Klicken auf die Untermenüsymbbole oder Auswahl der Zusammenklapp- bzw. Erweiterungsoption im Menü **Bearbeiten** für eine bequeme Anzeige zusammengeklappt oder erweitert werden.

Mit den Optionen aus dem Menü **Datei** können RTM-Dateien geladen oder gespeichert werden. Verwenden Sie die Optionen im Menü **Bearbeiten** zum Ausschneiden, Kopieren und Einfügen von Menüpunkten (weitere Informationen hierzu finden Sie im Abschnitt *Menüoptionen des Menü-Editors* in diesem Kapitel). Hilfe zu einem beliebigen Element im Menü-Editor erhalten Sie, indem Sie **Hilfe»Hilfe anzeigen** wählen und auf das betreffende Element zeigen.

Menüoptionen des Menü-Editors

Im Menü **Datei** stehen die folgenden Optionen zur Verfügung.

Neu — Erstellt eine neue Ausführungsmenü-Datei und fordert zur Speicherung der eventuell zuvor bearbeiteten RTM-Datei auf.

Öffnen — Öffnet eine bestehende Ausführungsmenü-Datei. Durch Wählen einer RTM-Datei schaltet der Menü-Editor auf den Typ “Custom” um.

Speichern — Speichert das aktuelle Ausführungsmenü.

Speichern unter — Speichert das aktuelle Ausführungsmenü in einer anderen Datei.

Schließen — Beendet den Menü-Editor.

Im Menü **Bearbeiten** stehen die folgenden Optionen zur Verfügung:

Ausschneiden — Löscht den gegenwärtig gewählten Menüpunkt oder Text und kopiert ihn in die Zwischenablage.

Kopieren — Kopiert den gegenwärtig gewählten Menüpunkt oder Text in die Zwischenablage.

Gesamtes Menü kopieren — Kopiert die gesamte Menühierarchie in die Zwischenablage.

Einfügen — Fügt den Inhalt der Zwischenablage ein.

Einblenden/Ausblenden — Erweitert den gegenwärtig gewählten Untermenüpunkt oder klappt ihn zusammen.

Alle einblenden — Erweitert alle Untermenüpunkte.

Ausblenden alle — Klappt alle Untermenüpunkte zusammen.

Benutzerspezif. Element einfügen — Fügt einen benutzerspezifischen Menüpunkt nach dem gewählten Menüpunkt ein.

Separator einfügen — Fügt eine Trennlinie nach dem gewählten Menüpunkt ein.

Anwendungselement einfügen — Ruft ein Untermenü auf, aus dem einer der verfügbaren Menüpunkte der Anwendung gewählt werden kann.

Objekt löschen — Löscht den gewählten Menüpunkt.

Übergeordnetes Element erstellen — Wandelt die nachfolgenden Menüpunkte in ein Untermenü des gewählten Menüpunkts um.

Untergeordnetes Element erstellen — Wandelt den gewählten Menüpunkt in einen Untermenüpunkt des vorhergehenden Menüpunkts um.

Element nach oben bewegen — Verschiebt den gewählten Menüpunkt in der Hierarchie nach oben. Besitzt der gewählte Menüpunkt untergeordnete Menüpunkte, so werden diese Menüpunkte mitverschoben.

Element nach unten bewegen — Verschiebt den gewählten Menüpunkt in der Hierarchie nach unten. Besitzt der gewählte Menüpunkt untergeordnete Menüpunkte, so werden diese mitverschoben.

Symbolleisten-Optionen des Menü-Editors

Die folgenden Optionen stehen in der Symbolleiste zur Verfügung.



Fügt einen benutzerspezifischen Menüpunkt im Anschluß an den gewählten Menüpunkt ein.



Löscht den gewählten Menüpunkt.



Wandelt die dem gewählten Menüpunkt folgenden Menüpunkte in Untermenüpunkte des gewählten Menüpunkts um.



Wandelt den gewählten Menüpunkt in einen Untermenüpunkt des vorhergehenden Menüpunkts.



Verschiebt den gewählten Menüpunkt nach oben. Besitzt der gewählte Menüpunkt Untermenüpunkte, so werden diese mitverschoben.



Verschiebt den gewählten Menüpunkt nach unten. Besitzt der gewählte Menüpunkt Untermenüpunkte, so werden diese mitverschoben.

Handhabung der Menüauswahl



Dieser Abschnitt enthält einen Überblick über die Funktionen **Menüauswahl bekommen** und **Aktivieren der Menüüberwachung**, die daran anschließend im Abschnitt *Funktionen zur Handhabung der Menüauswahl* einzeln beschrieben werden. Die Funktionen bedienen sich zur Identifikation von Menüs eines Refnum-Werts. Die Menü-Refnum eines VIs wird durch das Menü des konstanten aktuellen VIs identifiziert, das sowohl als Icon links und in den Abbildungen 6-3 und 6-4 zu sehen ist.

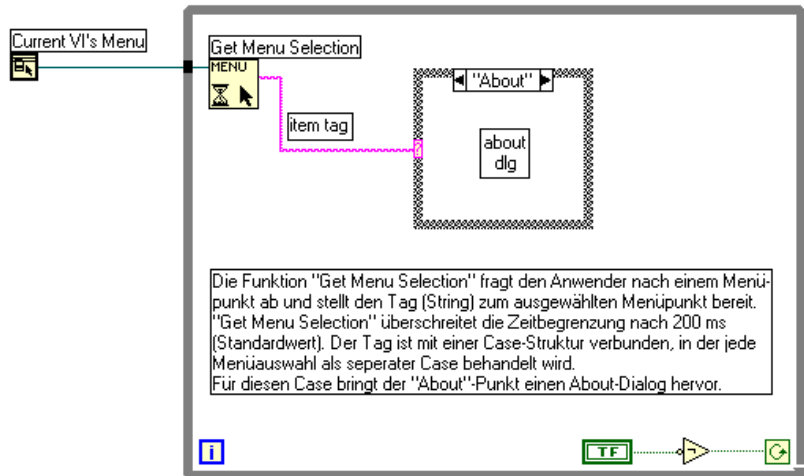


Abbildung 6-3. Funktion "Menüauswahl bekommen"

Sobald Sie einen Menüpunkt ausgewählt haben, können Sie erst dann einen anderen auswählen, wenn die Funktion **Menüauswahl bekommen** den ersten Menüpunkt gelesen hat. Sobald jedoch ein Menüpunkt gelesen worden ist, kann ein neuer Menüpunkt ausgewählt werden, selbst dann, wenn der vorherige Menüpunkt noch nicht vollständig bearbeitet worden ist. Dieses Vorgehen ist jedoch nicht üblich, wenn die Bearbeitung einer Auswahl länger dauert. Unter diesen Umständen wird **Menüauswahl bekommen** im Menüblockiermodus aufgerufen, wobei die Menüfassung nach dem Lesen einer Auswahl blockiert wird. Mit der Funktion

Aktivieren der Menüüberwachung wird das Menü aktiviert, sobald die Auswahl bearbeitet worden ist (siehe nachstehende Abbildung).

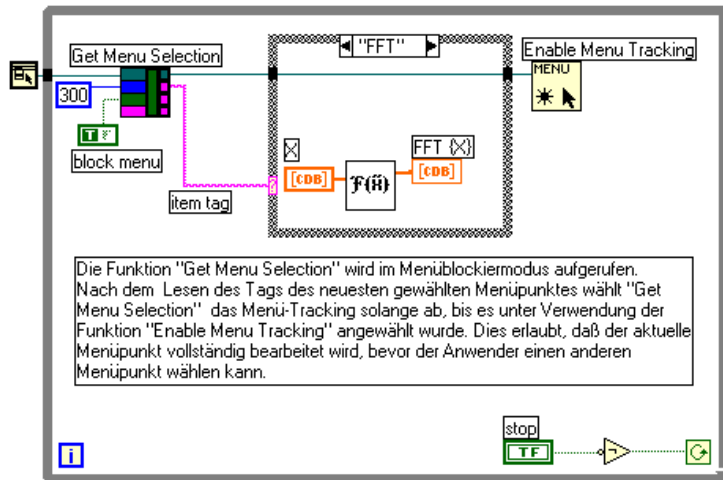


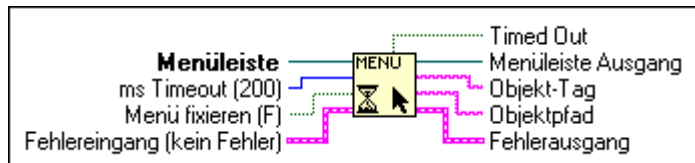
Abbildung 6-4. Funktion "Aktivieren der Menüüberwachung"

Funktionen zur Handhabung der Menüauswahl

Die folgenden Funktionen zur Handhabung der Menüauswahl stehen zur Verfügung.

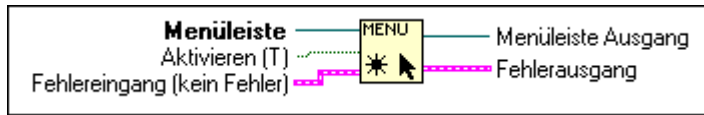
Menüauswahl bekommen

Gibt den **Objekt-Tag**-Wert des zuletzt gewählten Menüpunktes zurück, optional nach Ablauf von vorgegebenen `ms Timeout`. **Objektpfad** ist ein String, der die Position des Menüpunkts in der Menühierarchie beschreibt, wobei es sich um das Format einer Liste von Menü-Tags handelt, die durch Doppelpunkt (:) getrennt werden. Ist **Menü fixieren** auf True gesetzt, wird die Menüauswahl nach dem Lesen eines Menüpunkt-Tags blockiert.



Aktivieren der Menüüberwachung

Aktiviert oder deaktiviert die Erfassung der Menüauswahl.



Die Funktionen bedienen sich zur Identifikation von Menüs eines Refnum-Wertes. Die Menü-Refnum eines VIs wird durch das Menü des konstanten aktuellen VIs ermittelt (siehe Icon links). Menüpunkte werden durch ein Tag (String) und manchmal durch einen Pfad (String) identifiziert, bei dem es sich um eine Liste der Menüpunkt-Tags (getrennt durch Doppelpunkte) aus der Baumstruktur des Menüs vom Stamm bis zum betreffenden Menüpunkt handelt.

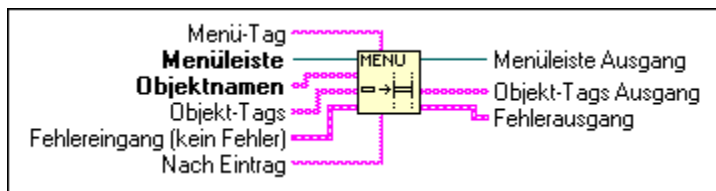
Dynamische Menüfunktionen

Während der Ausführung eines VIs können Sie die Menüpunkte in der VI-Menüleiste mit Hilfe der Menüverwaltungsfunktionen **Menüeinträge einfügen**, **Menüeinträge löschen**, **Menüpunkt-Info bekommen**, **Set Menu Item Info** und **Menü-Kurzinformation bekommen** ändern. Diese Funktionen sind auf der Palette **Anwendungssteuerung** zu finden, auf die von **Funktionen**»**Anwendungssteuerung** aus Zugriff besteht. Die Funktionen identifizieren Menüs anhand ihres Refnum-Wertes. Die Menü-Refnum eines VIs wird durch das Menü des konstanten aktuellen VIs ermittelt (siehe Icon links).



Menüeinträge einfügen

Fügt Menüpunkte in eine Menüleiste oder ein Untermenü einer Menüleiste ein.



Menü-Tag spezifiziert das Untermenü, in das die Menüpunkte eingefügt wurden. Wird die Option **Menü-Tag** nicht spezifiziert, werden die Menüpunkte in die **Menüleiste** eingefügt.

Objektnamen und **Objekt-Tags** identifizieren die in das Menü einzufügenden Menüpunkte. Der Typ der **Objektnamen** und **Objekt-Tags** kann ein String-Array (zum Einfügen mehrerer Menüpunkte) oder lediglich ein String (zum Einfügen eines einzigen Menüpunkts) sein. Die Eingangsverbindung kann entweder aus den **Objektnamen** oder **Objekt-Tags** bestehen, wodurch die Namen und die Tags dieselben Werte erhalten. Ist es erforderlich, daß sich Name und Tag jedes Menüpunkts voneinander unterscheiden, müssen Sie für **Objektnamen** und **Objekt-Tags** separate Werte in der Eingangsverbindung festlegen.

Nach Eintrag spezifiziert die Position, an der die Menüpunkte eingefügt werden. **Nach Eintrag** kann ein Tag (String) eines bestehenden Menüpunkts oder ein Positionsindex (Null-basierter Integer-Wert) im Menü sein. Um Menüpunkte am Anfang des Menüs einzufügen, verbinden Sie die Option **Nach Eintrag** mit einer Zahl, die kleiner als 0 ist. Um Menüpunkte am Ende des Menüs einzufügen, geben Sie bei der Verbindung eine Zahl ein, die größer als die Anzahl der Menüpunkte im Menü ist. Sie können Anwendungsmenüpunkte einfügen, indem Sie die Anwendungs-Tags aus der Tabelle 6.1 am Ende dieses Kapitels verwenden. Mit dem Anwendungs-Tag APP_SEPARATOR können Sie eine Trennlinie einfügen. Die Funktion stellt stets sicher, daß die Tags in einer bestimmten Menühierarchie eindeutig sind, indem der Wert falls notwendig hochgezählt wird.

Objekt-Tags Ausgang gibt die eigentlichen Tags der eingefügten Menüpunkte zurück. Falls weder **Menü-Tag** noch **Nach Eintrag** (Tag) ermittelt werden können, gibt die Funktion einen Fehler zurück.

Bei dem Beispiel in den nachstehenden Abbildungen wird eine aus zwei Menüs höchster Ebene (**Datei** und **Bearbeiten**) und einem Untermenü (**Test**) bestehende Menü-Baumstruktur dynamisch erstellt.

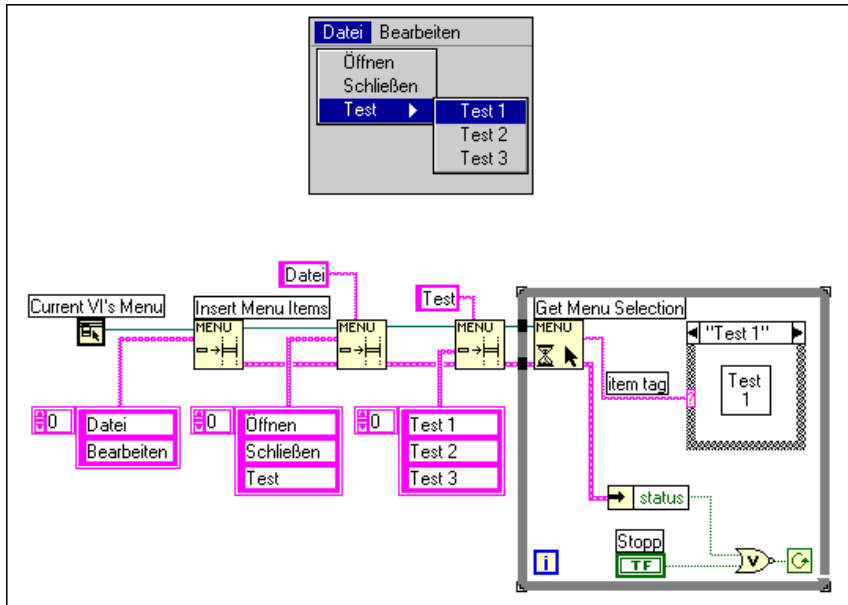
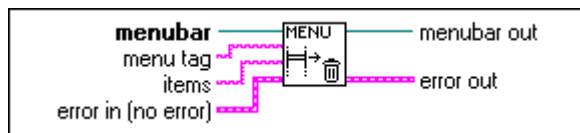


Abbildung 6-5. Dynamisches Einfügen eines Menüs

Menüeinträge löschen

Löscht Menüpunkte aus der Menüleiste oder einem Untermenü der Menüleiste.

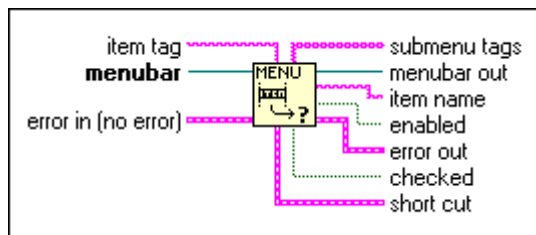


Wird **Menü-Tag** spezifiziert, werden die Menüpunkte aus dem durch **Menü-Tag** spezifizierten Untermenü gelöscht, andernfalls werden die Menüpunkte aus der Menüleiste gelöscht. Kann **Menü-Tag** oder einer der angegebenen Menüpunkte nicht gefunden werden, gibt die Funktion einen Fehler zurück.

Objekte kann ein Tag (String) eines bestehenden Menüpunkts, ein Tag-Array bestehender Menüpunkte, ein Positionsindex (Null-basierter Integer-Wert) eines Menüs oder ein Array von Positionsindizes von Menüpunkten im Menü sein. Wenn Sie keine Menüpunkte im Diagramm verbinden, werden alle Menüpunkte der Menüleiste gelöscht. Enthält ein angegebener Menüpunkt ein Untermenü, wird dieses und dessen gesamter Inhalt automatisch mitgelöscht. Sie können Anwendungsmenüpunkte über die Anwendungs-Tags löschen, die in Tabelle 6-1 am Ende dieses Kapitels aufgelistet sind. Da Trennlinien keine eindeutigen Tags besitzen, ist es am besten, sie über ihre Positionsindizes zu löschen.

Menüpunkt-Info bekommen

Gibt die Attribute des angegebenen Menüpunkts über das betreffende Tag zurück.

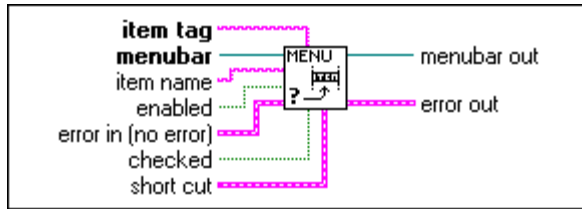


Attribute von Menüpunkten sind **Objektname** (der im Menü erscheinende String), **Aktiviert** ("False" bedeutet, daß der Menüpunkt deaktiviert ist), **Ausgewählt** (gibt an, ob der Menüpunkt durch ein Häkchen als aktiv gekennzeichnet wird) und **Shortcut** (Zugriffstaste). Gibt es ein Untermenü zu dem Menüpunkt, werden die Tags der Untermenüpunkte in **submenu tags** als String-Array zurückgegeben. Ist **Objekt-Tag** unverbunden, werden die Menüpunkte der Menüleiste zurückgegeben. Ist Objekt-Tag ungültig, wird ein Fehler zurückgegeben.

Set Menu Item Info

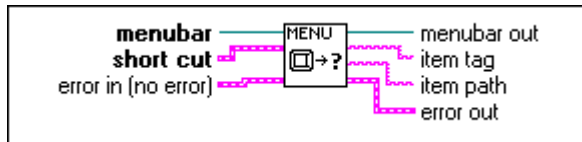
Setzt die Attribute eines durch Menü- und Menüpunkt-Tag spezifizierten Menüpunkts. Attribute von Menüpunkten sind **Objektname** (der im Menü erscheinende String), **Aktiviert** ("False" bedeutet, daß der Menüpunkt deaktiviert ist), **Ausgewählt** (gibt an, ob der Menüpunkt durch ein Häkchen als aktiv gekennzeichnet wird) und **Shortcut** (Zugriffstaste).

Unverbundene Attribute bleiben unverändert. Ist **Objekt-Tag** ungültig, wird ein Fehler zurückgegeben.



Menü-Kurzinformation bekommen

Gibt den Menüpunkt zurück, auf den über eine bestimmte Tastenkombination zugegriffen werden kann.



Objektpfad ist ein String, dessen einzelne Menüpunkt-Tags durch Doppelpunkt (:) getrennt werden.

Shortcut besteht aus einem String (Taste) und einem Booleschen Operator (der festlegt, ob die Umschalttaste mitverwendet wird oder nicht).

Tags von Menüpunkten der Anwendung

Die folgende Tabelle enthält eine Liste der Anwendungsmenüpunkte, die in ein Menü eingefügt werden können, sowie die zugehörigen Anwendungs-Tags.

Tabelle 6-1. Tags von Anwendungselementen

Anwendungselement	Tag
Datei	APP_FILE
Neu	APP_NEW
Öffnen...	APP_OPEN
Schließen	APP_CLOSE
Speichern	APP_SAVE

Tabelle 6-1. Tags von Anwendungselementen (Fortsetzung)

Anwendungselement	Tag
Speichern unter...	APP_SAVEAS
Kopie speichern unter...	APP_SAVE_A_COPY_AS
Mit Optionen speichern...	APP_SAVE_WITH_OPTIONS
Druckereinstellungen...	APP_PRINTER_SETUP
Dokumentation drucken...	APP_PRINT_DOCUMENTATION
Fenster drucken...	APP_PRINT_WINDOW
VI-Bibliothek bearbeiten...	APP_EDIT_VI_LIBRARY
Vorlage bearbeiten...	APP_EDIT_TEMPLATE
Massenkompilierung...	APP_MASS_COMPILE
CVI FP Datei konvertieren...	APP_CONVERT_CVI
Beenden	APP_EXIT
Bearbeiten	APP_EDIT
Rückgängig	APP_UNDO
Wiederherstellen	APP_REDO
Ausschneiden	APP_CUT
Kopieren	APP_COPY
Einfügen	APP_PASTE
Löschen	APP_CLEAR
Bild aus Datei importieren...	APP_IMPORT_PICT_FROM_FILE
Voreinstellungen...	APP_PREFERENCES
Anwendername...	APP_USER_NAME
Paßwort-Cache löschen	APP_CLEAR_PASSWORD_CACHE
Palettensatz auswählen	APP_SELECT_PALETTE_SET
Bedienelemente- und Funktionenpalette bearbeiten...	APP_EDIT_PALETTES
Ausführen	APP_OPERATE

Tabelle 6-1. Tags von Anwendungselementen (Fortsetzung)

Anwendungselement	Tag
Stopp	APP_STOP
Drucken nach Beendigung	APP_PRINT_AT_COMPLETION
Protokoll nach Beendigung	APP_LOG_AT_COMPLETION
Datenprotokollierung	APP_DATA_LOGGING
Protokoll...	APP_LOG
Wiedergewinnen...	APP_RETRIEVE
Daten freigeben...	APP_PURGE_DATA
Log-Dateibindung ändern...	APP_CHANGE_LOG_BINDING
Bindung zu Log-Datei löschen	APP_CLEAR_LOG_BINDING
Bei Aufruf anhalten	APP_SUSPEND_WHEN_CALLED
Zurücksetzen auf Standard	APP_REINIT_ALL_TO_DEFAULT
Projekt	APP_PROJECT
VIs Benutzerprojekte *	APP_USER_PROJECT_VIS
VI-Hierarchie anzeigen	APP_SHOW_VI_HIER
Aufrufer des VIs	APP_THIS_VIS_CALLERS
SubVIs des VIs	APP_THIS_VIS_SUBVIS
Ungeöffnete SubVIs	APP_UNOPENED_SUBVIS
Ungeöffnete Type Defs	APP_UNOPENED_TYPEDEFS
Suchen...	APP_FIND
Suchergebnisse	APP_SHOW_SRCH_RSLTS
Weitersuchen	APP_FIND_NEXT
Vorherigen suchen	APP_FIND_PREV
Profilfenster anzeigen	APP_SHOW_PROFILE_WINDOW
VIs vergleichen...	APP_COMPARE_VIS
Unterschiede anzeigen	APP_SHOW_DIFFERENCES
Strings exportieren...	APP_EXPORT_STRINGS

Tabelle 6-1. Tags von Anwendungselementen (Fortsetzung)

Anwendungselement	Tag
Strings importieren...	APP_IMPORT_STRINGS
Fenster	APP_WINDOWS
Panels anzeigen	APP_SHOW_PANEL_DIAGRAM
VI-Info anzeigen...	APP_SHOW_VI_INFO
Historie anzeigen	APP_SHOW_HISTORY
Funktionspalette anzeigen	APP_SHOW_CTRLN_FCTNS
Werkzeugpalette anzeigen	APP_SHOW_TOOLS
Zwischenablage anzeigen	APP_SHOW_CLIPBOARD
Fehlerliste anzeigen	APP_SHOW_ERRORS
Nebeneinander anordnen	APP_TILE_LEFT_RIGHT
Übereinander anordnen	APP_TILE_UP_DOWN
Maximiert	APP_FULL_SIZE
Offene Fenster *	APP_OPEN_WINDOWS
Hilfe	APP_HELP
Hilfe anzeigen	APP_SHOW_HELP
Hilfe fixieren	APP_LOCK_HELP
Einfache Hilfe	APP_SIMPLE_HELP
Online Referenz...	APP_ONLINE_REF
Online-Hilfe für VI	APP_ONLINE_HELP_FOR_VI
Hilfdateien *	APP_HELP_FILES
Über LabVIEW...	APP_ABOUT

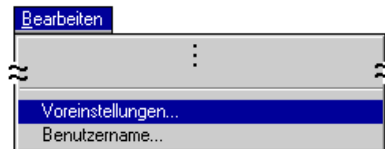
Anpassen Ihrer Umgebung

Dieses Kapitel erläutert, wie die Umgebung durch Festlegen von Editor-Voreinstellungen für Funktionen wie z.B. Druck, Anzeige und den Befehl **Rückgängig** und durch Ändern des Inhalts der Paletten **Elemente** und **Funktionen** angepaßt werden kann.

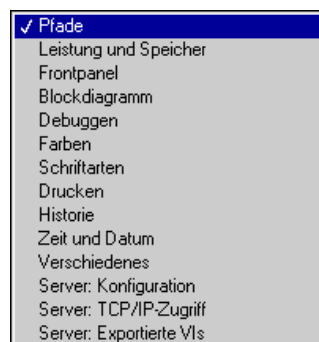
Festlegen von Voreinstellungen

Zum Anpassen der Anwendung und Konfigurieren des Editors benutzen Sie das Dialogfeld “Voreinstellungen”.

Zugriff auf das Dialogfeld “Voreinstellungen” erhalten Sie durch Auswählen von **Bearbeiten»Voreinstellungen...** (siehe Abbildung unten).

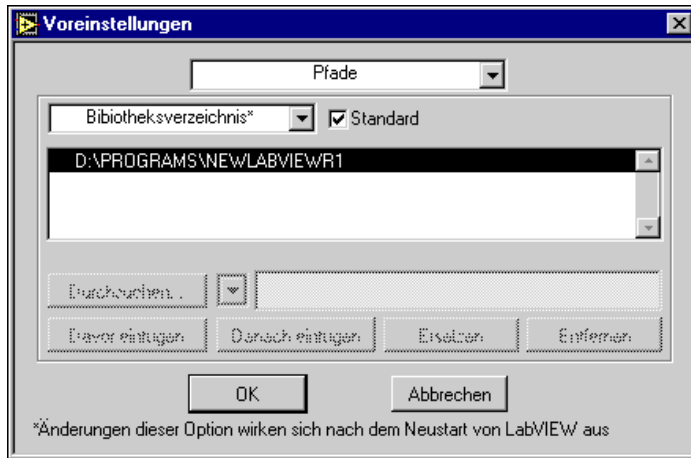


Benutzen Sie das Pulldown-Menü am oberen Rand des Dialogfeldes, um eine Auswahl unter den verschiedenen Kategorien von Voreinstellungen zu treffen (siehe nachstehende Abbildung).



Pfadvoreinstellungen

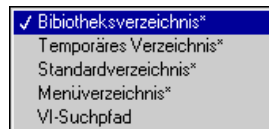
Sie können die Verzeichnisse angeben, die beim Suchen von VIs durchsucht werden, sowie die Pfade für temporäre Dateien und das Bibliothekverzeichnis. Wählen Sie, sofern dies noch nicht der Fall ist, **Pfade** aus dem Pull-down-Menü im Dialogfeld “Voreinstellungen”, um das in der folgenden Abbildung gezeigte Dialogfeld anzuzeigen.



Hinweis

Das Format von Pfadnamen — die Verwendung von Doppelpunkten, Schrägstrichen und umgekehrten Schrägstrichen — unterscheidet sich je nach Plattform geringfügig.

Im Dialogfeld “Pfade Voreinstellungen” gibt es ein weiteres Pull-down- Menü, das in der folgenden Abbildung gezeigt wird und in dem Sie die Pfadkategorie auswählen, die Sie anzeigen oder bearbeiten wollen.

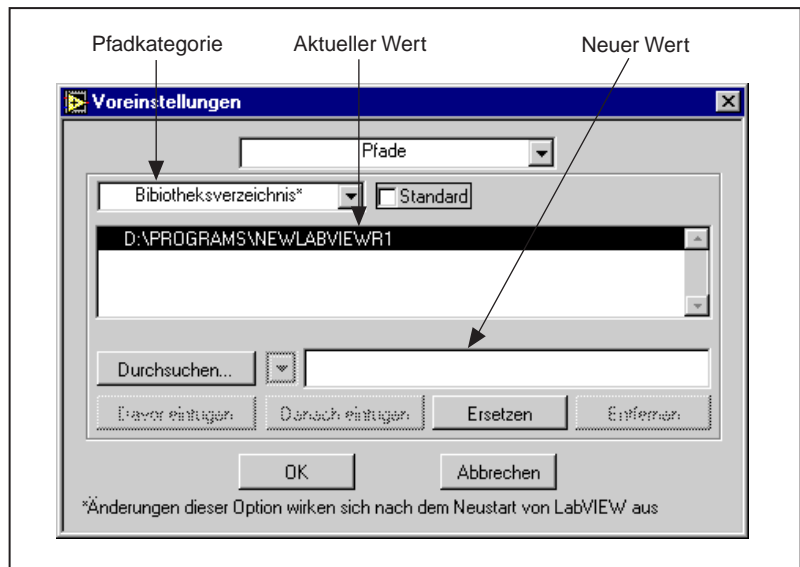


In dem anschließend angezeigten Dialogfeld sind viele Menüpunkte als nicht bedienbar markiert und nicht aktiv, da G auf den Standardpfad voreingestellt ist. Wenn Sie eine dieser Voreinstellungen ändern wollen, heben Sie die Auswahl des Kontrollkästchens **Standard** auf (siehe nachstehende Abbildung).



Bibliothekverzeichnisse, temporäre Verzeichnisse, Standardverzeichnisse und Menüverzeichnisse

Die Einstellungen für **Bibliothekverzeichnis**, **Temporäres Verzeichnis**, **Standardverzeichnis** und **Menüverzeichnis** sind einzelne Verzeichnisse (Ordner). Wenn Sie einen dieser Pfade ändern, haben Sie die Wahl, einen neuen Pfad einzugeben und den bestehenden Pfad damit zu ersetzen oder das Dateidialogfeld zu durchsuchen, um einen Pfad auszuwählen (siehe folgende Abbildung).



Die Option **Bibliothekverzeichnis** zeigt den absoluten Pfadnamen zu dem Verzeichnis an, das die Bibliothek `vi.lib` enthält, sowie alle Bibliothekverzeichnisse, die Sie angeben. Die Standardeinstellung ist das Verzeichnis, das die G-Entwicklungsumgebung enthält.

Die Option **Temporäres Verzeichnis** zeigt den absoluten Pfadnamen zu dem Verzeichnis für temporäre Dateien an, das sich je nach Plattform in der folgenden Weise unterscheidet:

- **(Windows)** Die Standardeinstellung ist das Verzeichnis, das die G-Entwicklungsumgebung enthält.
- **(Macintosh)** Unter System 7 ist die Standardeinstellung ein unsichtbarer Ordner namens `Temporary Items` direkt auf dem Festplattenlaufwerk, den Apple für die Speicherung temporärer Dateien empfiehlt. Wenn die Entwicklungsumgebung abstürzt, werden die temporären Dateien beim Neustart in den Papierkorb verschoben.
- **(UNIX)** Die Standardeinstellung ist das Verzeichnis `/tmp`.

Die Option **Standardverzeichnis** zeigt den absoluten Pfadnamen zu dem Standardverzeichnis an, das im Dialogfeld zum Öffnen von Dateien angezeigt wird. Die Funktion **Standardverzeichnis** in der Palette **Funktionen»Datei I/O»Dateikonstanten** gibt ebenfalls diesen Wert zurück. Die Standardeinstellung ist das aktuelle Arbeitsverzeichnis. Änderungen am Standardverzeichnis treten sofort in Kraft.

Die Option **Menüverzeichnis** zeigt den Pfad zu dem Verzeichnis an, das die Palettenmenüdateien enthält, die den Aufbau der Palette **Elemente** und **Funktionen** beschreiben. Standardmäßig handelt es sich hierbei um das **Menüverzeichnis** im **Bibliothekverzeichnis**. Im allgemeinen ist es nicht notwendig, diesen Wert zu ändern, obgleich dies unter Umständen dann sinnvoll ist, wenn eine Netzwerkinstallation vorliegt und einzelne Benutzer ihre eigenen angepaßten Menüsätze benötigen.



Hinweis

*Alle Änderungen in den Optionen **Bibliothekverzeichnis**, **Temporäres Verzeichnis**, **Menüverzeichnis** und **Standardverzeichnis** treten in Kraft, sobald Sie die G-Entwicklungsumgebung neu starten.*

VI-Suchpfad

Die Option **VI-Suchpfad** kommt nur zur Anwendung, wenn die G-Entwicklungsumgebung ein SubVI, Bedienelement oder ein externes Unterprogramm sucht, das nicht an der erwarteten Stelle vorgefunden wurde. Sie ermöglicht Ihnen, den Verzeichnispfad aufzulisten, den G durchsuchen soll. Wenn Sie den Suchpfad ändern, erhalten Sie mehr Optionen als bei den anderen Pfaden — Sie können neue Objekte an bestimmten Stellen hinzufügen, Pfade entfernen und aus einer Liste *spezieller* Pfade auswählen.



Die Liste enthält Pfade, die die G-Entwicklungsumgebung durchsucht, in chronologischer Reihenfolge. Um ein neues Verzeichnis hinzuzufügen, treffen Sie zunächst eine Entscheidung, wann die G-Entwicklungsumgebung das betreffende Verzeichnis relativ zu anderen Verzeichnissen durchsucht. Wählen Sie ein benachbartes Verzeichnis aus der Liste. Verwenden Sie dann entweder die Taste **Durchsuchen...**, die ein Dialogfeld zum Öffnen von Dateien anzeigt, oder das besondere Pulldown-Menü, das eine Liste spezieller Pfade enthält, um das von der G-Entwicklungsumgebung zu durchsuchende Verzeichnis zu wählen. Ändern Sie diesen Pfad, oder geben Sie einen Pfad in die String-Steuerung neben diesen Optionen ein. Mit **Davor einfügen**, **Danach einfügen** oder **Ersetzen** schließlich wird der gewählte Pfad der Liste hinzugefügt.

Wenn Sie einen Pfad auswählen, durchsucht G normalerweise das jeweilige Verzeichnis, nicht aber dessen Unterverzeichnisse. Durch Anfügen eines Sternchens (*) als Pfadobjekt wird die Suche hierarchisch.

Nachstehend einige Beispiele auf verschiedenen Plattformen:

- **(Windows)** Um das Verzeichnis `C:\VIS\` rekursiv zu durchsuchen, geben Sie `C:\VIS*` ein.
- **(Macintosh)** Um den Ordner `VI`s auf der Festplatte rekursiv zu durchsuchen, geben Sie `HD:VI:s:*` ein.
- **(UNIX)** Um das Verzeichnis `/usr/home/gregg/vis` rekursiv zu durchsuchen, geben Sie `/usr/home/gregg/vis/*` ein.

Benutzen Sie das in der nachstehenden Abbildung gezeigte, spezielle Pulldown-Menü, das sich rechts von der Taste **Durchsuchen...** befindet, um unter mehreren Sonderverzeichnissen zu wählen.



Es handelt sich hierbei um die folgenden Verzeichnisse:

- `<vilib>` als Pfadeintrag ist ein symbolischer Pfad, der auf das Verzeichnis `vi.lib` innerhalb des Bibliothekverzeichnisses verweist.
- `<topvi>` bezieht sich auf das Verzeichnis, das das Top-Level VI enthält, das geöffnet wird.
- `<foundvi>` bezieht sich auf eine Liste von Verzeichnissen, die LabVIEW jedesmal erzeugt, wenn ein VI geladen wird. Während einer Suche wird jedes Verzeichnis, in dem LabVIEW ein SubVI findet, oder ein manuell ausgewähltes Verzeichnis zu dieser Liste hinzugefügt. Benutzen Sie den symbolischen Pfad in der Weise, daß Sie ein Verzeichnis von VIs nur einmal beim Laden suchen müssen, wenn Sie dieses verschoben oder umbenannt haben und dann ein aufrufendes VI öffnen.

Mit der Taste **Entfernen** entfernen Sie einen Pfad, wobei der entfernte Pfad in das String-Feld übertragen wird, falls Sie sich entscheiden, ihn in der Liste an anderer Stelle einzufügen.



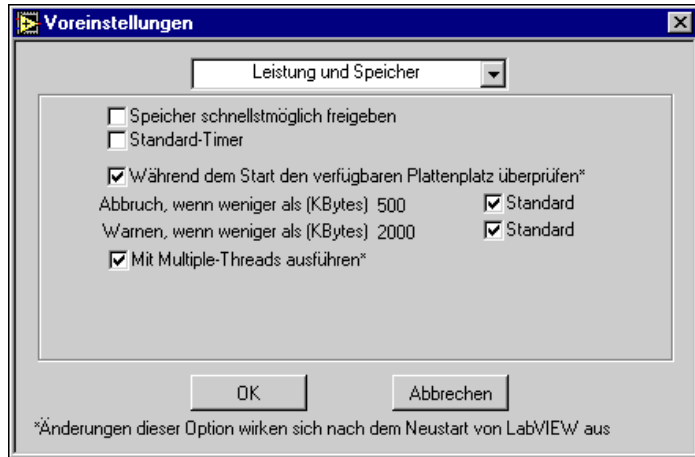
Hinweis

Änderungen in der Option VI-Suchpfad treten beim nächsten Starten der Anwendung in Kraft.

Voreinstellungen für Leistung und Speicher medium

Manche Bearbeitungsoperationen erfordern den Gebrauch von temporären Dateien auf dem Speichermedium. Wenn Sie zum Beispiel ein VI speichern, wird es zunächst in einer temporären Datei gespeichert. Ist die Speicherung erfolgreich, wird die ursprüngliche Datei durch die neue Datei ersetzt. Um Probleme zu vermeiden, wird der auf dem Speichermedium verfügbare Speicherplatz im temporären Verzeichnis zum Startzeitpunkt überprüft. Stehen weniger als 500 KB Speicherplatz zur Verfügung, erhalten Sie eine dementsprechende Nachricht, und der Start wird abgebrochen. Wenn weniger als 2 MB zur Verfügung stehen, erhalten Sie ebenfalls eine Nachricht, aber der Start wird fortgesetzt. Sie können diese Warnungen ausschalten oder die Stufen, auf denen Sie gewarnt werden, mit dem Dialogfeld "Leistung und Speichermedium" ändern, auf das Sie mit dem Pulldown-Menü "Voreinstellungen" Zugriff erhalten.

Das Dialogfeld "Leistung und Speichermedium Voreinstellungen" ist in der folgenden Abbildung zu sehen. Beachten Sie, daß unter Windows 3.1 nur ein Menüpunkt zur Verfügung steht und einige andere Optionen nur auf dem Macintosh zur Verfügung stehen.



In diesem Dialogfeld stehen die folgenden Optionen zur Verfügung:

Speicher schnellstmöglich freigeben — Hebt die Speicherzuweisung jedes VI nach Ausführungsabschluß auf. Hiermit läßt sich kann bei manchen Anwendungen die Speicherauslastung verbessern, da die Speicherzuweisung bei SubVIs unmittelbar nach der Ausführung erfolgt. Dies setzt die Leistung herab, da G öfter Speicher zuweisen und die Speicherzuweisung wieder aufheben muß und dies in manchen Fällen zu einer übermäßigen Speicherfragmentierung führen könnte.

(Windows 3.1) Standard-Timer — Die Timing-Funktionen in G benutzen den implementierten Windows-Timer. Dieser Timer hat standardmäßig eine Auflösung von 55 ms (er inkrementiert 18-mal pro Sekunde). Dies bedeutet, daß die VIs “Tick Count”, “Warten” und “Wartet bis zum nächsten Vielfachen von ms” in G nur eine Auflösung von 55 ms haben. Beachten Sie jedoch, daß Sie für die Datenerfassungs-VIs genauere Timingraten angeben können, da diese zur Steuerung der Erfassungsrate die implementierten Timer der Datenerfassungs-Karten verwenden.

Erhöhen Sie die Auflösung der VIs “Tick Count”, “Warten” und “Wartet bis zum nächsten Vielfachen von ms” auf 1 ms, indem Sie die Auflösung des implementierten Windows-Timers auf 1 ms erhöhen. Wenn Sie die Timer-Auflösung erhöhen, werden die Timing-Schleifen der Software genauer.

Unter bestimmten Umständen empfiehlt es sich nicht, die Timer-Auflösung von 55 ms zu erhöhen. Die Änderung der Auflösung erhöht die Anzahl von Timer Tick-Interrupts von 18 Interrupts pro Sekunde auf 1000 Interrupts pro Sekunde. Wenn Sie andere Interrupt-intensive Operationen

durchführen, überschreiten Sie unter Umständen die Kapazität Ihres PCs zur Bewältigung der Interrupt-Last, was eventuell zum Sperren oder Abstürzen des PCs führen kann.

DAQ-Operationen, die zur Datenübertragung zwischen DAQ-Karte und PC-Speicher statt DMA programmierte I/O verwenden, sind Interrupt-intensiv. Erhöhen Sie die Tick-Auflösung des Timers nicht auf 1 ms, wenn Sie beabsichtigen, bei hohen Übertragungsraten eine der folgenden Operationen durchzuführen:

- Gepufferte analoge Eingaben unter Verwendung der PC-LPM-16
- Gepufferte analoge Ausgaben unter Verwendung der Karten AT-MIO-16, MC-MIO-16 oder der Lab-Serie
- Gepufferte digitale I/O unter Verwendung der Karten DIO-24, DIO-96 oder der Lab-Serie
- Gepufferte analoge Ein- oder Ausgänge, wobei DMA in `WDAQCONF` deaktiviert ist, und unter Verwendung einer beliebigen DAQ-Karte

Übertragungsraten von ca. 30.000 oder mehr Abtastwerten/Sek auf einem 386-er Computer mit 25 MHz Taktrate verursachen bei der Timer-Auflösung 1 ms Probleme bei der Interrupt-Last. Übertragungsraten von ca. 75.000-80.000 oder mehr Abtastwerten/Sek auf einem 486-er Computer mit 33 MHz Taktrate oder 100.000 oder mehr Abtastwerte/Sek bei einem 486-er Computer mit 50 MHz verursachen bei der Timer-Auflösung 1 ms Probleme bei der Interrupt-Last.

Die Timing-Funktionen in G für Windows 95/NT haben eine Timer-Auflösung von 1 ms, was der Maximalauflösung unter Windows 95/NT entspricht. Langsamere Computer haben unter Umständen eine geringere Auflösung.

(Macintosh) Verdichten des Speichers während der Ausführung — Weist G an, den Speicher ca. alle 30 Sekunden zu komprimieren. Die Komprimierung reduziert die Fragmentierung, indem zugewiesener Speicher an eine einzelne Stelle verschoben wird. Ein Nachteil ist, daß die Komprimierung Zeit in Anspruch nimmt und einen kurzen Leistungsabfall auslöst, während G den Speicher komprimiert.

(Macintosh) Kooperationsebene — Mit dieser Option können Sie steuern, wie kooperativ G mit anderen Hintergrundanwendungen ist. Wenn Sie diese Option auf **Niedrig** setzen, teilt G nicht häufig Zeit mit anderen Anwendungen. Bei einer niedrigen Kooperationsebene erhöht sich die Leistung von G, die Leistung anderer, parallel dazu ausgeführter Anwendungen wird jedoch herabgesetzt. Wenn Sie die Kooperationsebene

auf **Hoch** setzen, beeinträchtigt dies die Leistung von G, während anderen Anwendungen mehr Zeit zur Ausführung gegeben wird.

Beim Start verfügbare Datenträgerkapazität prüfen — Prüft beim Start von G den im temporären Verzeichnis verfügbaren Speicherplatz. Sie können in zwei Fällen wählen, ob G den Standardwert verwenden soll: Abbruch bei weniger als 500 KB oder Warnmeldung bei weniger als 2.000 KB.

Mit multiplen Threads ausführen — Diese Option ist nur bei Betriebssystemen verfügbar, die Multithreading unterstützen. Wenn Sie diese Option ausschalten, verhält sich das ausführende System wie eines mit nur einem Thread, so daß Interaktion mit der Benutzerschnittstelle und Ausführung von VIs mit dem gleichen Thread erfolgen. Sie können diese Option etwa aus Gründen der Kompatibilität benutzen, wenn VIs in einem System mit Multi-Thread-Ausführung unzuverlässig sind.

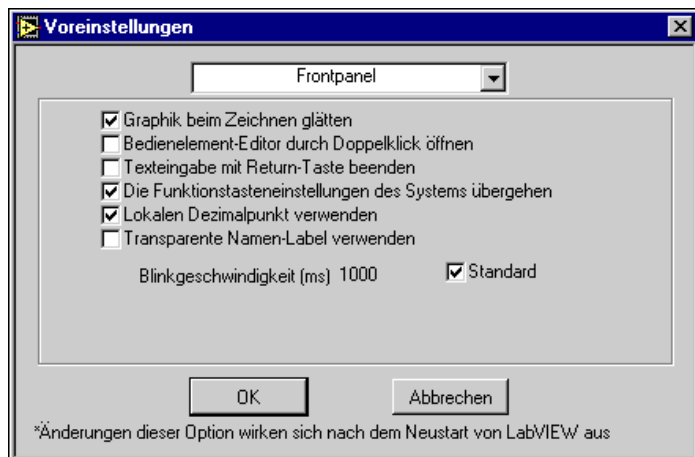


Hinweis

Änderungen in den Optionen Beim Start verfügbare Datenträgerkapazität prüfen und Standard-Timer und Mit multiplen Threads ausführen treten in Kraft, sobald Sie G neu starten. Änderungen in den Optionen Speicher während der Ausführung verdichten, Speicher baldmöglichst freigeben und Kooperationsebene treten sofort in Kraft.

Voreinstellungen für Frontpanels

Die folgende Abbildung zeigt das Dialogfeld "Frontpanel Voreinstellungen".



In diesem Dialogfeld stehen die folgenden Optionen zur Verfügung:

Bedienelement-Editor durch Doppelklick öffnen — Ermöglicht den direkten Zugriff auf das Fenster des Bedienelement-Editors, in dem das Aussehen eines Bedien- oder Anzeigeelements eines Frontpanels angepaßt werden kann.

Texteingabe mit Return-Schaltfläche beenden — Paßt die Funktionsweise der <Return>-Taste auf der alphanumerischen Tastatur an die der <Eingabe>-Taste auf dem numerischen Tastenfeld an, wodurch die Texteingabe beendet wird. Wenn Sie diese Option wählen, können Sie durch Drücken von <Strg-Eingabe> (**Windows**), <Option-Return> (**Macintosh**) <Meta-Return> (**Sun**) oder <Alt-Return> (**HP-UX**) neue Zeilen einbetten.

(Windows, Macintosh) Die Funktions-Tasteneinstellungen des Systems übergehen — Standardmäßig benutzt Ihr Betriebssystem möglicherweise einige Funktionstasten zu Systemzwecken. Auf dem PC entspricht das Drücken von F10 dem Drücken der <Break>-Taste. Beim Macintosh werden die Tasten F1 bis F4 gewöhnlich für **Ausschneiden, Kopieren, Einfügen** und **Löschen** verwendet. Wenn Sie diese Übersteuerung einschalten, werden die Funktionstasten nicht zu Systemzwecken verwendet, sondern in G als standardmäßige Funktionstasten zur Verfügung gestellt.

Lokalen Dezimalpunkt verwenden — Benutzt statt des Punktes das Dezimaltrennzeichen des Systems. In Deutschland und in vielen anderen Ländern wird beispielsweise das Komma als Dezimaltrennzeichen verwendet. Schalten Sie diese Option ein, wenn G die Systemkonfiguration verwenden soll. Schalten Sie die Option aus, wenn G in allen Fällen den Punkt als Dezimaltrennzeichen verwenden soll.

Graphik beim Zeichnen glätten — Wenn G ein Bedienelement mit ausgeschalteter Grafikglättung (smooth update off) aktualisiert, wird der Inhalt des Bedienelements gelöscht und mit dem neuen Wert neu gezeichnet. Hierbei kann ein bemerkbares Flackern auftreten, da der alte Wert gelöscht und ersetzt wird. Beim Glätten der Grafik zeichnet G Daten in einem Offscreen-Puffer und kopiert dieses Bild dann auf den Bildschirm, anstatt einen Teil des Bildschirms zu löschen. Auf diese Weise wird das durch Löschen und Zeichnen ausgelöste Flackern vermieden. Dies kann jedoch die Leistung herabsetzen und mehr Anwendungsspeicher erfordern, da ein Offscreen-Zeichnungspuffer gepflegt werden muß.

Transparente Namen-Label verwenden — Weist die Software an, durchsichtige Labels zu verwenden.

(Sun) Unterstützt den Zahlenblock von Sun-Tastaturen — Aktiviert die Unterstützung für die Sun-Tastatur, einschließlich der Tastenfelder, Pfeiltasten und der Hilfetaste. Aktivieren Sie diese Option nicht, wenn Sie keine Sun-Tastatur verwenden (wenn Sie zum Beispiel ein X-Terminal oder einen PC verwenden, auf dem X-Software ausgeführt wird).

Blinkgeschwindigkeit — Setzt die Blinkgeschwindigkeit für Frontpanel-Objekte auf die standardmäßigen 1.000 ms oder eine andere Geschwindigkeit, die Sie eingeben. Blinken ist ein grundlegendes Attribut, das mit Attributknoten aktiviert wird. Weitere Informationen hierzu finden Sie in Kapitel 22, [Attributknoten](#).

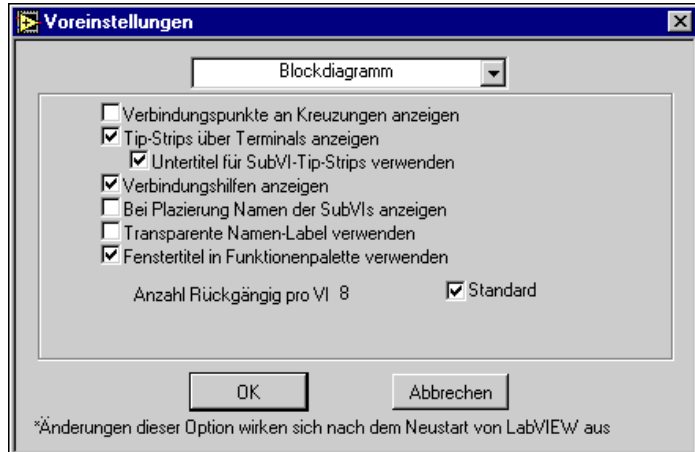


Hinweis

Änderungen an Optionen im Dialogfeld “Voreinstellungen Frontpanel” treten sofort in Kraft.

Voreinstellungen für Blockdiagramme

Die folgende Abbildung zeigt das Dialogfeld “Blockdiagramm Voreinstellungen”.



In diesem Dialogfeld stehen die folgenden Optionen zur Verfügung:

Verbindungspunkte an Kreuzungen anzeigen — Plaziert Punkte an die Stelle, an der Verbindungen zu mehreren Zielen abzweigen. Dies erleichtert die Unterscheidung solcher Kreuzungen und Verbindungen, die sich lediglich überschneiden.

Tip-Strips über Terminals anzeigen — Plaziert Parameternamen unter Terminals, wenn Sie in Funktionen und SubVIs mit dem Cursor darauf pausieren.

Untertitel für SubVI-Tip-Strips verwenden — Benutzt statt Bedienelementnamen Titel von Bedienelementen, wenn Hilfe zu VIs oder Hinweisstrips für VI-Terminals angezeigt werden.

Verbindungshilfen anzeigen — Zeigt Verbindungsstümpfe an, die den Datentyp und die Datenrichtung für jeden Parameter angeben, wenn der Cursor auf einem Knoten eines Blockdiagramms pausiert.

Bei Platzierung Namen der SubVIs anzeigen — Weist die Software an, einem SubVI den Namen des ursprünglichen VIs zuzuweisen, wenn es auf dem Blockdiagramm abgelegt wird, und das Label anzuzeigen.

Transparente Namen-Label verwenden — Weist die Software an, durchsichtige Labels zu benutzen.

Fenstertitel in Funktionspalette verwenden — Verwendet in den Funktionspaletten statt der Namen des VIs die Fenstertitel des VIs bei der Anzeige des Palettennamens eines VI.

Anzahl Rückgängig pro VI — Legt die Anzahl der Aktionen fest, die rückgängig gemacht und wiederhergestellt werden können. Ist die Anzahl der Aktionen Null, ist die Funktion "Rückgängigmachen" deaktiviert. Jedes VI zeichnet seine eigenen Instanzen zum Rückgängigmachen auf, daher bezieht sich die Anzahl, die Sie hier angeben, auf jedes einzelne VI. Wenn Sie diesen Wert geringer ansetzen, ist auch der Aufwand an Speicherressourcen geringer. Weitere Informationen hierzu finden Sie im Abschnitt *Rückgängig machen* in diesem Kapitel.

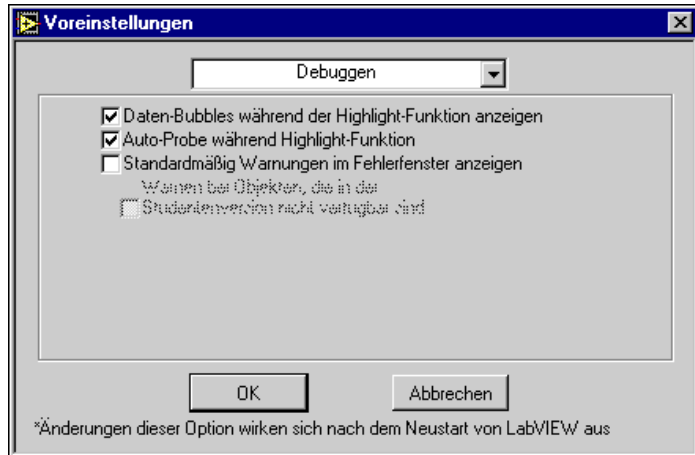


Hinweis

Änderungen an Optionen im Dialogfeld "Blockdiagramm Voreinstellungen" treten sofort in Kraft.

Debuggen-Voreinstellungen

Die folgende Abbildung zeigt das Dialogfeld “Debuggen-Voreinstellungen”.



In diesem Dialogfeld stehen die folgenden Optionen zur Verfügung:

Daten-Bubbles während der Highlight-Funktion anzeigen — Setzt den Ausführungsfluß durch Zeichnen von Bubbles entlang der Verbindungen in Bewegung. Schalten Sie diese Option für schnelleres Debugging mit weniger Animation aus.

Auto-Probe während Highlight-Funktion — Untersucht skalare Werte automatisch und zeichnet ihre Werte in das Diagramm. Schalten Sie diese Funktion aus, wenn die Anzeige dadurch zu unübersichtlich wird.

Standardmäßig Warnungen im Fehlerfenster anzeigen — Zeigt zusätzlich zu Fehlern im Dialogfeld “Fehlerliste” Warnungen an. Eine Warnung bedeutet nicht, daß das VI falsch ist; sie weist lediglich auf ein mögliches Problem im Blockdiagramm hin.

Warnen bei Objekten, die in der Studentenversion nicht verfügbar sind — Wenn Sie den vorherigen Menüpunkt über Warnungen wählen, enthält dieser Menüpunkt Warnungen über Objekte, die in der Studenten-Ausgabe von LabVIEW nicht zur Verfügung stehen.

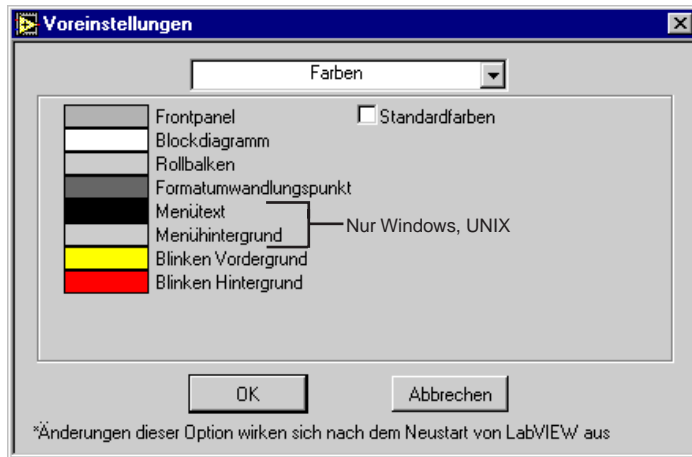


Hinweis

Änderungen an Optionen im Dialogfeld “Debug-Voreinstellungen” treten sofort in Kraft.

Voreinstellungen für Farben

Die folgende Abbildung zeigt das Dialogfeld “Voreinstellungen Farbe”.



Mit diesem Dialogfeld können Sie die Farben ändern, die LabVIEW für verschiedene Objekte verwendet. Wenn Sie das Kontrollkästchen **Standardfarben** nicht wählen, können Sie auf ein beliebiges Rechteck klicken, um die Farbe zu ändern. Die Optionen im einzelnen lauten folgendermaßen:

Frontpanel — Wählt eine Farbe für das Frontpanel neuer VIs. Das Ändern der Farbe wirkt sich nicht auf alte VIs aus.

Blockdiagramm — Wählt eine Farbe für das Blockdiagramm neuer VIs. Das Ändern der Farbe wirkt sich nicht auf alte VIs aus.

Rollbalken — Wählt eine Farbe für Rollbalken. Hiervon sind nur die gegenwärtig geöffneten VIs betroffen.

Formatumwandlungspunkt — Wählt eine Farbe für die Punkte, die die Formatumwandlung numerischer Daten anzeigen. Hiervon sind nur die gegenwärtig geöffneten VIs betroffen.

(Windows, UNIX) Menütext — Wählt eine Farbe für Text, der in Menüs verwendet wird.

(Windows, UNIX) Menühintergrund — Ermöglicht Ihnen, eine Farbe für den in Menüs verwendeten Hintergrund zu wählen.

Blinken Vordergrund — Ermöglicht Ihnen, die Vordergrundfarbe für ein blinkendes Objekt zu wählen. Hiervon ist nur ein blinkendes Objekt betroffen. Blinken ist ein grundlegendes Attribut, das mit Attributknoten eingeschaltet wird. Weitere Informationen hierzu finden Sie in Kapitel 22, [Attributknoten](#).

Blinken Hintergrund — Ermöglicht Ihnen, die Hintergrundfarbe für ein blinkendes Objekt zu wählen. Hiervon ist nur ein blinkendes Objekt betroffen.

Standardfarben — Benutzt die standardmäßige Farbauswahl für alle in diesem Dialogfeld aufgelisteten Optionen. Sie müssen dieses Kontrollkästchen deaktivieren, wenn Sie einzelne Farben ändern wollen.

(UNIX) Weitere Farben — Schaltet zwischen einer Farbpalette mit 216 Farben und einer Farbpalette mit 125 Farben um. Eine umfangreichere Palette löst möglicherweise ein sichtbares Blitzen aus, wenn Sie zwischen Fenstern wechseln und Ihr System nur 256 Farben unterstützt.

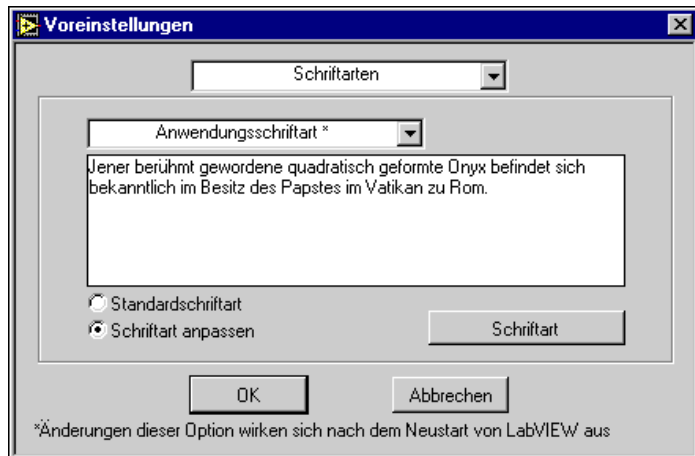


Hinweis

Änderungen an Optionen im Dialogfeld “Voreinstellungen Farbe” treten sofort in Kraft.

Voreinstellungen für Schriftarten

Das Dialogfeld “Voreinstellungen Schriftart” ermöglicht Ihnen, drei Kategorien von vordefinierten Schriftarten zu ändern: die Anwendungs-, System- und Dialogfeldschriftart. Sie wählen die zu ändernde Kategorie im Popup-Menü über dem Textfeld aus. Die folgende Abbildung zeigt das Dialogfeld, in dem die Option **Anwendungsschriftart** gewählt ist.



G benutzt für bestimmte Teile der Benutzerschnittstelle drei Kategorien von Schriftarten. Die Schriftarten sind je nach Plattform wie folgt vordefiniert:

Standardschriftart — Benutzt, wie zuvor beschrieben, die von G für die jeweilige Plattform definierte Standardschriftart.

Schriftart anpassen — Dieses Kontrollkästchen wird automatisch aktiviert, wenn Sie mit Hilfe der Option **Font Style...** einzelne Schriftartmerkmale ändern.

Schriftart... — Ruft ein Dialogfeld auf, in dem die Schriftartmerkmale geändert werden können.



Hinweis

Änderungen an Optionen im Dialogfeld “Voreinstellungen Schriftart” treten nicht sofort in Kraft. Die Software muß hierzu neu gestartet werden.

Weitere Einzelheiten über die Vorabdefinition der Schriftarten oder Informationen über die Portierbarkeit von Schriftarten finden Sie im Abschnitt [Auflösungs- und Schriftartunterschiede](#) in Kapitel 29, [Portierbarkeit und Lokalisierungsthemen](#).

Druckervoreinstellungen

Die folgende Abbildung zeigt das Dialogfeld “Drucker Voreinstellungen”. Beachten Sie, daß manche Optionen nur auf bestimmten Plattformen zur Verfügung stehen.



(Windows, Macintosh) Standarddruck — Formatiert die VI-Druckdaten (Frontpanel, Diagram, Icon usw.) und druckt sie mit den standardmäßigen

Zeichenbefehlen. Benutzen Sie diese Option, wenn Sie die Druckausgabe in eine Datei umleiten wollen (vorausgesetzt, daß Ihr Druckertreiber die Druckumleitung in eine Datei unterstützt), wenn Ihr Drucker keine PostScript-Unterstützung bietet oder wenn Sie wollen, daß der Druckertreiber und nicht die Software der Sprache G die PostScript-Umsetzung übernimmt.

(Nur Windows) Drucker-Dithering — Benutzt beim Erstellen der Zeichenbefehle “Drucker-Dithering”. Drucker-Dithering, eine Funktion, die für den Schwarzweiß-Druck gilt, ist eine Methode, bei der zu einem Bild Graustufen hinzugefügt werden, anstatt nur schwarze und weiße Bereiche zu verwenden. Dieser Menüpunkt ist vom Druckertyp unabhängig.

PostScript drucken — Setzt die VI-Druckdaten ins PostScript-Format (.ps) um und sendet sie als PostScript-Text an den Drucker. Wenn Sie einen PostScript-Drucker und einen PostScript-Druckertreiber haben, handelt es sich beim dem Ausdruck um ein Grafikbild. Wählen Sie diesen Menüpunkt nicht, wenn Ihr Drucker oder Druckertreiber keinen PostScript-Druck unterstützt.

PostScript Level 2 — Teilt der Software mit, daß der angeschlossene Drucker **PostScript level 2** unterstützt. Ist dieses Feld aktiviert, sendet G **PostScript level 2**-Code der Ebene 2 an den Drucker. Wählen Sie diese Option nur, wenn Ihr Drucker die **PostScript level 2** unterstützt.

(Windows, UNIX) Bitmap Drucken — Erstellt eine Bitmap-Datei, zeichnet alle Daten der betreffenden Seite in die Bitmap-Datei, und sendet diese an den Drucker. Diese Methode dauert länger als die anderen beiden Methoden, kann aber eine bessere Darstellung des Textes und der Schriftarten erzielen als der standardmäßige Druck, selbst wenn der Ausdruck nicht dieselbe hohe Auflösung wie der PostScript-Druck hat. Das Bitmap-Bild ist bei Farbdruckern in Farbe und andernfalls in Schwarz/Weiß.

(Nur Windows) Ausdruck in Farbe/Graustufen — Dieser Menüpunkt generiert eine Farb-/Graustufenausgabe mit 8 Bits-pro Pixel (BPP) an einem Drucker und deaktiviert damit die vom Druckertreiber stammenden Tiefeneinstellungen. Die meisten Drucker melden die richtigen Farbdruckfähigkeiten, manche Laserdruckertreiber melden jedoch eine Bittiefe von 1 an, während der Drucker eigentlich 8 BPP verarbeiten kann.

Seitenränder — Sie können absolute Ränder für Ausdrücke in Millimetern und Zoll festlegen. Alle vier Ränder (oben, unten, links und rechts) können separat angegeben werden. Ränder sind durch physische

Parameter des Druckers begrenzt. Wenn Sie Ränder kleiner einstellen, als sie vom Drucker akzeptiert werden, sind die eigentlichen Ränder die vom Drucker akzeptierten Mindestränder. Die Rändereinstellung im Referenzdialogfeld betrifft neu erstellte oder umgewandelte VIs. Wenn Sie die Ränder eines bestimmten VIs festlegen wollen, verwenden Sie die Option **VI-Einstellungen**.



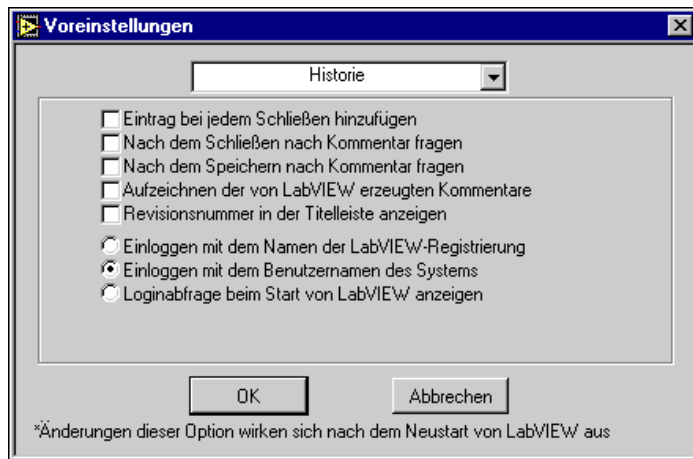
Hinweis

Änderungen an Optionen im Dialogfeld “Drucker Voreinstellungen” treten sofort in Kraft.

Historie-Voreinstellungen

Wie in Kapitel 27, *Verwalten Ihrer Anwendungen*, erläutert besitzt jedes VI ein Historie-Fenster, in dem die Entwicklungshistorie des VIs angezeigt wird. Mit dem Dialogfeld “Historie Voreinstellungen” können Sie die Standardeinstellungen für das Historie-Fenster neuer VIs wählen.

Die folgende Abbildung zeigt das Dialogfeld “Historie Voreinstellungen”.



Die Optionen im Dialogfeld “Historie” werden in zwei Gruppen aufgeteilt. Die erste Gruppe enthält fünf Optionen, die anzeigen, wann und wie Einträge im Historie-Fenster für neue VIs hinzugefügt werden.

Die zweite Gruppe von Optionen im Dialogfeld “Historie Voreinstellungen” betrifft die Art und Weise Ihrer Anmeldung bei der Software der G-Sprache, die wiederum den Namen bestimmt, der in Kopfzeilen von Kommentaren eingefügt wird, die im VI-Historie-Fenster eingegeben werden.

Optionen bezüglich Einträgen im Historie-Fenster werden für *individuelle* VIs über das Dialogfeld **VI-Einstellungen»Dokumentation** angegeben. Informationen über den Zugriff auf diese Optionen finden Sie unter *Dokumentationsoptionen* in Kapitel 6, *Einrichten von VIs und SubVIs*.

In diesem Dialogfeld stehen die folgenden Optionen zur Verfügung:

Eintrag bei jedem Speichern hinzufügen — Fügt der VI-Historie jedesmal, wenn Sie das VI speichern, einen Eintrag hinzu. Wenn Sie keinen Kommentar in das Feld **Kommentar** des Historie-Fensters eingeben, wird der Historie nur eine Kopfzeile hinzugefügt. (Die Kopfzeile enthält die Revisionsnummer, falls der Menüpunkt weiter unten im Dialogfeld aktiviert ist, sowie das Datum und die Uhrzeit und den Namen des VIs.)

Nach dem Speichern nach Kommentar fragen — Ruft das Historie-Fenster für Kommentare zu Änderungen auf, die seit der letzten Speicherung vorgenommen wurden. Dies ist nützlich, wenn Sie es vorziehen, Änderungen erst hinterher mit Kommentaren zu versehen, anstatt sie während der Bearbeitung einzugeben. Wenn Sie diese Option nicht setzen, erhalten Sie beim Speichern der VI-Historie erst nach Abschluß der Speicherung wieder die Gelegenheit, die Historie zu ändern.

Nach dem Schließen nach Kommentar fragen — Mit der vorherigen Option vergleichbar, mit der Ausnahme, daß Sie die Aufforderung von G nur beim Schließen eines VIs erhalten anstatt bei jeder Speicherung. Sie erhalten eine Aufforderung, wenn das VI seit dem Laden geändert worden ist, auch dann, wenn Sie die Änderungen bereits gespeichert haben.



Hinweis

Sie werden nicht aufgefordert, einen Kommentar einzugeben, wenn Sie ein VI speichern oder schließen und lediglich Änderungen an der Historie vorgenommen wurden.

Aufzeichnen der von LabVIEW erzeugten Kommentare — Veranlaßt, daß der Editor beim Auftreten bestimmter Ereignisse Kommentare in das Historie-Fenster einfügt. Zu diesen Ereignissen, bei denen automatisch Kommentare erzeugt werden, gehören die Umwandlung in eine neue Version von LabVIEW oder BridgeVIEW, Änderungen von SubVIs und Änderungen am Namen oder Pfad des VIs.

Revisionsnummer in der Titelleiste anzeigen — Fügt die Revisionsnummer in der Kopfzeile des Historie-Fensters ein. Die Revisionsnummer fängt bei Null an und nimmt jedesmal inkremental zu, wenn Sie das VI speichern. Sie wird jedoch nicht erhöht, wenn Sie lediglich an der VI-Historie Änderungen vorgenommen haben.

Die folgenden Anmeldeoptionen stehen zur Verfügung:

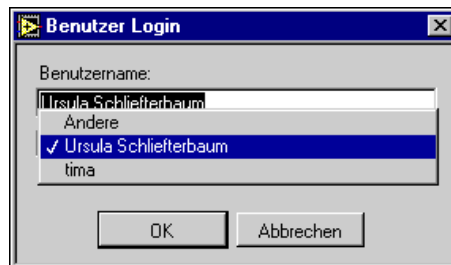
(Nur Windows und Macintosh) Einloggen mit dem Namen der LabVIEW-Registrierung — Benutzt den Namen des Benutzers, unter dem die Anwendung registriert ist.

Loginabfrage beim Start von LabVIEW anzeigen — Fordert beim Starten der Anwendung einen Benutzernamen von Ihnen an. Der Benutzername kann über **Bearbeiten»Benutzername** jederzeit geändert werden. Die Aufforderung zeigt ein Menü aller Benutzernamen an, die mit dieser Option zuvor in LabVIEW eingegeben worden sind.



Hinweis

Diese Funktion ist nur bei LabVIEW gegeben. BridgeVIEW verfügt über ein eigenes Anmeldesicherheitssystem, in dem Bearbeiten»Benutzername nicht erscheint.



Sie können unter Windows oder auf einem Macintosh zu einem anderen Benutzernamen (Benutzername des Systems oder der Name der Anwendungsregistrierung) bzw. unter UNIX zum Benutzernamen des Systems wechseln, ohne diesen eingeben zu müssen, indem Sie wie in der Abbildung oben gezeigt einen Namen aus dem Pulldown-Menü wählen. Sie können auch einen Namen eingeben, der dann danach in der Liste erscheint.

(Windows NT, Macintosh nur mit Dateifreigabe und UNIX) Einloggen mit dem Benutzernamen des Systems — Vorausgesetzt, daß ein Benutzername definiert ist, verwendet die Software bei dieser Option zur Anmeldung das Systemkonto des aktuellen Benutzers.

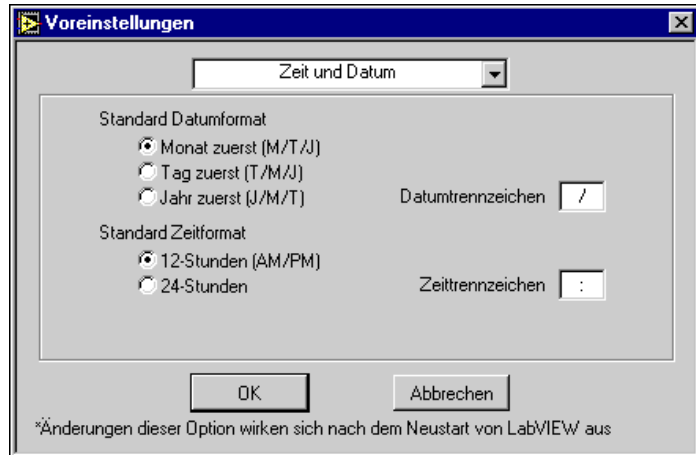


Hinweis

Änderungen an Optionen im Dialogfeld "Historie Voreinstellungen" treten sofort in Kraft.

Voreinstellungen für Uhrzeit und Datum

Die folgende Abbildung zeigt das Dialogfeld "Voreinstellungen/Datum und Uhrzeit".



Die Optionen in diesem Dialogfeld steuern die Standardanzeige der Uhrzeit und des Datums in den digitalen Anzeigen neuer Bedien- und Anzeigeelemente. Deaktivieren Sie die Standardwerte in diesem Dialogfeld für einzelne digitale Anzeigen wie im Abschnitt *Format und Präzision von digitalen Anzeigen* von Kapitel 9, *Numerische Bedien- und Anzeigeelemente* beschrieben.

In diesem Dialogfeld stehen die folgenden Optionen zur Verfügung:

Standard Datumformat — Legt fest, ob der Monat, Tag oder das Jahr in digitalen Anzeigen an erster Stelle stehen soll.

Standard Zeitformat — Legt fest, ob die Uhrzeit in digitalen Anzeigen im 12- oder 24-Stunden-Takt angegeben werden soll.

Datentrennzeichen — Legt das Zeichen fest, mit dem Monat, Tag und Jahr (in der bei der Option "Standard Datumformat" gewählten Reihenfolge) in digitalen Anzeigen voneinander getrennt werden.

Zeittrennzeichen — Legt das Zeichen fest, mit dem die Stunden und Minuten in digitalen Anzeigen voneinander getrennt werden.

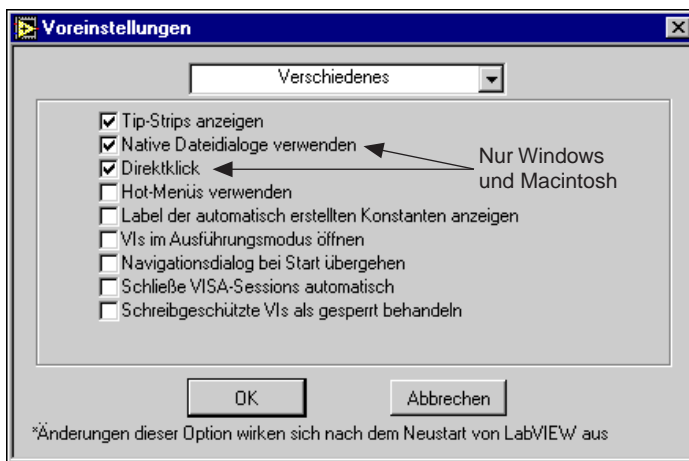


Hinweis

Änderungen an Optionen im Dialogfeld "Voreinstellungen/Datum und Uhrzeit" treten sofort in Kraft.

Verschiedene Voreinstellungen

Die folgende Abbildung zeigt das Dialogfeld “Verschiedene Voreinstellungen”.



In diesem Dialogfeld stehen die folgenden Optionen zur Verfügung:

Tip-Strips anzeigen — Schaltet die Anzeige von Hinweisstrips um.

(Windows, Macintosh) Native Dateialoge verwenden — Benutzt die Datei-Dialogfelder des Betriebssystems, damit diese denen anderer Anwendungen auf Ihrem Computer ähnlich sind (mit einigen wenigen Ausnahmen, die im nächsten Absatz behandelt werden). Wird diese Voreinstellung nicht gewählt, benutzt die Software der Sprache G ihre eigenen, plattformunabhängigen Datei-Dialogfelder, die über einige bequeme Funktionen verfügen, wie z.B. eine Liste der zuletzt aktiven Pfade und eine Reduzierung der Schritte, die zum Speichern von VIs in VI-Bibliotheken erforderlich sind.

(Windows, Macintosh) Direktklick — Sie können G so einstellen, daß ein Objekt in einem inaktiven Fenster durch einen einzigen Mausklick aktiviert und ausgewählt wird. Dies reduziert die Anzahl der erforderlichen Mausklicks von zwei Mausklicks auf einen. (Normalerweise wird mit dem ersten Klick das Fenster aktiviert, und der zweite Klick zählt dann als Klick in dem Fenster.)

Hot-Menüs verwenden — Mit dieser Option können Sie die Menüs mit der Maus durchgehen, ohne die linke Maustaste gedrückt zu halten. Diese ergonomische Funktion senkt die Belastung Ihrer Hand.

Label der automatisch erstellten Konstanten anzeigen — Hiermit wird die Erstellung von Labels für automatisch erstellte Konstanten umgeschaltet. Ist diese Voreinstellung gesetzt, zeigen Labels automatisch den Namen des Terminals an, aus dem die Konstante erstellt worden ist.

VI im Ausführungsmodus öffnen — Öffnet VIs nicht im Bearbeitungsmodus, sondern im Ausführungsmodus.

Navigationsdialog bei Start übergehen — Überspringt das erste Fenster, das eine Schnittstelle zu weiteren allgemeinen Dialogfeldern darstellt, und öffnet sofort das Fenster “Unbenanntes VI”.

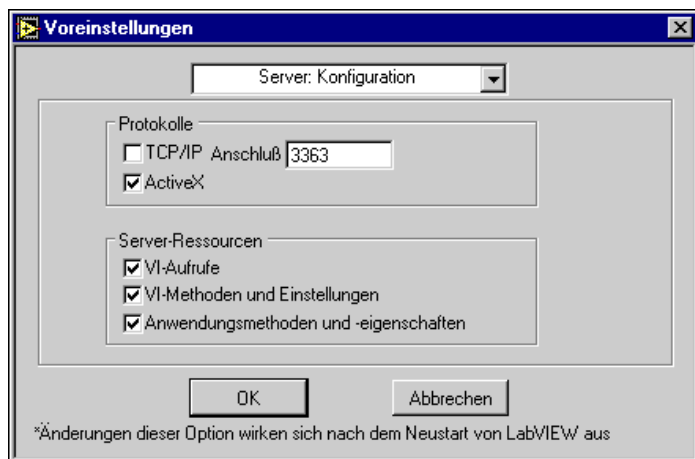


Hinweis

Änderungen an Optionen im Dialogfeld “Verschiedene Voreinstellungen” treten sofort in Kraft.

Server: Konfiguration

Die folgende Abbildung zeigt das Dialogfeld “Server: Konfiguration”.



Diese Optionen geben an, mit welchem Kommunikationsprotokoll andere Anwendungen auf den VI-Server zugreifen, und zwar entweder über TCP/IP oder ActiveX. Wenn Sie **TCP/IP** aktivieren, müssen Sie die **Anschluß**-Nummer eingeben, die Client-Anwendungen verwenden, um eine Verbindung zum Server herzustellen. Wenn Sie anderen Anwendungen erlauben, Verbindungen über TCP/IP herzustellen, ist es am besten, zu konfigurieren, welche Internet-Hosts Zugriff auf den Server haben. Weitere Einzelheiten hierzu finden Sie im Abschnitt [Server: TCP/IP-Zugriff](#) in diesem Kapitel. Weitere Informationen über die ActiveX-Schnittstelle für VI-Server finden Sie in Kapitel 21, [VI-Server](#).

Wird **Server: Konfiguration** gewählt, geben Sie auch an, welche der folgenden Serverressourcen Anwendungen zur Verfügung stehen, die auf den VI-Server zugreifen.

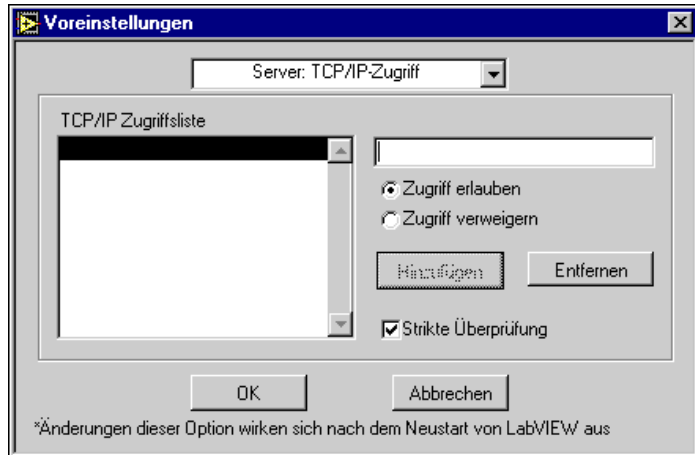
- **VI-Aufrufe** erlaubt Anwendungen, Aufrufe an VIs zu richten, die auf dem Server exportiert sind. Wenn Sie den Zugriff auf VIs durch andere Anwendungen zulassen, ist es am besten, mit Hilfe des Dialogfeldes “Voreinstellungen/Exported VIs” anzugeben, welche VIs exportiert sind. Weitere Informationen hierzu finden Sie im Abschnitt *Server: Exportierte VIs* in diesem Kapitel.
- **VI-Methoden und Einstellungen** erlaubt Anwendungen, VI-Eigenschaften auf dem Server zu lesen und festzulegen. Auch hierbei können Sie im Dialogfeld “Voreinstellungen/Exported VIs die exportierten VIs angeben. Weitere Informationen hierzu finden Sie im Abschnitt *Server: Exportierte VIs* in diesem Kapitel.
- **Anwendungsmethoden und Einstellungen** erlaubt Anwendungen, die Eigenschaften auf dem Server zu lesen und festzulegen.

Bei den standardmäßigen Servereinstellungen ist **ActiveX** aktiviert und **TCP/IP** deaktiviert. Ferner ist **VI-Aufrufe** standardmäßig aktiviert, aber **VI-Methoden und Einstellungen** und **Anwendungsmethoden und Einstellungen** sind deaktiviert.

Server: TCP/IP-Zugriff

Wenn Sie Remote-Anwendungen den Zugriff auf den VI-Server via TCP/IP erlauben, ist es am besten, anzugeben, welche Internet-Hosts Zugriff auf den Server haben.

Die folgende Abbildung zeigt das Dialogfeld **Server: TCP/IP-Zugriff**.



Mit **Server: TCP/IP-Zugriff** geben Sie an, welche Clients Zugriff auf den VI-Server haben sollen. Die "TCP/IP-Zugriffsliste" beschreibt Clients, denen der Zugriff auf den Server gewährt oder verweigert ist. Um einen Eintrag zu ändern, wählen Sie ihn aus der Liste, und geben Sie die Änderung im Textfeld rechts neben der TCP/IP-Zugriffsliste ein. Klicken Sie auf das Optionsfeld **Zugriff erlauben**, um dem Client den Zugriff auf den Server zu gestatten. Klicken Sie auf das Optionsfeld **Zugriff verweigern**, um dem Client den Zugriff auf den Server zu verweigern. Klicken Sie auf die Taste **Hinzufügen**, um nach der aktuellen Auswahl einen Eintrag einzufügen. Klicken Sie auf die Taste **Entfernen**, um die aktuelle Auswahl von der Liste zu entfernen. Ziehen Sie einen Eintrag bei gedrückter Maustaste, um seine Position innerhalb der TCP/IP-Zugriffsliste zu ändern. Ist in einem Eintrag der Zugriff von einer Adresse aus gestattet, wird neben dem Eintrag ein Häkchen angezeigt. Wird in einem Eintrag der Zugriff von einer Adresse aus verweigert, wird neben dem Eintrag ein **X** angezeigt. Erscheint neben einem Eintrag kein Symbol, ist die Syntax des Eintrags falsch.

Versucht ein Client, eine Verbindung zum Server zu öffnen, überprüft der Server die Einträge in der TCP/IP-Zugriffsliste, um festzustellen, ob dem Client der Zugriff gestattet ist. Stimmt die Client-Adresse mit einem Eintrag in der Liste überein, gewährt oder verweigert der Server den Zugriff, je nachdem, wie der Eintrag definiert ist. Stimmt die Client-Adresse mit einem weiteren Eintrag überein, wird statt der ersten diese zweite Befugnis verwendet. In der obigen Abbildung ist sowohl `a.test.site.com` als auch `b.test.site.com` der Zugriff gestattet, obwohl in der Liste (mit Hilfe des Platzhalters *) angezeigt wird, daß keiner der Adressen, die auf `.test.site.com` enden, Zugriff gestattet wird.

Stimmt kein Eintrag mit der Client-Adresse überein, wird der Zugriff verweigert. Weitere Informationen über den Platzhalter "*" und den Zugriff von übereinstimmenden Einträgen finden Sie in Tabelle 7-1.

Eine Internet-Adresse (IP-Adresse), wie z.B. 130.164.123.123, kann einen oder mehrere zugeordnete Domännennamen haben, wie z.B. www.natinst.com. Die Konvertierung eines Domännennamens in die entsprechende IP-Adresse wird als *Namensauflösung* bezeichnet. Die Konvertierung einer IP-Adresse in ihren Domännennamen wird als *Namensabgleich (Lookup)* bezeichnet.

Namensabgleich und -auflösung erfolgen mittels Systemaufrufen, die auf DNS-Server (Domain Name System) im Internet zugreifen. Ein Namensabgleich oder eine Namensauflösung können fehlschlagen, wenn das System keinen Zugriff auf einen DNS-Server hat oder wenn die Adresse oder der Name nicht richtig sind. Ein Auflösungsproblem tritt auf, wenn ein Eintrag einen Domännennamen enthält, der nicht in eine IP-Adresse aufgelöst werden kann. Ein Abgleichproblem tritt auf, wenn ein Eintrag einen unvollständigen Domännennamen enthält, wie z.B. *.natinst.com, und der Abgleich für die IP-Adresse des Clients fehlschlägt.

Strikte Überprüfung bestimmt, wie der Server mit Einträgen in der Zugriffsliste verfährt, die aufgrund von Auflösungs- oder Abgleichproblemen nicht mit einer IP-Clientadresse verglichen werden können. Ist die Funktion "Strict Checking" aktiviert, wird ein Eintrag in der TCP/IP-Zugriffsliste, dem der Zugriff verweigert wird und bei dem ein Auflösungsproblem vorliegt, so behandelt, als stimmte die IP-Adresse des Clients mit dem Eintrag überein. Ist die Option **Strikte Überprüfung** deaktiviert, wird ein Eintrag in der Zugriffsliste, bei dem ein Auflösungsproblem auftritt, ignoriert.

Um eine Internet-Hostadresse einzugeben, geben Sie den Domännennamen oder die IP-Adresse ein. Benutzen Sie den Platzhalter "*" zur Angabe von Internet-Hostadressen. Sie können beispielsweise alle Hosts innerhalb der Domäne domäne.com mit dem Eintrag *.domäne.com spezifizieren. Sie können alle Hosts im Teilnetz, die mit 130.164 anfangen, mit dem Eintrag 130.164.* spezifizieren. Der Eintrag * stimmt mit allen Adressen dieser Domäne überein.

Die folgende Tabelle enthält Beispiele für Einträge in der TCP/IP-Zugriffsliste.

Tabelle 7-1. Server: TCP/IP-Zugriff

Zugriffs-String	Übereinstimmungen
*	Alle Hosts
test.site.com	der Host mit dem Domännennamen test.site.com
*.site.com	alle Hosts, deren Domännennamen mit *.site.com enden
130.164.123.123	der Host mit der IP-Adresse 130.164.123.123
130.164.123.*	alle Hosts, deren IP-Adresse mit 130.164.123. beginnt

Laut der vorherigen Abbildung haben alle Hosts in der Domäne site.com Zugriff auf den Server, mit Ausnahme aller Hosts der Domäne test.site.com. Außerdem haben die Hosts a.test.site.com, b.test.site.com und 130.164.123.123 ebenfalls Zugriff auf den Server. Der Host public.site.com hat keinen Zugriff, obgleich der sich in der Domäne site.com befindet.

Die standardmäßigen TCP/IP-Zugriffseinstellungen gewähren nur Clients Zugriff auf den Serverrechner.



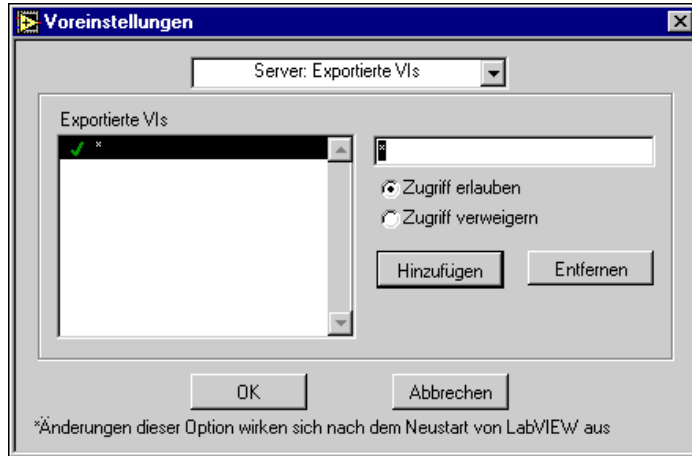
Hinweis

Wenn der VI-Server auf einem System ausgeführt wird, das keinen Zugriff auf einen DNS-Server hat, verwenden Sie keine Domännennamen in der TCP/IP-Zugriffsliste. Anforderungen zur Auflösung des Domännennamens oder einer IP-Adresse werden fehlschlagen und das System verlangsamen. Geben Sie häufig abgeglichene Einträge aus Gründen der Systemleistung weiter unten in der TCP/IP-Zugriffsliste ein.

Server: Exportierte VIs

Wenn Sie Remote-Anwendungen den Zugriff auf VIs auf dem VI-Server gestatten, ist es am besten, anzugeben, auf welche VIs diese Anwendungen zugreifen dürfen.

Die folgende Abbildung zeigt das Dialogfeld **Server: Exportierte VIs**.



Mit den Menüpunkten im Dialogfeld **Server: Exportierte VIs** können Sie angeben, auf welche VIs andere Anwendungen über den VI-Server zugreifen können. In der Liste “Exportierte VIs” werden die exportierten VIs aufgeführt. Um einen Eintrag zu ändern, wählen Sie ihn aus der Liste, und geben Sie die Änderung im Textfeld rechts neben der Liste “Exported VIs” ein. Klicken Sie auf das Optionsfeld **Zugriff erlauben** oder **Zugriff verweigern**, um anzugeben, ob Remote-Computer Zugriff auf das betreffende VI haben oder nicht. Klicken Sie auf die Taste **Hinzufügen**, um nach der aktuellen Auswahl einen Eintrag einzufügen. Klicken Sie auf die Taste **Entfernen**, um die aktuelle Auswahl von der Liste zu entfernen. Ziehen Sie einen Eintrag bei gedrückter Maustaste, um seine Position innerhalb der Liste “Exportierte VIs” zu ändern. Wird mit einem Eintrag der Zugriff auf VIs gestattet, wird neben dem Eintrag ein Häkchen angezeigt. Wird mit einem Eintrag der Zugriff auf VIs verweigert, wird neben dem Eintrag ein **X** angezeigt. Erscheint neben einem Eintrag kein Symbol, ist die Syntax des Eintrags falsch.

Jeder Eintrag in der Liste beschreibt einen VI-Namen oder einen VI-Pfad und kann Platzhalter enthalten. Einträge, die Pfad-Trennzeichen enthalten, werden mit VI-Pfaden verglichen, während Einträge, die keine Pfad-Trennzeichen aufweisen, nur mit VI-Namen verglichen werden. Wenn ein Remote-Client versucht, auf ein VI zuzugreifen, überprüft der Server die Einträge in der Liste “Exported VIs”, um festzustellen, ob der Zugriff auf das angeforderte VI gewährt werden soll. Stimmt das angeforderte VI mit einem Eintrag in der Liste überein, gewährt oder verweigert der Server den Zugriff auf das VI, je nachdem, wie der Eintrag definiert ist. Stimmt das angeforderte VI mit einem weiteren Eintrag in der

Liste überein, wird statt der ersten diese zweite Befugnis verwendet. Stimmt kein Eintrag mit dem angeforderten VI überein, wird der Zugriff auf das VI verweigert.

Sie können in der Liste “Exportierte VIs” Platzhalter verwenden, damit mehrere VIs in der Liste mit einem einzigen Eintrag spezifiziert werden können. Benutzen Sie hierbei die folgenden Zeichen als Platzhalter:

‘?’	Exakte Übereinstimmung mit einem beliebigen Zeichen, mit Ausnahme des Pfad-Trennzeichens
‘*’	Übereinstimmung mit null oder mehreren beliebigen Zeichen, mit Ausnahme des Pfad-Trennzeichens
‘**’	Übereinstimmung mit null oder mehreren beliebigen Zeichen einschließlich des Pfad-Trennzeichens

Wenn Sie ein VI mit einem Namen vergleichen wollen, der einen Platzhalter beinhaltet, müssen Sie dieses Zeichen mit ‘\’ (Macintosh und UNIX), oder ‘\\’ (Windows) ausschließen.

Die folgende Tabelle enthält Beispiele für Listeneinträge unter “Exportierte VIs”. Die Beispiele benutzen UNIX-Pfad-Trennzeichen.

Tabelle 7-2. Server: Listeneinträge für exportierte VIs

VI-Zugriffs-String	Übereinstimmungen
*	Alle VIs
/usr/labview/*	Alle VIs im Verzeichnis /usr/labview/.
/usr/labview/**	Alle VIs im Verzeichnis /usr/labview/ oder in einem der Unterverzeichnisse
test.vi	ein VI mit dem Namen Test.vi.
OK\?	ein VI mit dem Namen OK?

Laut der vorherigen Abbildung sind alle VIs im Verzeichnis c:\labview\server exportiert. Alle VIs im Verzeichnis c:\labview\test und allen Unterverzeichnissen sind ebenfalls exportiert, mit Ausnahme des VIs c:\labview\test\private.vi. Außerdem sind alle VIs, die mit dem

String `srvr_` anfangen und mit dem String `.vi` enden, ebenfalls exportiert. Kein VI, das mit dem String `local_` anfängt und mit dem String `.vi` endet, ist exportiert, auch dann nicht, wenn es sich im Verzeichnis `c:\labview\server` befindet.

Die standardmäßigen Einstellungen für “Exportierte VIs” gewähren Zugriff auf alle VIs.

Wie werden Voreinstellungen gespeichert?

Es ist gewöhnlich nicht erforderlich, Voreinstellungsinformationen manuell zu bearbeiten oder ihr genaues Format zu wissen, da das Dialogfeld “Voreinstellungen” dies für Sie erledigt. Voreinstellungen werden je nach Plattform unterschiedlich gespeichert, wie in den folgenden Absätzen beschrieben ist.

(Windows) Wenn Sie LabVIEW-Benutzer sind, werden die Voreinstellungen in einer Datei `LabVIEW.INI` im LabVIEW-Verzeichnis gespeichert. Das Dateiformat ist dem anderer `.INI`-Dateien, wie z.B. der Datei `WIN.INI`, ähnlich. Die Datei beginnt mit dem Abschnittsmarker `[LabVIEW]`.

Dem schließen sich Variablenlabels und deren Werte an, wie z.B. `offscreenUpdates=True`. Ist ein Konfigurationswert nicht in `LABVIEW.INI` definiert, prüft LabVIEW, ob die Datei `WIN.INI` einen Abschnitt mit dem Titel `[LabVIEW]` enthält, und ist dies der Fall, wird der Konfigurationswert in dieser Datei überprüft.

Wenn Sie BridgeVIEW-Benutzer sind, werden die Voreinstellungen in einer Datei `BridgeVIEW.INI` im BridgeVIEW-Verzeichnis gespeichert. Das Dateiformat ist dem anderer `.INI`-Dateien ähnlich. Die Datei beginnt mit dem Abschnittsmarker `[BridgeVIEW]`.



Hinweis

Die Verweise auf `WIN.INI`-Dateien gelten nur für Windows 3.1. Diese Informationen gelten nicht für BridgeVIEW-Benutzer.

Sie können auch beim Starten von LabVIEW oder BridgeVIEW eine Voreinstellungsdatei in der Befehlszeile angeben. Um beispielsweise statt der Datei `labview.ini` eine Datei namens `lvrc` zu benutzen, geben Sie `labview -pref lvrc` ein.

(Macintosh) LabVIEW erstellt im Ordner “Voreinstellungen” des Systemordners standardmäßig eine Textdatei mit dem Namen `LabVIEW Preferences`.

Wenn Sie wollen, können Sie die Datei "LabVIEW-Voreinstellungen" in den LabVIEW-Ordner kopieren. Beim Starten der Software sucht LabVIEW im LabVIEW-Ordner nach der Voreinstellungsdatei. Wenn diese dort nicht gefunden werden kann, sucht die Software im Ordner "Voreinstellungen", und wenn sie auch dort nicht gefunden werden kann, wird sie im Ordner "Voreinstellungen" erstellt. Durch Verschieben der Voreinstellungsdatei zwischen dem Schreibtisch und dem LabVIEW-Ordner können Sie für mehrere Benutzer oder Verwendungszwecke mehrere Voreinstellungen erstellen.

(UNIX) Voreinstellungsinformationen werden normalerweise in Ihrem Stammverzeichnis in einer Textdatei namens `.labviewrc` gespeichert. Wenn Sie einen Parameter im Dialogfeld "Voreinstellungen" ändern, schreibt G die Informationen in diese Datei. Die folgenden Ausführungen richten sich an diejenigen, die weitere Einzelheiten über das Speicherformat und über Regeln zum Suchen von Voreinstellungsinformationen erfahren wollen.

Voreinstellungseinträge bestehen aus einem *Voreinstellungsnamen*, dem ein Doppelpunkt und ein Wert folgt. Der Voreinstellungsname ist der Programmname, gefolgt von einem Punkt (.) und einem Token. Die Suche von Voreinstellungsnamen erfolgt bei LabVIEW ohne Beachtung von Groß- und Kleinschreibung. Sie haben die Wahl, den Voreinstellungswert in doppelte oder einfache Anführungszeichen zu setzen. Um beispielsweise "double" als standardmäßige Präzision vorzugeben und als Suchpfad einer rekursiven Suche ein bestimmtes Verzeichnis anzugeben, geben Sie die folgenden Voreinstellungseinträge an.

```
labview.defPrecision : double
labview.viSearchPath : "/usr/lib/labview/*"
```

LabVIEW sucht Voreinstellungen auch in einer Datei namens `labviewrc`, die sich im Anwendungsverzeichnis befindet. Sind die LabVIEW-Dateien beispielsweise in `/opt/labview` installiert, werden sowohl aus der Datei `/opt/labview/labviewrc` als auch der Datei `.labviewrc` im Stammverzeichnis Voreinstellungen gelesen.

Bei Widersprüchen haben Einträge in der Datei `.labviewrc` Vorrang vor Einträgen in der Datei `labviewrc`. Sie können diese globale Voreinstellungsdatei eventuell zur Speicherung von Parametern verwenden, die für alle Benutzer gelten, wie z.B. den VI-Suchpfad.

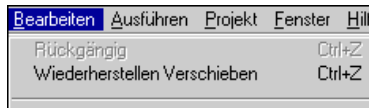
Sie können auch beim Starten von LabVIEW eine Voreinstellungsdatei in der Befehlszeile angeben. Um beispielsweise eine Datei namens `lvrc` zu benutzen, geben Sie `labview -pref lvrc` ein.

**Vorsicht**

Die Datei .labviewrc wird in diesem Fall trotzdem gelesen. Beachten Sie auch, daß Änderungen, die im Dialogfeld “Voreinstellungen” vorgenommen werden, in diesem Beispiel in die Datei lvrc geschrieben werden.

Rückgängig machen

Mit den Menüpunkten **Bearbeiten»Rückgängig** und **Bearbeiten»Wiederherstellen** können Sie eine Aktion unmittelbar nach ihrer Durchführung rückgängig machen bzw. eine rückgängig gemachte Aktion wiederherstellen. Der Tastaturbefehl für **Rückgängig** ist <Strg-Z> (bzw. auf dem Macintosh <Befehlstaste-Z>), und der Tastaturbefehl für **Wiederherstellen** ist <Strg-Umschalt-Z> (bzw. auf dem Macintosh <Befehlstaste-Umschalt-Z>). Die Menüpunkte **Rückgängig** und **Wiederherstellen** im Menü **Bearbeiten** enthalten eine kurze Beschreibung (in einem oder zwei Worten) der Aktion, die rückgängig gemacht bzw. wiederhergestellt werden kann. Wenn Sie den Menüpunkt **Rückgängig: Verschieben** wählen würden, der in der nachstehenden Abbildung zu sehen ist, würde das betreffende Objekt an seine ursprüngliche Position zurückversetzt werden.



Mit **Voreinstellungen»Blockdiagramm»Anzahl Rückgängig pro VI** können Sie die Anzahl der Aktionen festlegen, die rückgängig gemacht bzw. wiederhergestellt werden können. Weitere Informationen über das Festlegen der Schritte, die bei einem VI maximal rückgängig gemacht werden können, finden Sie im Abschnitt [Festlegen von Voreinstellungen](#) unter [Voreinstellungen für Blockdiagramme](#) in diesem Kapitel.

Jedes VI zeichnet seine eigenen Instanzen für die Operation “Rückgängig machen” auf, d.h. die Anzahl der Aktionen, die Sie unter **Anzahl Rückgängig pro VI** eingeben, gilt pro VI. Mit einem kleinen Zahlenwert lassen sich Speicherressourcen einsparen.

VI-Aufrufende werden automatisch aktualisiert, wenn sich die Schnittstelle des VIs ändert. In dieser Situation werden alle Informationen

über Aktionen, die rückgängig gemacht werden können, verworfen. Dies ist bei den folgenden Vorgängen der Fall:

- wenn Sie ein Muster eines VI-Anschlußfeldes oder den Datentyp eines Bedienelements im Anschlußfeld ändern;
- wenn Sie das Aussehen einer Typendefinition (nur sofern es strikt ist), den Datentyp oder die Striktheit oder eine Typendefinition in ein benutzerspezifisches Bedienelement ändern;
- wenn Sie den Datentyp oder Namen eines Bedienelements in einem globalen VI ändern oder einem globalen VI ein Bedienelement hinzufügen oder daraus löschen.

Anpassen der Elemente- und Funktionenpaletten

G besitzt einen leistungsfähigen Satz von Werkzeugen, mit denen die Paletten **Elemente** und **Funktionen** folgendermaßen angepaßt werden können.

- Hinzufügen Ihrer eigenen VIs und Bedienelemente zu den Paletten, indem Sie sie im Verzeichnis `user.lib` oder `instr.lib` speichern.
- Hinzufügen Ihrer eigenen VIs und Bedienelemente zu den Paletten mit Hilfe der Option **Bearbeiten»Bedienelemente- und Funktionenpalette bearbeiten**.
- Einrichten der Paletten in einer Art und Weise, daß diese sich automatisch ändern, wenn in bestimmten Verzeichnissen Dateien hinzugefügt bzw. gelöscht werden.
- Neuordnen der integrierten Paletten, um häufig verwendete Funktionen leichter zugänglich zu machen.
- Einrichten unterschiedlicher Ansichten auf Benutzerbasis, Verbergen einiger Funktionen, um G für einen Benutzer zu vereinfachen, während einem anderen Benutzer der vollständige Funktionsumfang zur Verfügung gestellt wird.
- Unter Windows und auf dem Macintosh wird die Palette **Funktionen** verborgen, wenn ein Frontpanel das aktive Fenster ist, und die Palette **Elemente** wird verborgen, wenn ein Blockdiagramm das aktive Fenster ist. Wenn Sie es vorziehen, daß stets alle Paletten sichtbar sind, bearbeiten Sie Ihre Voreinstellungsdatei, und fügen Sie ihr eine Zeile hinzu, mit der die Voreinstellung `showAllPalettes` auf `True` gesetzt wird.

Hinzufügen von VIs und Bedienelementen zu `user.lib` und `instr.lib`

Die einfachste Methode, den Paletten **Elemente** und **Funktionen** neue Einträge hinzuzufügen, besteht darin, sie im Verzeichnis `user.lib` zu speichern. Wenn Sie G neu starten, enthält die Unterpalette **User Libraries** der Palette **Funktionen** für jedes Verzeichnis, jede `.lib`- oder `.mnu`-Datei im Verzeichnis `user.lib` und Einträge für jede Datei im Verzeichnis `user.lib`.

Die Palette **Instrumententreiber** der Palette **Funktionen** entspricht dem Verzeichnis `instr.lib`. Sie können Instrumententreiber in diesem Verzeichnis ablegen, um sie von den Paletten aus leicht zugänglich zu machen.



Hinweis

Das Verzeichnis `vi.lib` enthält Dateien von National Instruments. Speichern Sie im Verzeichnis `vi.lib` keine Dateien, da diese überschrieben werden, wenn Sie die Software aktualisieren.

Installieren und Ändern von Ansichten

Informationen zu den Paletten **Elemente** und **Funktionen** werden im Verzeichnis `menus` des Anwendungsverzeichnisses gespeichert. Das Verzeichnis `menus` enthält Verzeichnisse zu jeder Ansicht, die Sie erstellen oder installieren. Wird die Anwendung über das Netzwerk ausgeführt, empfiehlt es sich unter Umständen, für jeden Benutzer eine individuelle Instanz des Verzeichnisses `menus` zu erstellen. Sie können Ihrer Voreinstellungsdatei eine Zeile hinzufügen, die die Voreinstellung für `menuDir` auf einen alternativen Pfad setzt, der für die Voreinstellungsdatei jedes Benutzers eindeutig ist.

Diese Organisation erleichtert die Übertragung von Ansichten an andere Benutzer. Um einem anderen Benutzer eine Kopie einer Ansicht zur Verfügung zu stellen, stellen Sie diesem eine Kopie des Verzeichnisses zur Verfügung, das die Ansicht aus dem Verzeichnis `menus` enthält. Nachdem Sie sie in das Verzeichnis `menus` des betreffenden Benutzers plaziert haben, steht die Ansicht beim nächsten Start zur Verfügung.

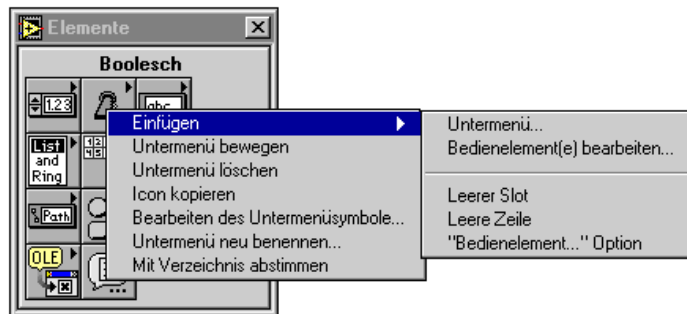
Wählen Sie **Bearbeiten»Bedienelemente- und Funktionenpalette bearbeiten**, um zu einer anderen Ansicht zu wechseln. Wählen Sie dann die gewünschte Ansicht aus dem Ring **menu setup**.

Informationen zum Erstellen Ihrer eigenen Ansicht können Sie der Beschreibung im nachfolgenden Abschnitt, *Paletten-Editor*, entnehmen. Weitere Einzelheiten über die Organisation von Ansichten finden Sie im Abschnitt *Funktionsweise von Ansichten* weiter unten in diesem Kapitel.

Paletten-Editor

Eine bessere Kontrolle über Layout und Inhalt der Paletten **Elemente** und **Funktionen** erhalten Sie mit der Option **Bearbeiten»Palettensatz bearbeiten...** Mit dieser Option wird der Paletten-Editor aufgerufen, und das Dialogfeld “Paletten bearbeiten” erscheint.

In diesem Editor können Sie den Inhalt von Paletten neu anordnen, indem Sie Objekte an andere Positionen ziehen. Wenn Sie Objekte löschen, anpassen oder einfügen möchten, rufen Sie das Popup-Menü einer Palette (siehe Abbildung unten) oder eines Objekts innerhalb der Unterpalette auf.



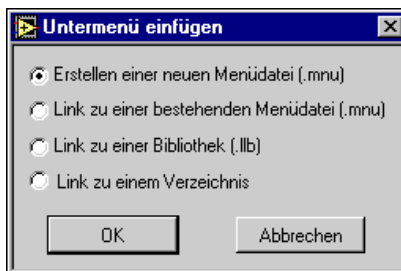
Mit den Optionen im Popup-Menü können Sie alle Optionen im Menü **Benutzer-Bibliotheken** (entspricht `user.lib`) oder im Menü **Geräte I/O** (entspricht `instr.lib`) ändern. Wenn Sie die Palette **Elemente** oder **Funktionen** der höchsten Ebene oder ein beliebiges anderes vordefiniertes Menü bearbeiten wollen, müssen Sie zunächst eine neue Ansicht erstellen, indem Sie **Neue Einstellungen...** aus dem Ring **menu setup** im Dialogfeld “Paletten bearbeiten” wählen. Sobald Sie neue Einstellungen gewählt haben, kopieren Sie jedes Menü, das Sie modifizieren und das Bestandteil des Standardmenüsatzes ist, in das Untermenü `menus` Ihrer Ansicht, bevor Sie Änderungen vornehmen. Diese Schutzmaßnahme für implementierte Paletten ermöglicht Ihnen, mit den Paletten zu experimentieren, ohne die Standardansicht zu zerstören.

Wenn Sie ein neues Objekt in eine neue Reihe oder Spalte einer Unterpalette einfügen wollen, rufen Sie an einer Stelle an der rechten Kante oder unten in der Unterpalette das Popup-Menü auf. Sie können auch neue Reihen oder Spalten erstellen, indem Sie Objekte in den Bereich rechts oder unten auf der Palette ziehen.

Erstellen von Unterpaletten

Wenn Sie eine Palette hinzufügen, können Sie sie mit dem Paletten-Editor an eine neue Position verschieben, das Icon der Unterpalette bearbeiten oder die Palette umbenennen.

Wenn Sie eine Palette von Grund auf neu erstellen oder an eine Palette “anknüpfen” wollen, die nicht in `user.lib` oder `vi.lib` gespeichert ist, können Sie dazu **Einfügen»Untermenü...** aus dem Popup-Menü im Paletten-Editor verwenden. Wenn Sie diese Option wählen, wird das folgende Dialogfeld angezeigt.



Sie können Informationen zu Untermenüs in VI-Bibliotheken oder in `.mnu`-Dateien speichern. Eine Menüdatei oder `.LLB`-Datei kann eine **Funktionen**-Palette und eine **Elemente**-Palette enthalten.

Wählen Sie **Erstellen einer neuen Menüdatei (.mnu)**, um eine neue, leere Palette einzufügen. Sie werden dazu aufgefordert, einen Palettennamen und einen Dateinamen anzugeben. Es wird empfohlen, daß Sie eine `.mnu`-Erweiterung an die Datei anfügen, um darauf hinzuweisen, daß es sich um ein Menü (Palette) handelt. Ferner wird empfohlen, daß Sie die `.mnu`-Datei im Verzeichnis Ihrer Ansicht im Unterverzeichnis `menus` oder in dem Verzeichnis speichern, das auch die Bedienelemente oder VIs enthält, die das Menü hauptsächlich repräsentiert.

Wählen Sie **Link zu einer bestehenden Menüdatei (.mnu)**, wenn Sie über eine bestehende Palette verfügen, die Sie zur Palettendatei für die Palette “Bedienelemente” oder “Funktionen” hinzufügen wollen.

Wählen Sie **Link zu einem Verzeichnis**, wenn Sie eine Palette mit Einträgen für alle Dateien erstellen wollen, die sich in dem Verzeichnis befinden. Durch die Auswahl dieser Option werden rekursiv für jedes Unterverzeichnis, jede VI-Bibliothek oder `.mnu`-Datei in diesem Verzeichnis auch Unterpaletten erstellt. Diese Paletten werden automatisch aktualisiert, wenn Sie in den gewählten Verzeichnissen neue Dateien

hinzufügen oder bestehende löschen. Aktivieren oder deaktivieren Sie die automatische Aktualisierung für eine Unterpalette, indem Sie im Popup-Menü die Option **Automatic Update From Directory** auswählen. Mit diesem Menüpunkt werden eigentlich in jedem Verzeichnis Menüdateien (Palettendateien) erstellt.

Wählen Sie **Link zu einer Bibliothek (.lib)**, wenn Sie eine Verknüpfung zur Palette **Elemente** oder **Funktionen** herstellen wollen, die Teil einer VI-Bibliothek sein soll. Wie bei der obigen Einstellung wird die Bibliothek beim Hinzufügen von Dateien automatisch aktualisiert.

Beachten Sie, daß Sie VIs, Funktionen und Unterpaletten in einer Palette frei kombinieren können. Eine Palette kann auch VIs verschiedener Speicherstellen enthalten.

Verschieben von Unterpaletten

Da eine Unterpalette durch Klicken geöffnet wird, läßt sie sich nicht durch Ziehen verschieben. Wählen Sie zum Verschieben einer Unterpalette den Menüpunkt **Untermenü bewegen** aus deren Popup-Menü. Halten Sie als Tastenkombination die <Umschalt>-Taste gedrückt, während Sie auf eine Unterpalette klicken, um Sie zu ziehen, anstatt sie zu öffnen.

Funktionsweise von Ansichten

Sowohl .mnu-Dateien als auch VI-Bibliothekdateien sind binäre Dateien, die eine Palette **Elemente** und eine Palette **Funktionen** enthalten können. Außerdem können beide Arten von Dateien ein Icon für die Paletten **Elemente** und **Funktionen** enthalten. Jedes Untermenü, das Sie erstellen, muß in einer separaten Datei gespeichert werden.

Wenn Sie eine Ansicht wählen, sucht G im Verzeichnis `menus` nach einem Verzeichnis. G erstellt die Paletten **Elemente** und **Funktionen** auf höchster Ebene anhand der Datei `root.mnu` in diesem Verzeichnis. Die Paletten der höchsten Ebene werden dann für jede Unterpalette innerhalb der Palette mit anderen .mnu- oder VI-Bibliothekdateien verknüpft.

Bei der Verknüpfung mit einem Verzeichnis wird eine Suche eingeleitet, die prüft, ob das Verzeichnis eine Datei namens `dir.mnu` enthält. Ist dies der Fall, wird diese .mnu-Datei als Unterpalette dieses Verzeichnisses verwendet. Andernfalls erstellt Ihre Anwendung anhand des Verzeichnisinhalts eine Datei mit dem Namen `dir.mnu`. Für jedes VI (oder Bedienelement) wird ein Eintrag erstellt. Für jedes Unterverzeichnis, jede .mnu-Datei oder VI-Bibliothek erstellt Ihre Anwendung eine Unterpalette.

G aktualisiert die Paletten innerhalb einer VI-Bibliothek automatisch, wenn Dateien zur VI-Bibliothek hinzugefügt oder daraus entfernt werden. Sie haben die Wahl, eine .mnu-Datei festzulegen, die anhand des Inhalts eines Verzeichnisses aktualisiert wird. Um diese Einstellung zu ändern, wählen Sie **Mit Verzeichnis abstimmen** aus dem Popup-Menü des Unterpalettensymbols. Obgleich es eine gute Idee ist, die .mnu-Datei im selben Verzeichnis zu belassen, mit dem sie synchronisiert ist, können Sie eigentlich ein beliebiges Verzeichnis auswählen. Abgesehen davon, daß ein bestimmtes Verzeichnis widergespiegelt wird, können Sie zusätzliche VIs oder Bedienelemente aus anderen Verzeichnissen in dieselbe Palette einfügen. Sie können ferner VIs von der Palette entfernen, indem Sie **Objekt löschen** aus dem Popup-Menü wählen.

Frontpanel-Objekte

Dieser Abschnitt enthält Informationen über Frontpanel-Objekte, Bedien- und Anzeigeelemente und ActiveX-Bedienelemente.

Teil II, *Frontpanel-Objekte*, enthält folgende Kapitel.

- Kapitel 8, *Einführung in Frontpanel-Objekte*, bietet eine Einführung in das Frontpanel und seine zwei Komponenten—Bedien- und Anzeigeelemente. Außerdem erläutert es, wie man Grafiken von anderen Programmen zur Verwendung in Ihren Bedienelementen importieren kann.
- Kapitel 9, *Numerische Bedien- und Anzeigeelemente*, beschreibt, wie man die verschiedenen Stile der numerischen Bedien- und Anzeigeelemente erstellt, ausführt und konfiguriert.
- Kapitel 10, *Boolesche Bedien- und Anzeigeelemente*, beschreibt, wie man Boolesche Bedien- und Anzeigeelemente erstellt, ausführt und konfiguriert.
- Kapitel 11, *Bedien- und Anzeigeelemente vom Typ String*, beschreibt, wie man Bedien- und Anzeigeelemente vom Typ String und die Tabelle verwendet. Diese Objekte können über die Palette **Elemente»String & Tabelle** aufgerufen werden.
- Kapitel 12, *Bedien- und Anzeigeelemente vom Typ Pfad und Refnum*, beschreibt, wie Dateipfad-Bedienelemente und Refnums verwendet werden, die in der Palette **Elemente»Pfad & RefNum** zur Verfügung stehen.
- Kapitel 13, *Bedien- und Anzeigeelemente vom Typ Liste und Ring*, beschreibt die Bedien- und Anzeigeelemente vom Typ Listenfeld und Ring, die in der Palette **Elemente»Liste & Ring** zur Verfügung stehen.

- Kapitel 14, *Bedien- und Anzeigeelemente vom Typ Array und Cluster*, beschreibt die Verwendungsweise von Arrays und Clustern. Der Zugriff auf Arrays und Cluster erfolgt über die Palette **Elemente»Array & Cluster**.
- Kapitel 15, *Bedien- und Anzeigeelemente vom Typ Graph und Diagramm*, beschreibt die Erstellung und Verwendung von Anzeigeelementen in Form von Graphen und Diagrammen mit Hilfe der Palette **Elemente»Graph**.
- Kapitel 16, *ActiveX-Bedienelemente*, erläutert die Funktionalität von ActiveX-Containern, mit denen die Kommunikation zwischen G-basierter Software und anderen Anwendungen verbessert wird.

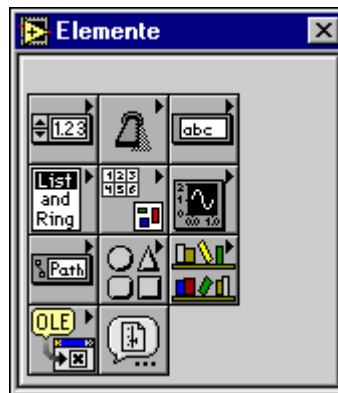
Einführung in Frontpanel-Objekte

Dieses Kapitel stellt das Frontpanel und seine zwei Komponenten vor — Bedien- und Anzeigeelemente. Es erläutert außerdem das Vorgehen beim Importieren von Grafiken aus anderen Programmen zur Verwendung in Ihren Bedienelementen.

Erstellen des Frontpanels

Bedien- und Anzeigeelemente auf dem Frontpanel sind die interaktiven Eingabe- und Ausgabeterminals des VIs. Sie verwenden Bedienelemente, um ein VI mit Daten zu versorgen, und Anzeigeelemente, um die vom VI generierten Daten anzuzeigen. Dieser Abschnitt erläutert einige Bearbeitungsoptionen, die für Bedien- und Anzeigeelemente gleichermaßen gelten.

Die folgende Abbildung zeigt die Palette **Elemente** auf dem Frontpanel.



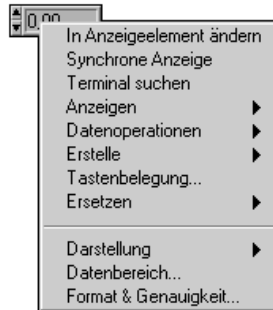
Wenn Sie mit dem Cursor auf einem beliebigen Quadrat der Palette pausieren, wird der Sammelbegriff der Bedienelemente am oberen Rand des Palettenfensters angezeigt.

Die Palette enthält die folgenden Arten von Bedienelementen (auf der Palette von links nach rechts und von oben nach unten angeordnet):

- **Numerik** — Zur Eingabe und Anzeige numerischer Mengen.
- **Boolesch** — Zur Eingabe und Anzeige der Werte TRUE/FALSE.
- **String & Tabelle** — Zur Eingabe und Anzeige von Text.
- **Liste & Ring** — Zur Anzeige und/oder Auswahl aus einer Liste von Optionen.
- **Array & Cluster** — Zur Gruppierung von Datensätzen.
- **Graph** — Zum Plotten von numerischen Daten in Diagramm- oder Graphenform.
- **Pfad & Refnum** — Zur Eingabe und Anzeige von Dateipfaden und Übermittlung von Refnum-Werten von einem VI zu einem anderen.
- **Gestaltungselemente** — Zum Hinzufügen von Grafik zur Anpassung von Frontpanels. Diese Objekte dienen nur zu Gestaltungszwecken, sie zeigen keine Daten an.
- **User Controls** — Zur automatischen Speicherung von benutzerspezifischen Bedienelementen in der Datei `user.lib`.
- **ActiveX** — Zur Erweiterung der Unterstützung von ActiveX. Diese Objekte umfassen den ActiveX-Container und die ActiveX-Variante.
- **Wählen Sie ein Element...** — Zur Auswahl eines selbsterstellten, benutzerspezifischen Bedienelements.

Optionen für Bedien- und Anzeigeelemente von Frontpanels

Wenn Sie das Popup-Menü eines Bedien- oder Anzeigeelements auf dem Frontpanel aufrufen, während Sie ein VI bearbeiten, werden Sie ein Menü sehen, das wie die folgende Abbildung aussieht. Die Optionen oberhalb der Linie gelten für alle Bedien- und Anzeigeelemente. Die Optionen unterhalb der Linie, wie z.B. **Anzeigen**, **Datenbereich...** und **Format & Genauigkeit...**, sind je nach Objekt angepaßt.



Objekte in der Palette **Elemente** werden anfangs als Bedien- oder Anzeigeelemente konfiguriert, je nachdem, wie sie üblicherweise verwendet werden. Wenn Sie z.B. einen Umschalter aus der Palette **Boolesch** wählen, erscheint dieser als Bedienelement auf dem Frontpanel, da ein Umschalter gewöhnlich ein Eingabegerät ist. Wenn Sie im umgekehrten Fall ein LED-Objekt wählen, erscheint dieses als Anzeigeelement auf dem Frontpanel, da eine LED gewöhnlich ein Ausgabegerät ist. Sie können jedoch alle Bedienelemente als Anzeigeelemente und alle Anzeigeelemente als Bedienelemente neu konfigurieren, indem Sie den Befehl **In Bedienelement ändern** oder **In Anzeigeelement ändern** aus dem Popup-Menü des Objekts wählen. Die Palette **Numerik** enthält sowohl digitale Bedienelemente als auch digitale Anzeigeelemente, da beide Arten häufig gebraucht werden. Die Palette **String & Tabelle** enthält ebenfalls sowohl ein Bedien- als auch ein Anzeigeelement für Strings.

Der Menüpunkt **Terminal suchen** im Popup-Menü von Bedien- und Anzeigeelementen hebt das Terminal des Bedien- oder Anzeigeelements im Blockdiagramm hervor. Dieser Menüpunkt ist beim Identifizieren eines bestimmten Objekts in einem sehr vollen Blockdiagramm nützlich.

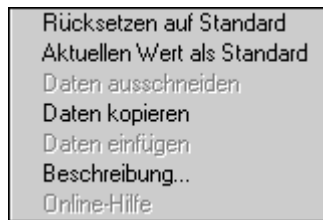
Der Menüpunkt **Erstelle»Attributknoten** erstellt einen "Attributknoten" für das Objekt. Benutzen Sie Attributknoten, um verschiedene Eigenschaften des Objekts programmatisch zu steuern.

Das Untermenü **Anzeigen** zeigt eine Liste der Bestandteile eines Bedienelements an, die wie das Namen-Label verborgen oder angezeigt werden können.

Wenn Sie ein VI bearbeiten, enthält das Popup-Menü eines Bedienelements ein Untermenü namens **Datenoperationen**. Mit den Menüpunkten dieses Menüs können Sie den Inhalt des Bedienelements ausschneiden, kopieren, oder einfügen, das Bedienelement auf seinen Standardwert setzen, den

aktuellen Wert des Bedienelements als Standardwert festlegen sowie die Beschreibung des Bedienelements lesen und ändern. (Sie können bei der Bearbeitung eines VIs Daten in ein Anzeigeelement einfügen.) Einige komplexere Bedienelemente besitzen zusätzliche Optionen; ein Array verfügt z.B. über Optionen, mit denen ein Wertebereich kopiert und das letzte Element des Arrays angezeigt werden kann.

Die folgende Abbildung zeigt das Untermenü **Datenoperationen** im Bearbeitungsmodus. Während der VI-Ausführung ist vom Popup-Menü eines Bedienelements nur dieses Untermenü verfügbar.



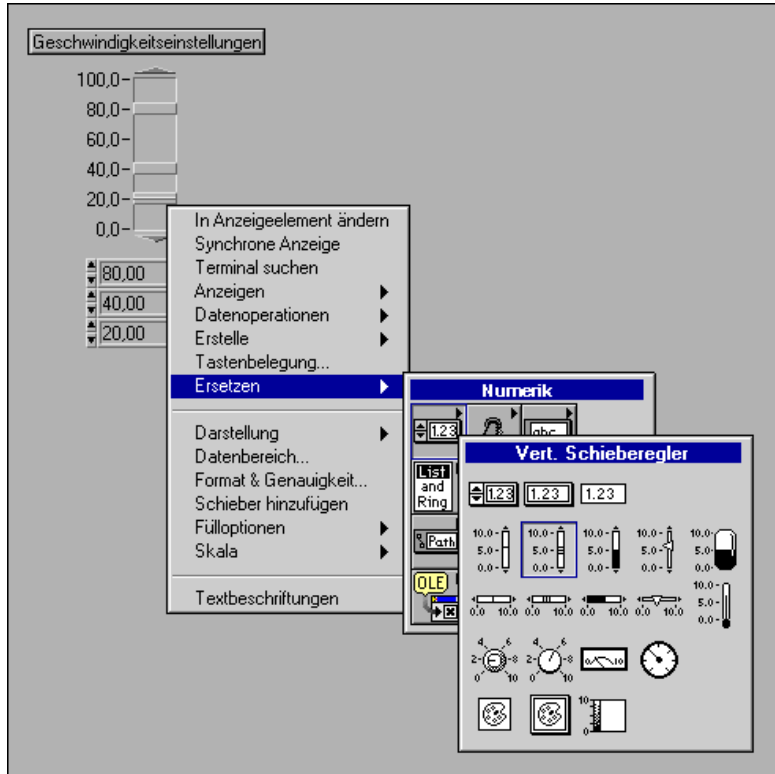
Wenn Sie während der Ausführung eines VIs das Popup-Menü eines Bedienelements aufrufen, können Sie nur den Wert des Bedienelements ändern. Die meisten Merkmale eines Bedienelements, wie z.B. der Standardwert oder die Beschreibung, können während der Ausführung des VIs nicht geändert werden.

Ersetzen von Bedienelementen

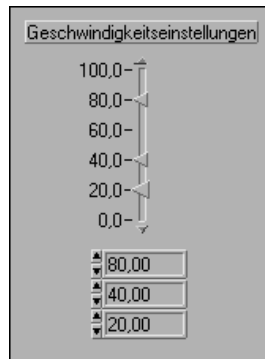
Der Menüpunkt **Ersetzen** im Popup-Menü eines Objekts zeigt die Palette **Elemente** an, aus der Sie ein Bedien- oder Anzeigeelement wählen können, das den Platz des aktuellen Objekts auf dem Frontpanel einnehmen soll.

Benutzen Sie den Menüpunkt **Ersetzen**, wenn Sie eine andere Art Bedienelement wählen wollen, aber nicht alle Bearbeitungsoptionen verlieren wollen, die Sie für das Bedienelement bis zu diesem Punkt ausgewählt haben. Durch Wählen von **Ersetzen** aus dem Popup-Menü bleiben von dem ursprünglichen Bedienelement so viele Informationen wie möglich erhalten, wie z.B. Name, Beschreibung, Standarddaten, Datenflußrichtung (Bedien- oder Anzeigeelement), Farben, Größe usw. Das neue Bedienelement behält jedoch seinen eigenen Datentyp. Vom Terminal des Bedienelements ausgehende Verbindungen oder lokale Variablen bleiben mit dem Blockdiagramm verbunden.

Je ähnlicher das Bedienelement dem zu ersetzenden ist, desto mehr ursprüngliche Merkmale können beibehalten werden. Die folgende Abbildung zeigt als Beispiel einen Schieberegler, der durch eine andere Art Schieberegler ersetzt wird.



Die nachfolgende Abbildung zeigt das Ergebnis der Ersetzung.



Der neue Schieberegler hat dieselbe Höhe, Skalierung, Beschreibung, denselben Wert, Namen usw.

Wenn Sie nun das Popup-Menü des Schiebereglers aufrufen und diesen durch einen String ersetzen wollten, blieben nur der Name, die Beschreibung und die Datenflußrichtung erhalten, da ein Schieberegler und ein String nicht vieles gemein haben.

Eine andere Methode zum Ersetzen eines Bedienelements, bei der weder das Popup-Menü **Ersetzen** verwendet wird noch irgendwelche Merkmale des alten Bedienelements erhalten bleiben, besteht darin, das Bedienelement in die Zwischenablage zu kopieren. Die Verbindungen im Blockdiagramm und VI-Anschluß bleiben allerdings bestehen. Kopieren Sie das Bedienelement in die Zwischenablage. Wählen Sie dann mit dem Positionierwerkzeug das zu ersetzende Bedienelement, und wählen Sie **Bearbeiten»Einfügen**. Das alte Bedienelement wird gelöscht und das neue an seiner Stelle eingefügt.

Optionen der Tastenbelegung für Bedienelemente

Alle Bedienelemente von Frontpanels besitzen einen Menüpunkt namens **Tastenbelegung...** Dies ermöglicht Ihnen, eine Tastenkombination (Tastatur) mit einem Bedienelement zu verknüpfen. Wenn ein Benutzer bei der Ausführung des VIs diese Tastenkombination eingibt, verfährt G, als ob der Benutzer auf das betreffende Bedienelement geklickt hätte. Das verknüpfte Bedienelement erhält den Tastenfokus. Handelt es sich um ein Textbedienelement, wird der im Bedienelement enthaltene Text hervorgehoben und kann bearbeitet werden. Handelt es sich um ein Boolesches Bedienelement, wird der Zustand der Taste umgeschaltet.



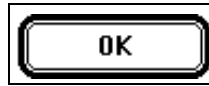
Hinweis

*Bei Anzeigeelementen ist der Menüpunkt **Tastenbelegung...** deaktiviert, da in diesem Fall ohnehin keine Eingabe möglich ist.*

Bei dem Menüpunkt **Tastenbelegung...** können Sie auch angeben, ob ein Bedienelement mitberücksichtigt werden soll, wenn der Benutzer bei der Ausführung mit der Tab-Taste von Bedienelement zu Bedienelement wechselt.

Mit dem Menüpunkt **Tastenbelegung...** können Sie Funktionstasten mit verschiedenen Tasten belegen, die das Verhalten eines Panels steuern. Sie können mit diesem Menüpunkt eine Standardtaste für VIs definieren, die sich wie ein Dialogfeld verhalten, so daß das Drücken der <Eingabe>- (**Windows und HP-UX**) oder <Return>-Taste (**Macintosh und Sun**) dem Klicken auf die Standardtaste entspricht.

Wenn Sie die <Eingabe>- oder <Return>-Taste mit einer Taste eines Dialogfeldes verknüpfen, zeichnet G diese Taste automatisch mit einer dicken Umrandung (siehe nachfolgende Abbildung).



Zwei wichtige Punkte sind hinsichtlich der <Eingabe>- oder <Return>-Taste zu beachten:

1. Wenn Sie die <Eingabe>- oder die <Return>-Taste mit einem Bedienelement verknüpfen, empfängt keines der Bedienelemente auf dem Frontpanel mehr einen Wagenrücklauf. Folglich sind alle Strings auf dem Frontpanel auf eine Zeile beschränkt.
2. Wenn Sie mit der Tabulatortaste von einem Bedienelement zum nächsten wechseln, werden Tasten besonders behandelt. Wenn Sie mit der Tabulatortaste zu einer Taste gehen und die <Eingabe>- oder <Return>-Taste drücken, wird die damit aktivierte Taste auch dann umgeschaltet, wenn die <Eingabe>- oder <Return>-Tastenkombination mit einem anderen Bedienelement belegt ist.

Eine Tastenkombination kann auf einem bestimmten Frontpanel immer nur ein Bedienelement zugewiesen werden.

Wenn Sie die Option **Tastenbelegung...** wählen, zeigt G das folgende Dialogfeld an.



Sie können den Ring links oben im Dialogfeld zur Auswahl der Taste für das betreffende Bedienelement verwenden. Sie können eine einzelne Taste oder eine Tastenkombination in Verbindung mit einer Steuertaste, wie z.B. der <Umschalt>-Taste definieren. Außer der <Umschalt>-Steuertaste können Sie die Menütaste wählen, und zwar die <Alt>-Taste (**Windows und HP-UX**), <Befehlstaste> (**Macintosh**) oder <Meta>-Taste (**Sun**). Die Optionen in diesem Ring werden in der folgenden Abbildung gezeigt.

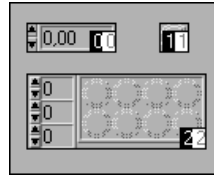
Unter "Aktuelle Zuordnungen" auf der rechten Seite des Dialogfeldes **Tastenbelegung** befindet sich eine Liste der aktuellen Tastaturbelegungen (siehe Abbildung unten).



Die Option "Panelreihenfolge"

Bedien- und Anzeigeelemente eines Frontpanels haben eine logische Reihenfolge, die sogenannte Panelreihenfolge, die von ihrer Position auf dem Frontpanel unabhängig ist. Das erste Bedien- oder Anzeigeelement, das Sie auf dem Frontpanel erstellen, ist Element 0, das zweite Element 1 usw. Wenn Sie ein Bedien- oder Anzeigeelement löschen, wird die Panelreihenfolge automatisch dementsprechend angepaßt. Wählen Sie die Panelreihenfolge mit der Option **Bearbeiten»Panelreihenfolge...** Das Aussehen des Frontpanels ändert sich, da es sich nun im

Bearbeitungsmodus für die Panelreihenfolge befindet, wie Sie in der folgenden Abbildung sehen können.



Die weißen Felder auf den Bedien- und Anzeigeelementen zeigen ihre gegenwärtige Position in der Panelreihenfolge an. Schwarze Felder zeigen die neue Position des Bedien- oder Anzeigeelements an. Durch Klicken auf ein Element mit dem Cursor für die Panelreihenfolge wird die Position des Elements in der Panelreihenfolge auf die Zahl gesetzt, die in der Symbolleiste angezeigt wird. Sie ändern diese Zahl, indem Sie eine neue Zahl eingeben. Wenn die Panelreihenfolge Ihren Wünschen entspricht, klicken Sie auf die Taste **Enter**, um den Bearbeitungsmodus für die Panelreihenfolge zu beenden. Klicken Sie auf die **“X”**-Taste, um die alte Panelreihenfolge wiederherzustellen und den Bearbeitungsmodus zu beenden.



X-Taste

Die Panelreihenfolge bestimmt die Reihenfolge, in der die Bedien- und Anzeigeelemente in den Aufzeichnungen von Datenprotokolldateien erscheinen, die bei der Protokollierung des Frontpanels erzeugt werden.

Anpassen von Steuerungen in Dialogfeldern



Dialog-Ring

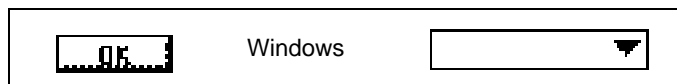
Es gibt mehrere Arten von Steuerungen in Dialogfeldern: Dialogringe (numerische Steuerung), Dialogtasten (Boolesche Steuerung) und Dialogtasten.



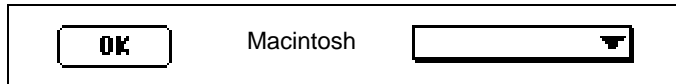
Dialogtaste

Das Aussehen der Steuerungen in Dialogfeldern ist je nach der vorliegenden Computer-Plattform verschieden. Jede Steuerung erscheint in der für die betreffende Plattform üblichen Farbe und Gestalt.

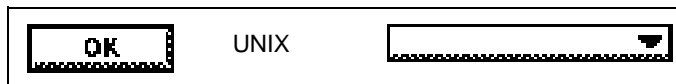
(Windows) Die Dialogtaste erscheint als dreidimensionale, graue Taste. Der Dialog-Ring erscheint als flaches, schwarzweißes Rechteck (siehe nachfolgende Abbildung).



(Macintosh) Die Dialogtaste erscheint als zweidimensionales, schwarzweißes Oval. Der Dialog-Ring erscheint als schwarzweißes Rechteck mit einem Tiefenschatten (siehe nachfolgende Abbildung).



(UNIX) Die Dialogtaste erscheint als dreidimensionale, rechteckige, graue Taste. Der Dialog-Ring erscheint als flaches, schwarzweißes Rechteck mit einem Tiefenschatten (siehe nachfolgende Abbildung). Weder die Dialogtaste noch der Dialog-Ring können mit Farben ausgestattet werden.



Da diese Steuerungen das Aussehen ändern, können Sie VIs mit Bedienelementen erstellen, deren Aussehen mit allen Computern kompatibel ist, auf denen G ausgeführt werden kann. Mit diesen Bedienelementen, den Booleschen Objekten, dem Häkchen und dem Optionsfeld, einfachen numerischen Bedienelementen, einfachen Strings und den Dialog-Schriftarten können Sie ein VI erstellen, dessen Aussehen sich an jeden beliebigen Computer anpaßt, auf dem das VI verwendet werden soll. Durch Anwendung der Optionen unter **VI Einstellungen**, mit denen Sie die Menüleiste und Rollbalken verbergen und das Fensterverhalten steuern können, sind Sie in der Lage, VIs zu erstellen, die wie Standarddialogfelder für diesen Computer aussehen.

Anpassen von Bedienelementen durch den Import von Grafiken

Sie können Grafiken von anderen Programmen importieren, um sie als Hintergrundbilder, als Objekte in Ring-Steuerungen oder als Bestandteile anderer Bedienelemente einzusetzen. (Weitere Informationen zur Verwendung von Grafiken in diesen Bedienelementen finden Sie in Kapitel 13, *Bedien- und Anzeigeelemente vom Typ Liste und Ring*, und Kapitel 24, *Benutzerspezifische Bedienelemente und Typendefinitionen*.)

Bevor Sie ein Bild verwenden können, müssen Sie es in die von BridgeVIEW oder LabVIEW gebrauchte Zwischenablage laden. Je nach der Computer-Plattform stehen hierfür eine bzw. zwei Methoden zur Verfügung, wie im folgenden beschrieben.

(Windows) Wenn Sie ein Bild direkt von einem Zeichenprogramm in die Windows-Zwischenablage kopieren können und dann zur Software der Sprache G wechseln, wird das Bild automatisch in die Zwischenablage der Sprache G importiert. Ziehen Sie eine Grafikdatei direkt vom Dateisystem aus, oder wählen Sie den Menüpunkt **Bild aus Datei importieren...** aus dem Menü **Bearbeiten**, um eine Grafikdatei in die Zwischenablage von G zu importieren. Verwenden Sie letztere Methode unter Windows 3.1, um Dateien der folgenden Formate zu importieren: BMP, CLP, EMF, GIF, JPG, LZW, PCX, PNG, TARGA, TIFF, WMF und WPG. Verwenden Sie unter Windows 95/NT Dateien der folgenden Formate: BMP, CLP, EMF, JPG, PNG und WMF.

Unter Windows 95/NT können Sie Bitmap- und Metadateien direkt auf LabVIEW- und BridgeVIEW-Panels ziehen und ablegen.

(Macintosh) Wenn Sie ein Bild direkt von einem Zeichenprogramm in die Zwischenablage kopieren können und dann zur Software der Sprache G wechseln, wird das Bild automatisch in die Zwischenablage der Sprache G importiert.

(UNIX) Benutzen Sie den Menüpunkt **Bild aus Datei importieren...** aus dem Menü **bearbeiten**, um ein Bild im Format X Window Dump (XWD), das Sie mit dem Befehl `xwd` erstellen können, oder ein JPG- oder PNG-Bild zu importieren.

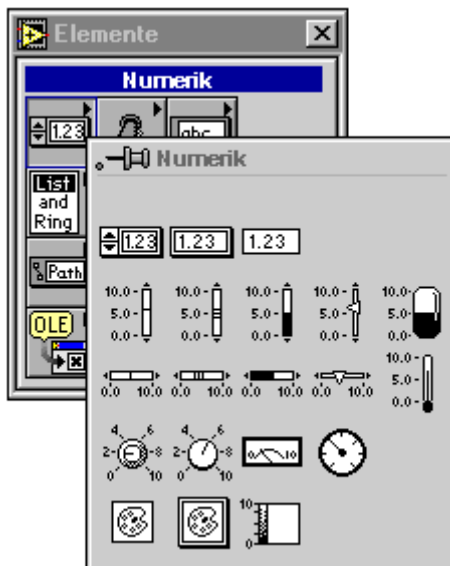
(Alle Plattformen) Befindet sich ein Bild in der Zwischenablage der Software der Sprache G, können Sie es als statisches Bild in das Frontpanel einfügen oder den Menüpunkt **Bild importieren** eines Popup-Menüs oder die Optionen unter **Bild importieren** im Bedienelement-Editor verwenden.

Ein Beispiel zum Importieren von Grafiken aus anderen Programmen finden Sie unter `examples\general\controls\custom.llb`.

Numerische Bedien- und Anzeigeelemente

Dieses Kapitel beschreibt, wie verschiedene Stilarten von numerischen Bedien- und Anzeigeelementen erstellt, bedient und konfiguriert werden.

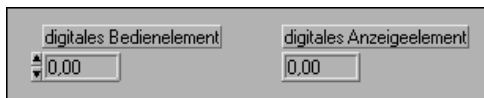
Wenn **Elemente»Numerik** von der Palette ausgewählt wird, erscheint eine neue Palette mit Bedien- und Anzeigeelementen (siehe nachstehende Abbildung).



Numerische Bedien- und Anzeigeelemente sind entweder Digital-, Schieberegler-, Dreh-, Ring-, numerierte, Farbfeld- oder Farbrampen-Elemente.

Digitale Bedien- und Anzeigeelemente

In folgender Abbildung sind digitale Bedien- und Anzeigeelemente dargestellt.



Digitalnumerik ist die einfachste Art und Weise zur Eingabe und Anzeige von numerischen Daten.



Bedienelement

Mit dem Bedienelement und einer der folgenden Methoden kann der angezeigte Wert erhöht oder verringert werden:

- In das digitale Anzeigefenster klicken und Zahlen von der Tastatur eingeben.
- Auf den Inkrement- oder Dekrement-Pfeil des digitalen Bedien- oder Anzeigeelements klicken.
- Den Cursor rechts neben die Stelle setzen, die geändert werden soll, und die Pfeil-Auf- oder Pfeil-Ab-Taste auf der Tastatur drücken.



Enter-Taste

Die **Enter**-Taste erscheint auf der Werkzeugpalette, um Sie daran zu erinnern, daß der neue Wert nur dann den alten ersetzt, wenn Sie die **<Eingabe>**-Taste auf dem numerischen Tastenfeld drücken, außerhalb des Anzeigefensters klicken oder auf die **Enter**-Taste klicken. Während der Ausführung des VIs bewirkt jeder der vorgenannten Vorgänge, daß Zwischenwerte vom Programm als Eingaben gedeutet werden. Wird z.B. ein Wert in der digitalen Anzeige auf 135 geändert, darf das VI nicht die Werte 1 und 13 erhalten, bevor der Wert 135 eingeht. Diese Regel gilt nicht für Werte, die mit den Inkrement-/Dekrement-Pfeilen geändert werden.

Numerische Bedienelemente akzeptieren nur folgende Eingaben— Dezimalstellen, ein Dezimalkomma, +, -, den Groß- oder Kleinbuchstaben e und die Begriffe Inf (Unendlich) und NaN (keine Zahl); im absoluten Zeitformat werden folgende Zeichen akzeptiert: /, : und die Groß- oder Kleinbuchstaben am und pm. Wenn die Grenze für die ausgewählte Darstellung überschritten wird, wandelt das Programm die Zahl auf das Format der natürlichen Grenze um. Wenn Sie z.B. 1234 in einen Bedienelementsatz für Byte-Integer-Werte eingeben, wandelt das Programm die Zahl in das Format 127 um. Wenn Sie nicht-numerische Werte wie aNN für NaN oder Inf für Inf eingeben, werden diese von der Anwendung ignoriert, und es wird der vorherige Wert verwendet.

Mit Inkrement-Tasten wird normalerweise die Einerstelle geändert. Durch die Erhöhung einer Stelle auf mehr als 9 oder durch Verringern der Stelle auf weniger als 0 wird die entsprechende angrenzende Stelle beeinflusst. Erhöhung und Verringerung werden automatisch wiederholt. Wenn Sie auf eine Inkrement-Taste klicken und die Maustaste gedrückt halten, wird die Anzeige wiederholt erhöht oder verringert. Wenn Sie die <Umschalt>-Taste gedrückt halten und dabei den Wert wiederholt erhöhen, erhöht sich die Größe des Inkrements um aufeinander folgende Größenordnungen, d.h. erst um Einer, dann Zehner, dann Hunderter usw. Wenn die Bereichsgrenze erreicht wird, verringert sich das Inkrement um Größenordnungen und verlangsamt sich auf normal, wenn der Wert die Grenze erreicht.

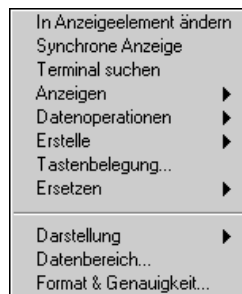
In einem digitalen Bedienelement für Zeit und Datum können individuelle Komponenten (Sekunde, Minute, Stunde, Tag, Monat, Jahr) sowie individuelle relative Zeitkomponenten erhöht und verringert werden.

Zum Erhöhen einer digitalen Anzeige um eine Stelle, die nicht die Einerstelle ist, mit dem Bedienwerkzeug den Einfügungspunkt rechts neben die Zielstelle setzen. Wenn Sie fertig sind, wird die Einerstelle wieder zu einer Inkrement-Stelle.

Zahlen werden eventuell zu groß und passen dann nicht mehr in die digitale Anzeige des Bedienelements. Der vollständige Wert kann durch Ändern der Größe des Bedienelements und durch horizontales Erhöhen der Länge betrachtet werden.

Optionen der Digitalnumerik

Die Standardwerte für Digitalnumerik können mit den entsprechenden Popup-Menüs geändert werden. Das Popup-Menü für eine Digitalnumerik wird in folgender Abbildung dargestellt.

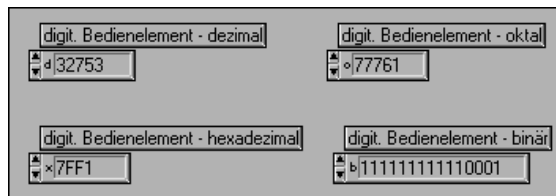


Anzeigen von Integer-Werten in anderen Radizes

Vorzeichenbehaftete oder vorzeichenlose Integerdaten können in hexadezimaler, oktaler und binärer Form sowie in dezimaler Form angezeigt werden. Zum Ändern der Form den Menüpunkt **Anzeigen» Radix** im numerischen Popup-Menü wählen. Ein \bar{d} erscheint auf dem Gehäuse der numerischen Anzeige (siehe nachstehende Abbildung).

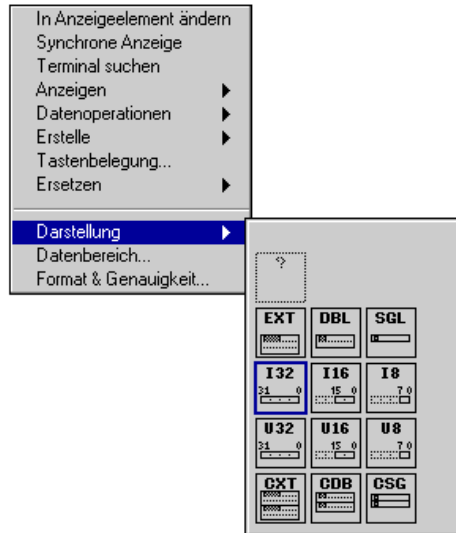


Wenn Sie auf das \bar{d} klicken, erscheint das in der vorherigen Abbildung dargestellte Menü. Die Zahl 32.753 wird in jedem Radix in folgender Abbildung angezeigt.



Ändern der Darstellung von numerischen Werten

Für ein Bedien- oder Anzeigeelement der Digitalnumerik stehen 12 Darstellungen zur Auswahl. Mit dem Menüpunkt **Darstellung** aus dem Popup-Menü Bedien- oder Anzeigeelement können Sie die Darstellung auf 32-Bit Single- (SGL), 64-Bit Double- (DBL), Extended- (EXT) Fließkommazahlen oder auf eine der sechs Integerrepräsentationen umschalten: vorzeichenbehaftetes (I8) oder vorzeichenloses (U8) Byte (8-Bit), vorzeichenbehaftetes (I16) oder vorzeichenloses (U16) Wort (16-Bit) oder vorzeichenbehaftete (I32) oder vorzeichenlose (U32) lange (32-Bit) Integer-Werte. Sie können außerdem komplexe Extended- (CXT), komplexe Double- (CDB) oder komplexe Single- (CSG) Fließkommazahlen wählen. Diese Optionen sind in folgender Abbildung dargestellt.



Bereichsoptionen für numerische Bedien- und Anzeigeelemente

Jede Darstellung verfügt über natürliche Mindest- und Höchstbereichswerte. Vorzeichenbehaftete Byte-Integer-Werte sind z.B. auf Werte von –128 bis 127 beschränkt. Fließkommazahlen verfügen über die Bereiche, die in folgender Tabelle aufgeführt sind.

Tabelle 9-1. Bereichsoptionen für Fließkommazahlen

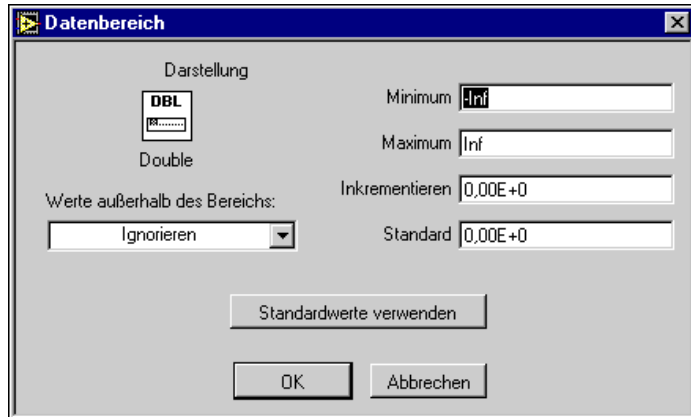
Präzision	Single	Double	Extended (plattformabhängig)
Minimale positive Zahl	1.5E–45	5.0E–24	1.9E–4951
Maximale positive Zahl	3.4E38	1.7E308	1.1E4932
Minimale negative Zahl	–1.5E–45	–5.0E–324	–1.9E–4951
Maximale negative Zahl	–3.4E38	–1.7E308	–1.1E4932



Hinweis

G kann zwar Daten im Bereich verarbeiten, der in Tabelle 9-1, Bereichsoptionen für Fließkommazahlen, aufgeführt ist; der Bereich von erweiterten Fließkommazahlen jedoch, der im Textformat dargestellt und angezeigt werden kann, ist ±9.999999999999999E999. Für HP-UX, Concurrent und Power Macintosh ist der Bereich für erweiterte Fließkommazahlen derselbe wie für Double-Zahlen.

Mit dem Menüpunkt **Datenbereich...** im Popup-Menü können Sie andere Grenzen innerhalb dieser natürlichen Beschränkungen wählen. Folgendes Dialogfeld erscheint.



Numerische Bereichsprüfung

Sie können außerdem Zwischenwerte auf gewisse Inkremente begrenzen. Wort-Integer-Werte können z.B. auf Inkremente von 10 bzw. Single Präzision-Fließkommazahlen auf Inkremente von 0,25 begrenzt werden. Wenn entweder die Grenzen oder das Inkrement geändert werden, müssen Sie festlegen, was geschehen soll, wenn ein VI oder der Bediener versucht, einen Wert außerhalb dieses Bereichs oder abweichend vom Inkrement einzustellen. Sie haben dabei folgende Optionen:

Ignorieren

Mit G werden ungültige Werte nicht geändert oder markiert. Durch Klicken auf die Inkrement- und Dekrementpfeile werden die Werte um das eingestellte Inkrement geändert, aber der Wert geht nicht über den Mindest- oder Höchstwert hinaus.

Format umwandeln

G ändert ungültige Werte automatisch auf den nächstgelegenen gültigen Wert. Wenn z.B. der Mindestwert 3, der Höchstwert 10 und das Inkrement 2 ist, sind die gültigen Werte 3, 5, 7, 9 und 10. G wandelt das Format des Wertes 0 auf 3, des Wertes 6 auf 7 und des Wertes 100 auf 10 um.

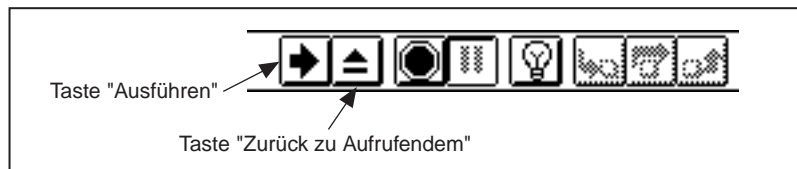
Unterdrücken

G unterdrückt die Ausführung eines VIs oder SubVIs, wenn ein Wert ungültig ist. Wenn Sie die Ausführung bei ungültigen Werten unterdrücken möchten, behält G eine Kopie aller Frontpaneldaten im Speicher für den Fall, daß Sie das Frontpanel öffnen müssen, um einen Fehler anzuzeigen.

Ein VI der höchsten Ebene mit Bedienelementen, dessen Werte ungültig sind, kann nicht ausgeführt werden. Wenn der Wert eines Bedienelements ungültig ist, bevor ein SubVI ausgeführt wird (wenn ein SubVI kurz vor der Ausführung steht), wird das VI unterdrückt. Das Frontpanel des VIs wird geöffnet (oder wird zum aktiven Fenster) und das (die) ungültigen Bedienelement(e) sind dann rot umrandet (bzw. auf einem Schwarz-Weiß-Monitor mit einer dicken schwarzen Linie umrandet). Die Werkzeugpalette auf dem Blockdiagramm eines unterdrückten SubVIs sieht wie folgende Abbildung aus.

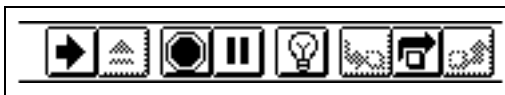


Das Bedienelement muß auf einen gültigen Wert eingestellt werden, bevor Sie fortfahren können. Wenn alle Werte für das Bedienelement gültig sind, sieht die Werkzeugpalette wie folgende Abbildung aus. Auf die Ausführen-Taste klicken, um mit der Ausführung fortzufahren.



Wenn der Wert eines Anzeige- oder Bedienelements ungültig wird, während ein SubVI ausgeführt wird, wird die Ausführung angehalten, als ob ein Breakpoint vorläge. Das Frontpanel des VIs wird geöffnet (oder wird zum aktiven Fenster). Das (die) Anzeige- und Bedienelement(e), die momentan ungültig sind, erscheinen mit roter Umrandung (oder bei Schwarz-Weiß-Monitoren mit einer dicken schwarzen Linie). Wenn ein unterbrochenes VI die Ausführung beendet und irgendwelche ungültigen

Werte auftreten, wird die Taste **Zurück zu Aufrufendem** deaktiviert (siehe nachstehende Abbildung).



Die Anzeigeelemente müssen auf gültige Werte eingestellt werden, bevor Sie zum aufrufenden VI zurückkehren können. Außerdem können die Werte des Bedienelements geändert werden, so daß gültige Ausgaben erzeugt werden, und das SubVI kann durch Klicken auf die Ausführen-Taste erneut ausgeführt werden. Wenn alle Anzeigeelement-Werte gültig sind, sieht die Werkzeugpalette wie in folgender Abbildung dargestellt aus. Auf die Taste Zurück zu Aufrufendem klicken, um mit der Ausführung fortzufahren.



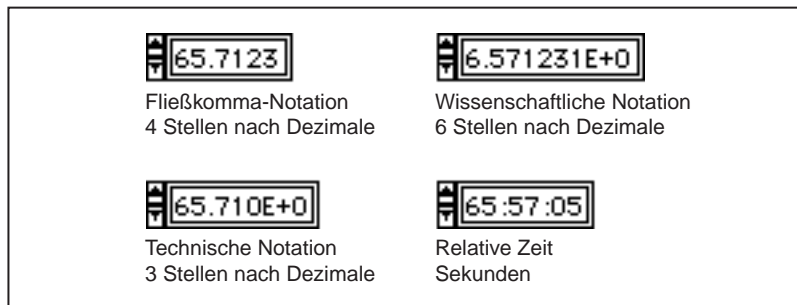
Format und Präzision von digitalen Anzeigen

Sie können das Format Ihrer digitalen Anzeigen für numerische Werte oder für Zeit und Datum auswählen. Wenn es sich um numerische Werte handelt, können Sie zwischen Fließkomma-, wissenschaftlicher, technischer Notation oder relativer Zeit in Sekunden wählen. Sie können die entsprechende Präzision von 0 bis 20 auswählen, d.h. wieviele Stellen rechts vom Dezimalkomma angezeigt werden. Die ausgewählte Präzision wirkt sich nur auf die Anzeige des Wertes aus; die interne Genauigkeit hängt noch von der Darstellung ab.

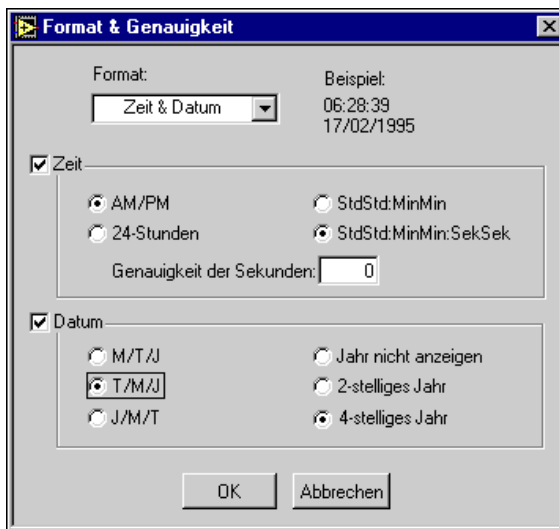
Zum Ändern dieser Parameter der digitalen Anzeige den Menüpunkt **Format & Genauigkeit...** vom Anzeige-Popup-Menü wählen. Das Dialogfeld wird im numerischen Format geöffnet (siehe nachstehende Abbildung).



Im Dialogfeld erscheint oben ein Beispiel, wenn Sie bestimmte Optionen auswählen. Beispiele der Format- und Präzisionseinstellungen auf einem digitalen Bedienelement werden in folgender Abbildung dargestellt.



Zur Formatierung absoluter Zeit und/oder Datum **Zeit & Datum** aus dem Formatting oben im Dialogfeld wählen. Das Dialogfeld ändert sich (siehe nachstehende Abbildung).



Es kann entweder die Zeit oder das Datum oder beides formatiert werden. Wenn Sie nur Zeit oder nur Datum eingeben, werden die unspezifizierten Komponenten abgeleitet. Wenn Sie die Zeit nicht eingeben, ist die angenommene Zeit 12:00 a.m. Wenn Sie das Datum nicht eingeben, wird der vorherige Wert für das Datum angenommen. Wenn Sie ein Datum eingeben, das Bedienelement jedoch nicht im Datumsformat ist, wird die Reihenfolge für Monat, Tag und Jahr aufgrund der Einstellungen im Dialogfeld "Voreinstellungen" angenommen. Wenn Sie für das Jahr nur zwei Stellen eingeben, wird folgendes angenommen: eine Zahl, die kleiner ist als 38, befindet sich im einundzwanzigsten Jahrhundert; andernfalls befindet sich die Zahl im zwanzigsten Jahrhundert.

Wenn die absolute Zeit auch als Zeit- und Datumszeichenkette angezeigt wird, so wird sie intern als Anzahl von Sekunden seit 00:00 Uhr, 1. Jan. 1904, universell koordinierte Zeit (Universal Time Coordinated - UTC), dargestellt, die zuvor als mittlere Zeit Greenwich (Greenwich Mean Time - GMT) bekannt war. Die Software überwacht diese Komponenten intern.



Hinweis

Wenn ein Bedienelement sich im absoluten Zeitformat befindet, haben Sie stets die Option, Zeit oder Datum bzw. Zeit und Datum einzugeben. Wenn kein Datum angenommen werden soll, muß die relative Zeit verwendet werden.

Beachten Sie die Beispiele oben rechts im Dialogfeld, die sich beim Auswählen von Optionen ändern.

Der gültige Bereich für Zeit und Datum ist je nach Computer-Plattform unterschiedlich, und zwar wie folgt:

(Windows) 12:00 a.m. Jan. 2, 1970 – 12:00 a.m. Feb. 4, 2106

(Windows 95/NT) 12:00 a.m. Jan. 2, 1970 – 12:00 a.m. Jan. 17, 2038

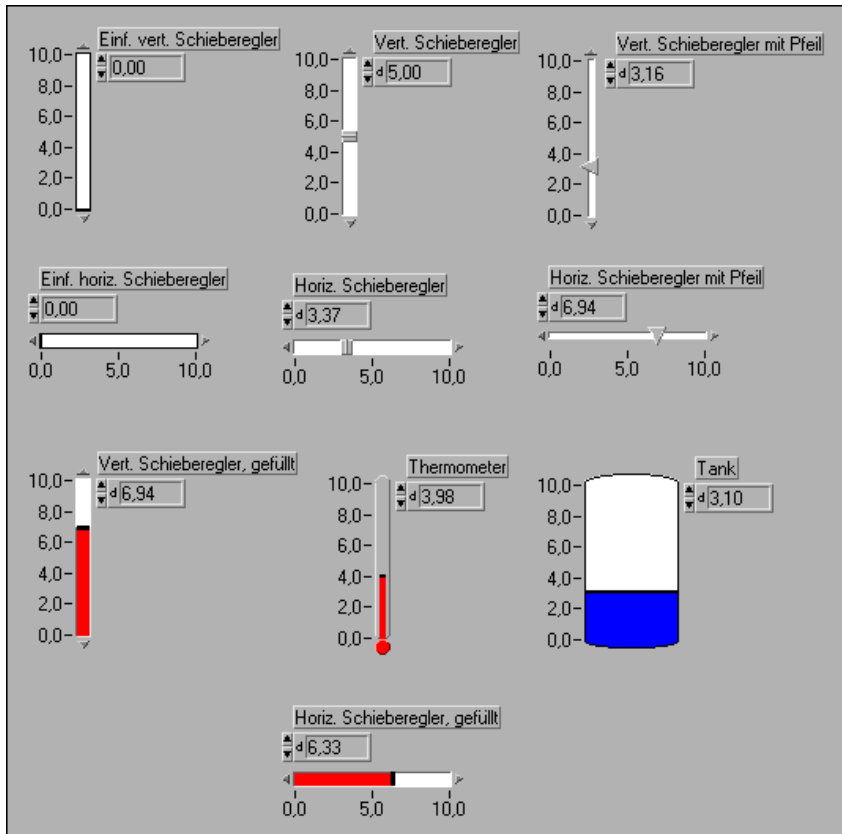
(Macintosh) 12:00 a.m. Jan. 2, 1904 – 12:00 a.m. Jan. 2, 2040

(UNIX) 12:00 a.m. Dez. 15, 1901 – 12:00 a.m. Jan. 17, 2038

Diese Bereiche sind eventuell bis zu einigen Tagen umfangreicher, und zwar je nach Ihrer Zeitzone und je nachdem, ob die Sommerzeit angewendet wird.

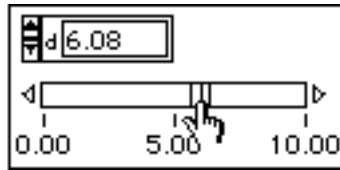
Numerische Bedien- und Anzeigeelemente mit Schieberegler

Die Bedien- und Anzeigeelemente mit Schieberegler werden in folgender Abbildung dargestellt.



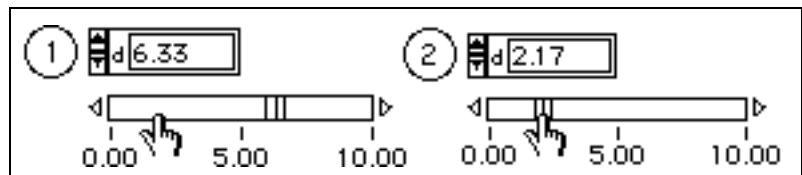
Jeder Schieberegler verfügt über eine digitale Anzeige. Mit den digitalen Anzeigen können Sie Daten in Bedienelemente mit Schieberegler eingeben, wie im Abschnitt *Digitale Bedien- und Anzeigeelemente* in diesem Kapitel erklärt wird. Mit dem Bedienwerkzeug können für Teile wie dem Schieber, dem Schieberegelergehäuse, der Skala und den Inkrement-Tasten Daten eingegeben oder Werte geändert werden. Der Schieber ist das Teil, das sich bewegt und den Wert des Bedienelements anzeigt. Das Gehäuse ist der stationäre Teil, auf dem sich der Schieber hin- und herbewegt. Die Skala gibt den Wert des Schiebers an, und die

Inkrement-Tasten sind kleine Dreiecke am einen oder anderen Ende des Gehäuses. Ein Beispiel eines Schiebers erscheint in folgender Abbildung.

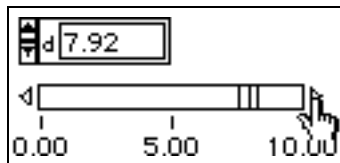


Der Schieber kann mit dem Bedienwerkzeug in eine neue Position gezogen werden. Wenn das VI während der Änderung ausgeführt wird, können Zwischenwerte an das Programm übergeben werden, je nachdem, wie häufig das VI das Bedienelement abliest.

Außerdem können Sie auf einen Punkt auf dem Gehäuse klicken, woraufhin der Schieber an dieser Position ausgerichtet wird (siehe nachstehende Abbildung). Zwischenwerte werden nicht an das Programm übergeben.

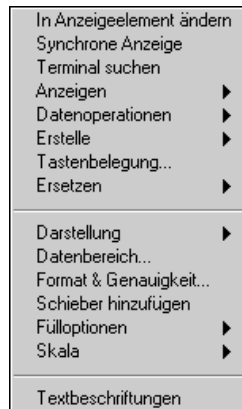


Wenn ein Schieberegler mit Inkrement-Tasten verwendet wird, können Sie auf eine Inkrement-Taste klicken, und der Schieber bewegt sich langsam in Pfeilrichtung (siehe nachstehende Abbildung). Die <Umschalt>-Taste gedrückt halten, um den Schieber schneller zu bewegen. Zwischenwerte können an das Programm übergeben werden.



Schieberegler verfügen wie die Digitalnumerik über die Optionen **Darstellung**, **Datenbereich...** und **Format & Genauigkeit** in den entsprechenden Popup-Menüs. Diese Optionen funktionieren auf gleiche Art und Weise wie bei digitalen Anzeigen, mit der Ausnahme, daß

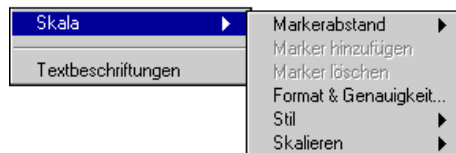
Schieberegler keine komplexen Zahlen darstellen können. Darüber hinaus verfügen Schieberegler über andere Optionen. In der folgenden Abbildung wird ein Popup-Menü für einen Schieberegler dargestellt.



Der Menüpunkt **Anzeigen»Digitale Anzeige** des Popup-Menüs regelt die Sichtbarkeit der digitalen Anzeige des Schiebereglers.

Schiebereglerskala

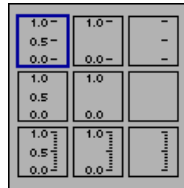
Die Menüpunkte des Skala-Untermenüs gelten nur für die Skala. Das Popup-Menü für die Skala wird in folgender Abbildung dargestellt.



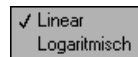
Der Menüpunkt **Markerabstand** wird im Abschnitt [Achsenmarker](#) in diesem Kapitel beschrieben.

Der Menüpunkt **Format & Genauigkeit** funktioniert so wie im Abschnitt [Format und Präzision von digitalen Anzeigen](#) in diesem Kapitel beschrieben.

Der Menüpunkt **Stil** zeigt die Palette, die in folgender Abbildung dargestellt ist. Sie können eine Skala ohne Tick-Marker bzw. ohne Skalawerte anzeigen, oder Sie können die Skala ganz verbergen.



Der Menüpunkt **Skalierung** zeigt die Option linearer oder logarithmischer Skalabstand (siehe nachstehende Abbildung).



Wenn auf Logarithmischen Abstand gewechselt wird und die untere Skalagrenze kleiner als oder gleich 0 ist, wird die Grenze automatisch zu einer positiven Zahl, und andere Marker werden entsprechend mit neuen Werten versehen. Denken Sie daran, daß Skalooptionen einschließlich der Skalierungsfunktionen von den Datenbereichswerten des Schiebereglers unabhängig sind; diese werden mit der Popup-Option **Datenbereich** geändert. Wenn Ihre Daten auf logarithmische Werte begrenzt werden sollen, muß der Datenbereich so abgeändert werden, daß Werte kleiner oder gleich Null eliminiert werden.

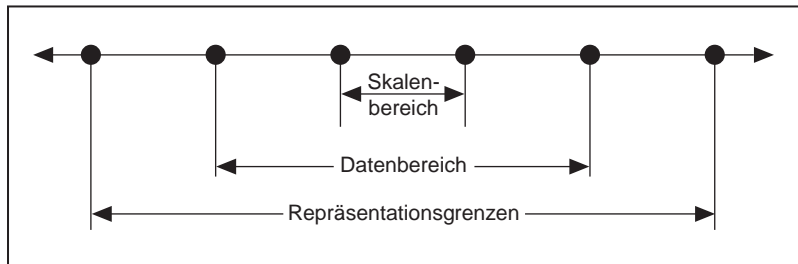
Achsenmarker

Die Skala eines numerischen Bedien- oder Anzeigeelements verfügt über zwei oder mehr Marker, bei denen es sich um Labels handelt, die den mit der Markerposition assoziierten Wert anzeigen.

Ändern der Skalagrenzen

Die beiden äußeren Marker sind die Skalagrenzen. Sie müssen nicht mit den Bereichsgrenzen übereinstimmen, können jedoch Subsets des Bereichs des Bedien- oder Anzeigeelement sein. Der Standardbereich eines Bedien- oder Anzeigeelements mit vorzeichenbehaftetem 16-Bit-Integer-Wert

z.B. ist -32.768 bis 32.767 . Sie können den Datenbereich jedoch so einstellen, daß er von -1.000 bis 1.000 reicht, und anschließend die Skalengrenzen auf 0 und 500 einstellen.



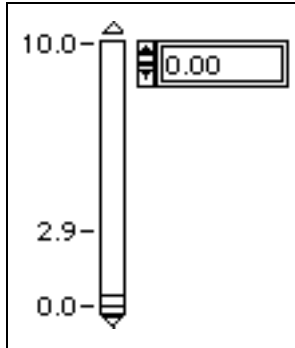
Das Minimum, Maximum und Inkrement einer Skala kann mit fünf Methoden interaktiv entweder mit dem Bedienwerkzeug oder dem Beschriftungswerkzeug geändert werden. Wenn eine Skala programmatisch geändert werden soll, den Attributknoten verwenden. Weitere Informationen über das programmatische Ändern einer Skala mit Hilfe des Attributknotens finden Sie in Kapitel 22, [Attributknoten](#).

- Wenn Sie einen neuen Maximalwert in die Anzeige eingeben, bleibt der Minimalwert gleich, und G berechnet das Inkrement automatisch neu.
- Wenn Sie einen neuen Minimalwert in die Anzeige eingeben, bleibt der Maximalwert gleich, und G berechnet das Inkrement automatisch neu.
- Wenn Sie den aktuellen Minimalwert in die Anzeige für den Maximalwert eingeben, kehrt G die Skala um, so daß der vorherige Minimalwert jetzt zum Maximalwert wird und der vorherige Maximalwert jetzt zum Minimalwert wird. Außerdem berechnet G die Inkremente.
- Wenn Sie irgendwelche Werte in den Zwischenmarker eingeben, wird das Inkrement zu diesem Wert minus dem Minimum.
- Wenn Sie die Größe des Schiebereglers ändern, wird das Inkrement so angepaßt, daß sich die Marker nicht überlappen.

Auswählen der nicht-uniformen Achsenmarker-Verteilung

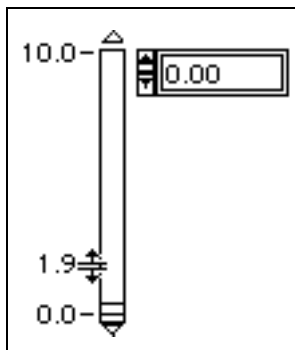
Standardmäßig verfügen Achsenmarker über gleichmäßige Abstände. Wenn gewünscht, kann genau angegeben werden, wo sich innere Marker für eine Skala befinden sollen. Dies ist dann nützlich, wenn einige spezifische Punkte auf dem Schieberegler markiert werden sollen (wie z.B. ein Sollwert oder ein Schwellwert). Wenn eine nicht-uniforme Marker-

Verteilung gewünscht wird, den Menüpunkt **Skala»Markerabstand» Arbiträr** im Popup-Menü der Skala auswählen. Dann können Sie das Popup-Menü des Schiebereglers aufrufen und **Skala»Marker hinzufügen** auswählen oder das Popup-Menü des Markers aufrufen und **Marker hinzufügen** auswählen. Ein Marker erscheint an einer arbiträren Stelle (siehe nachstehende Abbildung).



Zum Löschen eines arbiträren Markers das Popup-Menü des Schiebereglers aufrufen und **Skala»Marker löschen** auswählen oder das Popup-Menü des Markers aufrufen und **Marker löschen** auswählen.

Wenn ein Marker erstellt wird, kann eine Zahl in den Marker eingegeben werden; die Position paßt sich automatisch an die Zahl an. Außerdem können Marker verschoben werden, indem man Ticks, die kleinen Linien neben den Datenpunkten auf der Skala, an eine andere Stelle zieht. Dazu das Bedienwerkzeug über die Ticks fahren. Der Cursor ändert sich und zeigt an, wie die Ticks verschoben werden können (siehe nachstehende Abbildung).



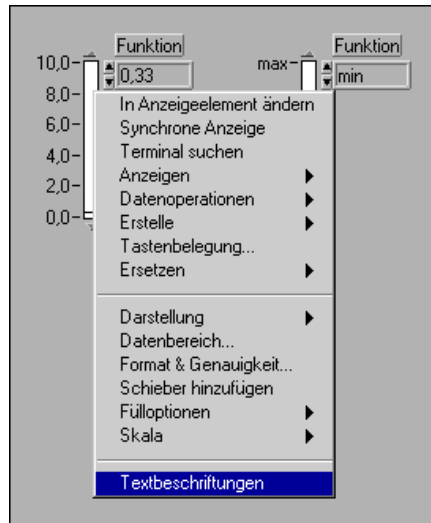
Im Uniform-Modus wird durch das Ziehen eines Ticks die Marker-Verteilung geändert. Im Arbiträr-Modus wird durch das Ziehen eines Ticks nur der aktuelle Marker verschoben, ohne daß andere Marker davon berührt werden. Durch das Drücken der <Strg>- (**Windows**); <Option>- (**Macintosh**); <Meta> (**Sun**) oder <Alt>- (**HP-UX**) Taste während des Ziehvorgangs wird ein neuer Marker erstellt. (Wenn die Ticks, wie im Dialogfeld **Skala»Stil** eingestellt, verborgen sind, können sie nicht gezogen werden.)

Die arbiträren Marker wirken sich nur auf die inneren Marker aus, nicht auf die zwei Endmarker. Wenn im aktuellen Bereich keine arbiträren Marker sichtbar sind, kehrt die Skala zeitweilig zu uniformen Markern zurück.

Textskala

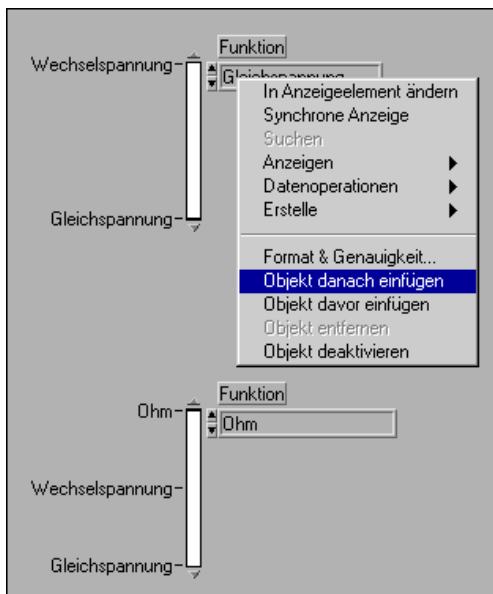
Auf numerischen Skalen können auch Text-Labels verwendet werden. Labels sind deshalb nützlich, weil Strings mit Integer-Werten assoziiert werden. Diese Konfiguration ist zur Auswahl von sich gegenseitig ausschließenden Optionen hilfreich. Zur Verwendung dieser Option **Textbeschriftungen** aus dem Popup-Menü des Schiebereglers auswählen. Das Schieberegler-Bedienelement erscheint mit den Standardtext-Labels `min` und `max`, und Sie können sofort mit der Eingabe der gewünschten Labels beginnen. Drücken Sie <Umschalttaste-Eingabetaste> (**Windows** und **HP-UX**) oder <Umschalttaste-Return> (**Macintosh** und **Sun**) nach jedem Label, um ein neues Label zu erstellen, und die <Eingabe>-Taste auf dem numerischen Tastenfeld, nachdem Sie das letzte Label fertiggestellt haben.

Mit dem Beschriftungswerkzeug können Sie die Labels `min` und `max` in der Textanzeige oder auf dem Schieberegler selbst bearbeiten.



Sie können das Popup-Menü der Textanzeige aufrufen und **Anzeigen» Digitale Anzeige** auswählen, um festzustellen, welche numerischen Werte mit den erstellten Text-Labels assoziiert sind. Diese Werte beginnen stets mit Null (bei senkrechten Schieberegler unten und bei waagrecchten Schieberegler links oder rechts) und erhöhen sich für jedes Text-Label um Eins.

Mit dem Menüpunkt **Objekt danach einfügen** oder **Objekt davor einfügen** aus dem Popup-Menü der Textanzeige können Sie neue Labels erstellen (siehe nachstehende Abbildung).



Außerdem können Sie die <Umschalt-Eingabetaste> drücken, um zu einem neuen Menüpunkt vorzurücken, wenn Sie vorhandene Punkte bearbeiten.

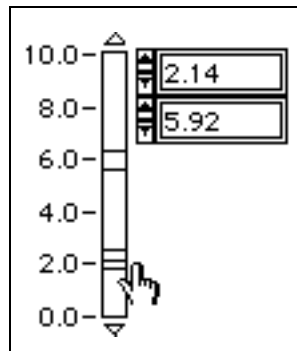
Ausgefüllte Schieberegler und Schieberegler mit Mehrfachwerten

Die Palette **Numerisch** enthält vier Bedienelemente, die so konfiguriert sind, daß sie vom minimalen bis zum Schieberwert ausgefüllt sind. Diese Bedienelemente umfassen einen senkrechten Schieberegler, einen waagrechten Schieberegler, den Tank und den Thermometer. Mit dem Menüpunkt **Fülloptionen** im Popup-Menü können alle Schieberegler in ausgefüllte Schieberegler abgeändert werden, und ausgefüllte Schieberegler können in normale Schieberegler geändert werden. Normalerweise haben Sie drei Optionen (siehe nachstehende Abbildung).

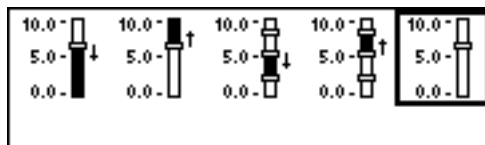
Diese sind: vom minimalen Wert zur Schiebereglerstelle ausfüllen, vom maximalen Wert zu Schiebereglerstelle ausfüllen oder keine Ausfüllung.



Außerdem kann mehr als ein Wert auf demselben Schieberegler angezeigt werden. Dazu den Menüpunkt **Schieber hinzufügen** aus dem Popup-Menü auswählen. Ein neuer Schieber erscheint zusammen mit einer neuen digitalen Anzeige (siehe nachstehende Abbildung).



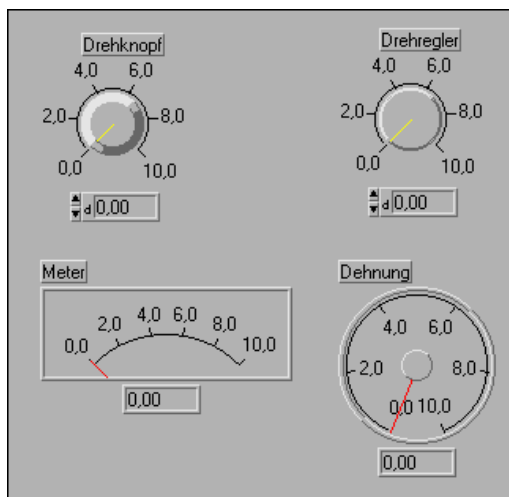
Danach erscheinen zwei weitere Optionen in der Palette **Fülloptionen**: **Füllen zu oberem Wert** und **Füllen zu unterem Wert**. Diese Optionen gelten nur für den aktiven Schieber. Alle fünf Optionen sind in folgender Abbildung dargestellt.



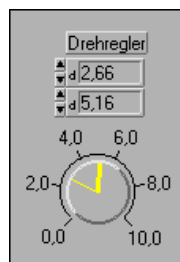
Ein Beispiel von Bedienelementen und Anzeigeelementen mit Schieberegler finden Sie in `examples\general\controls\alarmsld.11b`.

Runde numerische Bedien- und Anzeigeelemente

Die runden numerischen Bedien- und Anzeigeelemente sind in folgender Abbildung dargestellt.



Drehobjekte verfügen zum größten Teil über dieselben Optionen wie Schieberegler. Die Schieber (oder Nadeln) in Drehobjekten lassen sich drehen und nicht verschieben, aber sie werden auf dieselbe Weise bedient wie Schieberegler.



Mit runden Bedienelementen kann wie mit linearen Bedienelementen mehr als ein Wert angezeigt werden (siehe nachstehende Abbildung). Neue Werte können hinzugefügt werden, indem **Zeiger hinzufügen** aus dem Popup-Menü ausgewählt wird, und zwar in derselben Weise wie Schieberegler neue Werte hinzugefügt werden.

Wenn das Positionierwerkzeug zur Skala bewegt wird, verwandelt sich das Werkzeug in einen Drehcursor. Wenn man auf die äußeren Grenzen der

Skala klickt und sie zieht, wird der Bogen geändert, den die Skala umfaßt. Wenn Sie die inneren Marker der Skala anklicken und ziehen, drehen Sie den Bogen. Er verfügt noch über denselben Bereich, hat jedoch unterschiedliche Start- und Endwinkel. Dieses Verfahren wird in folgender Abbildung dargestellt. Die <Umschalt>-Taste gedrückt halten, um den Bogen auf einen 45-Grad-Winkel auszurichten.

Den Cursor auf die Drehknopfskala plazieren. Sein Aussehen ändert sich, und er gleicht einem Hufeisen mit Pfeilspitzen an beiden Enden.



Durch Ziehen der äußeren Marker ändert sich die Größe des Skalabogens.



Durch Ziehen der inneren Marker wird der Bogen gedreht.

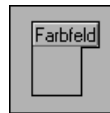


Tick-Markierungen der runden Skalen können genauso wie lineare Skalen mit dem Bedienwerkzeug gezogen werden. Ticks können außerdem in gleicher Weise wie bei linearen Skalen ungleichmäßig verteilt werden. Weitere Informationen finden Sie weiter vorne im Abschnitt [Auswählen der nicht-uniformen Achsenmarker-Verteilung](#).

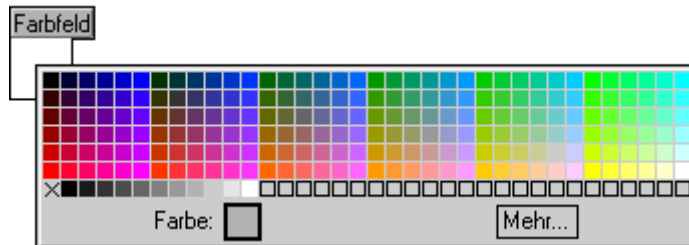
Weitere Informationen über Ring-Bedien- und Anzeigeelemente finden Sie in Kapitel 13, *Bedien- und Anzeigeelemente vom Typ Liste und Ring*.

Farbfeld

Das Farbfeld zeigt eine Farbe an, die einem spezifizierten Wert entspricht. Der Farbwert wird als hexadezimale Zahl in der Form `RRGGBB` ausgedrückt, wobei die beiden ersten Zahlen den roten Farbwert, die beiden nächsten Farben den grünen Farbwert und die beiden letzten den blauen Farbwert steuern. Ein Beispiel wird in folgender Abbildung dargestellt.



Die Farbe des Farbfeldes wird durch Klicken mit dem Bedien- oder Farbwerkzeug eingestellt, so daß eine **Farbe**-Palette erscheint (siehe nachstehende Abbildung). Durch Loslassen der Maustaste auf der gewünschten Farbe wird diese Farbe gewählt.

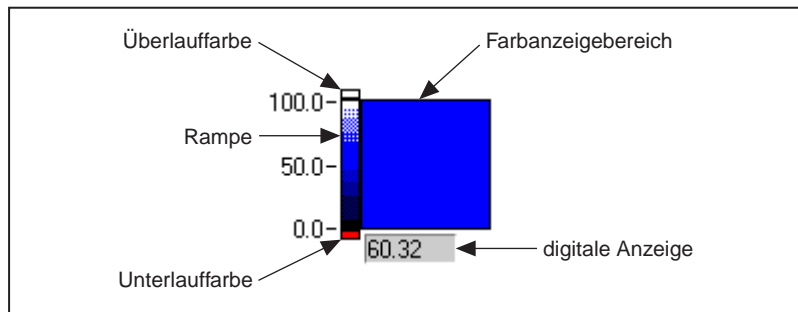


Farbfelder werden normalerweise auf dem Frontpanel als Anzeigeelemente verwendet. Die einfachste Methode zur Einstellung der Farbe eines Farbfeldes ist die Verwendung der Farbfeldkonstante aus der Palette **Funktionen»Numerisch»Zusätzliche Num Konstanten** im Blockdiagramm-Fenster. Die Farbfeldkonstante mit dem Farbfeldanschluß auf dem Blockdiagramm verbinden. Durch Klicken auf die Farbfeldkonstante mit dem Bedien- oder Farbwerkzeug wird dieselbe **Farbe**-Palette angezeigt, die zur Einstellung des Farbfeldes verwendet wird. Wenn das Farbfeld-Anzeigeelement auf dem Frontpanel auf verschiedene Farben eingestellt werden soll, die verschiedene Zustände anzeigen, können Sie innerhalb einer Fallstruktur eine Reihe von Farbfeldkonstanten einsetzen.

Das kleine T in der unteren linken Ecke der **Farbe**-Palette bedeutet “transparent”.

Farbrampe

Die Farbrampe verwendet Farben zur Anzeige des entsprechenden numerischen Wertes. Es kann eine Farbskala konfiguriert werden, die aus mindestens zwei arbiträren Markern besteht, wobei jeder dieser Marker aus einem numerischen Wert und der diesem Wert entsprechenden Farbe besteht. Wenn der Eingabewert sich ändert, ändert sich die angezeigte Farbe in die dem Wert entsprechende Farbe. Ein Beispiel ist in folgender Abbildung dargestellt.



Diese Paare können mit Hilfe der arbiträren Marker der Farbskala erstellt werden. Jeder arbiträre Marker spezifiziert einen Wert und verfügt über eine damit assoziierte Farbe. Die mit einem Marker assoziierte Farbe kann geändert werden, indem Sie das Popup-Menü des Markers aufrufen, nicht das der Rampe. Sie können Marker hinzufügen, die Werte ändern und die Farben nach Belieben einstellen.

Mit dem Menüpunkt **Farben interpolieren** des Popup-Menüs der Farbrampe kann gesteuert werden, ob Zwischenschattierungen für Werte zwischen den angegebenen Markern angezeigt werden sollen. Wenn **Farben interpolieren** deaktiviert ist, wird die Farbe auf die Farbe der größten Ebene abzüglich oder gleich dem aktuellen Wert eingestellt.

Das Farbarray ist stets nach Markerwerten sortiert. Die Skala auf der Rampe entspricht dem größten und kleinsten Wert im Array. Wenn das Minimum oder Maximum der Skala sich ändert, werden die Farbarray-Ebenen automatisch zwischen den neuen Werten verteilt.

Die Farbskala verfügt über eine Reihe von Optionen, die angezeigt oder verborgen werden können. Diese Optionen umfassen Einheiten-Label, digitale Anzeige und Rampe.

Die Rampenkomponente dieses Bedienelements hat oben und unten an der Skala eine zusätzliche Farbe. Aus diesen Farben kann eine zur Anzeige von Überlauf oder Unterlauf ausgewählt werden. Der Bedienelementwert liegt über dem Maximum der Farbskala oder unter dem Minimum der Farbskala. Mit dem Bedienwerkzeug auf diese Bereich klicken und die Überlauf- und Unterlauffarben aus der **Farbe**-Palette auswählen.

Die Farbrampe sowie das Farbfeld werden zur Anzeige eines numerischen Wertes als Farbe verwendet. Mit der Farbrampe können Sie einen beliebigen Bereich von Zahlen einem beliebigen Bereich von Farben zuweisen. Das Farbfeld zeigt jedoch nur die Farbe an, die von den roten, grünen und blauen Komponenten des numerischen Wertes spezifiziert wurde; ein gegebener Wert wird stets in derselben Farbe abgebildet.



Hinweis

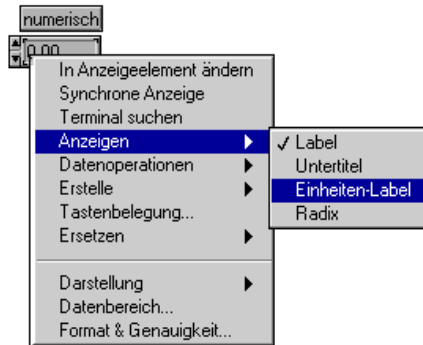
Die Farbrampe kann über so viele Farben verfügen, wie Ihr Monitor verarbeiten kann; es können jedoch nicht mehr Farben verwendet werden, als auf Ihrem Monitor zur Verfügung stehen.

Die Farbskala kann dazu verwendet werden, Farbtabelle für die Bedienelemente Intensitätsgraph und Intensitätsdiagramm anzuzeigen. Weitere Informationen finden Sie in Kapitel 15, *Bedien- und Anzeigeelemente vom Typ Graph und Diagramm*.

Einheitentyp

Numerische Bedienelemente können mit physischen Einheiten wie z.B. Meter oder Kilometer/Sekunde assoziiert werden. Alle numerischen Bedienelemente mit einer assoziierten Einheit sind auf einen Datentyp mit Fließkomma beschränkt.

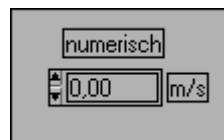
Einheiten für ein Bedienelement werden in einem separaten, jedoch eingebundenen Label, dem sogenannten Einheiten-Label angezeigt. Dieses Label kann aufgerufen werden, indem der Menüpunkt vom Popup-Menü ausgewählt wird (siehe nachstehende Abbildung).



Wenn das Einheiten-Label angezeigt wird, können Sie mit Hilfe von standardmäßigen Abkürzungen wie *m* für Meter, *ft* für Fuß, *s* für Sekunden und so weiter eine Einheit eingeben.

Wenn Sie sich nicht sicher sind, welche Einheiten akzeptabel sind, eine einfache Einheit wie z.B. *m* eingeben und anschließend das Popup-Menü des Einheiten-Labels aufrufen und **Einheit...**wählen. Es erscheint ein Dialogfeld mit Informationen über die verfügbaren Einheiten. Dieses Dialogfeld kann zum Ersetzen der ersten Einheit durch eine passendere Einheit verwendet werden.

Folgendes numerisches Bedienelement ist so eingestellt, daß die Einheiten als Meter pro Sekunde erscheinen.



In Tabelle 9-1 bis Tabelle 9-6 sind alle Einheiten aufgeführt, die bei der Einheitenfunktion verwendet werden können.

Tabelle 9-2. Basiseinheiten

Größe	Einheit	Abkürzung
Flächenwinkel	Radian	rad
Raumwinkel	Steradian	sr
Zeit	Sekunde	s
Länge	Meter	m
Masse	Gramm	g
Elektrischer Strom	Ampere	A
Thermodynamische Temperatur	Kelvin	K
Stoffmenge	Mol	mol
Leuchtkraft	Candela	cd

Tabelle 9-3. Abgeleitete Einheiten mit Sondernamen

Größe	Einheit	Abkürzung
Frequenz	Hertz	Hz
Kraft	Newton	N
Druck	Pascal	Pa
Energie	Joule	J
Leistung	Watt	W
Elektrische Ladung	Coulomb	C
Elektrisches Potential	Volt	V
Kapazität	Farad	F
Elektrischer Widerstand	Ohm	Ohm
Leitfähigkeit	Siemens	S
Magnetischer Fluß	Weber	Wb
Magnetische Flußdichte	Tesla	T
Induktivität	Henry	H
Lichtstrom	Lumen	lm

Tabelle 9-3. Abgeleitete Einheiten mit Sondernamen (Fortsetzung)

Größe	Einheit	Abkürzung
Beleuchtungsstärke	Lux	lx
Celsius-Temperatur	Grad Celsius	°C
Aktivität	Becquerel	Bq
Energiedosis	Gray	Gy
Äquivalentdosis	Sievert	Sv

Tabelle 9-4. Zusätzliche Einheiten, die mit SI-Einheiten verwendet werden

Größe	Einheit	Abkürzung
Zeit	Minute	min
Zeit	Stunde	h
Zeit	Tag	d
Flächenwinkel	Grad	deg
Flächenwinkel	Minute	'
Flächenwinkel	Sekunde	"
Volumen	Liter	l
Masse	metrische Tonne	t
Fläche	Hektar	ha
Energie	Elektronenvolt	eV
Masse	vereinheitlichte atomare Masseneinheit	u

Tabelle 9-5. CGS-Einheiten

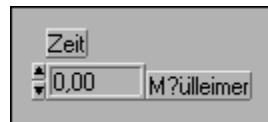
Größe	Einheit	Abkürzung
Fläche	barn	b
Kraft	Dyn	dyn
Energie	Erg	erg
Druck	Bar	bar

Tabelle 9-6. Sonstige Einheiten

Größe	Einheit	Abkürzung
Fahrenheit-Temperatur	Grad Fahrenheit	degF
Celsius-Temperatur-differenz	Celsius-Grad	Cdeg
Fahrenheit-Temperatur-differenz	Fahrenheit-Grad	Fdeg
Länge	Fuß	ft
Länge	Zoll	in
Länge	Meile	mi
Fläche	Acre	acre
Druck	Atmosphäre	atm
Energie	Kalorie	cal
Energie	British Thermal Unit (britische Kalorie)	Btu

Eingeben von Einheiten

Wenn versucht wird, eine ungültige Einheit in das Einheiten-Label einzugeben, markiert G die ungültige Einheit, indem ein ? in das Label plaziert wird (siehe nachstehende Abbildung).

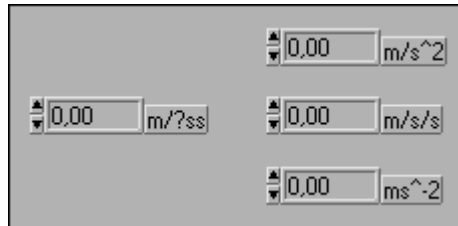


Beachten Sie, daß Einheiten mit den richtigen Abkürzungen eingegeben werden müssen, da sie ansonsten von G markiert werden (siehe nachstehende Abbildung).



Außerdem muß beachtet werden, daß keine doppeldeutigen Einheiten eingegeben werden dürfen. Aus diesem Grund ist m/ss in folgender Abbildung markiert, da es nicht klar ist, ob die Abkürzung Meter pro Sekunde zum Quadrat oder (Meter/Sekunde) * Sekunde bedeutet. Zum

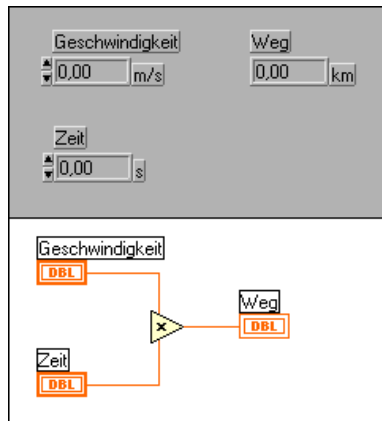
Vermeiden von Doppeldeutigkeit müssen die Einheiten anders eingegeben werden (siehe nachstehende Abbildung).



Einheiten können erst dann für ein Diagramm oder einen Graphen ausgewählt werden, wenn sie mit einem Objekt verbunden werden, das über eine damit assoziierte Einheit verfügt. Dann können Sie die Einheit anzeigen lassen, die das Diagramm oder der Graph durch die Verbindung erworben hat. Diese Einheit kann geändert werden, jedoch nur in eine kompatible Einheit, die dasselbe physische Phänomen mißt. Wenn ein Eingang zu einem Diagramm z.B. mit der Einheit m_i versehen ist, können Sie die Einheit im Diagramm auf ft , in oder m , jedoch nicht auf N , Hz , min , $acre$ oder A abändern.

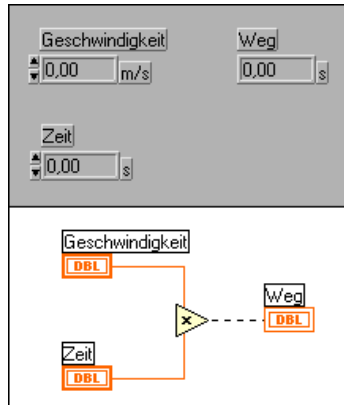
Überprüfen von Einheiten und strikten Typen

Eine Verbindung, die an eine Quelle angeschlossen ist und mit der eine Einheit assoziiert ist, kann nur an ein Ziel mit einer kompatiblen Einheit angeschlossen werden (siehe nachstehende Abbildung).



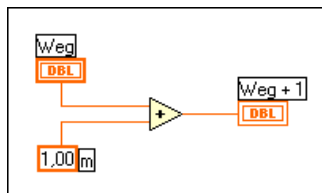
Im Fall der vorherigen Abbildung wird die Abstandsanzeige automatisch so skaliert, daß Kilometer anstatt Meter angezeigt werden, da diese Einheit für die Anzeige spezifiziert war.

Signale mit inkompatiblen Einheiten können nicht verbunden werden (siehe nachstehende Abbildung). Wenn **Fehler anzeigen** aus dem Popup-Menü für die unterbrochene Verbindung ausgewählt wird, zeigt das Fenster “Fehler anzeigen” den Fehler folgendermaßen an: Verbindung: Einheitenkonflikt.



Einige Funktionen sind in bezug auf Einheiten doppeldeutig und können nicht mit Signalen verwendet werden, denen Einheiten zugeordnet wurden. Die Funktion “Eins hinzufügen” z.B. ist doppeldeutig in bezug auf Einheiten. Wenn Sie Abstandseinheiten verwenden, kann “Eins hinzufügen” nicht feststellen, ob ein Meter, ein Kilometer oder ein Fuß hinzugefügt werden soll. Aufgrund dieser Doppeldeutigkeit können die Funktion “Eins hinzufügen” und ähnliche Funktionen nicht mit Daten verwendet werden, die über assoziierte Einheiten verfügen.

Eine Möglichkeit, dieses Problem zu umgehen, ist die Verwendung einer numerischen Konstante mit der richtigen Einheit und die Additionsfunktion auf dem Blockdiagramm zur Erstellung einer benutzerdefinierten Funktion “Eins hinzufügen”.



Hinweis

In Formelknoten können keine Einheiten verwendet werden.

Polymorphe Einheiten

Wenn Sie ein VI erstellen möchten, das den mittleren Quadratwurzelwert einer Kurvenform berechnet, muß die mit der Kurvenform assoziierte Einheit definiert werden. Für Spannungs-Kurvenformen, Strom-Kurvenformen, Temperatur-Kurvenformen und so weiter ist jeweils ein separates VI erforderlich. Damit ein VI dieselbe Berechnung unabhängig von der durch die Eingänge erhaltenen Einheiten durchführt, verfügt die Software der G-Sprache über die Möglichkeit der Verwendung von polymorphen Einheiten.

Eine polymorphe Einheit wird erstellt, indem Sie $\$x$ eingeben, wobei x eine Zahl darstellt (z.B. $\$1$). Dieser Ausdruck ist sozusagen der Platzhalter für die tatsächliche Einheit. Wenn das VI aufgerufen wird, ersetzt G die Einheiten, die für alle Vorkommen von $\$x$ in diesem VI übergeben werden. An den Platzhalter $\$x$ wird stets eine in Basiseinheiten ausgedrückte Einheit übergeben, z.B. Meter, Meter/Sek, nicht Fuß, Zoll usw.

Eine polymorphe Einheit wird als eindeutige Einheit behandelt. Sie kann nicht in eine andere Einheit konvertiert und im ganzen Diagramm verbreitet werden, wie es bei anderen Einheiten der Fall ist. Wenn sie an ein Anzeigeelement angeschlossen ist, das ebenfalls über die Abkürzung $\$1$ verfügt, stimmen die Einheiten überein, und das VI kann kompiliert werden.

$\$1$ kann wie andere Einheiten in Kombinationen verwendet werden. Wenn der Eingang z.B. mit 3 Sekunden multipliziert und dann an ein Anzeigeelement angeschlossen wird, muß das Anzeigeelement $\$1$ s Einheiten sein. Wenn das Anzeigeelement über unterschiedliche Einheiten verfügt, zeigt das Blockdiagramm eine ungültige Verbindung an.

Wenn mehr als eine polymorphe Einheit benötigt wird, können die Abkürzungen $\$2$, $\$3$ und so weiter verwendet werden.

Ein Aufruf an ein SubVI mit polymorphen Einheiten berechnet Ausgangseinheiten basierend auf den Einheiten, die von den dazugehörigen Eingängen empfangen wurden. Angenommen, es wird z.B. ein VI erstellt, das über zwei Eingänge mit den polymorphen Einheiten $\$1$ und $\$2$ verfügt und das einen Ausgang in der Form $\$1 \$2/s$ erstellt. Wenn das SubVI so verbunden wird, daß der Eingang mit m/s an den Eingang mit $\$1$ an der Eingang mit kg an den Eingang mit $\$2$ angeschlossen wird, wird die Ausgangseinheit als $kg m/s^2$ berechnet.

Angenommen, ein anderes VI verfügt über zwei Eingänge mit $\$1$ und $\$1/s$ und berechnet einen Ausgang mit $\$1^2$. Wenn dieses VI so verbunden

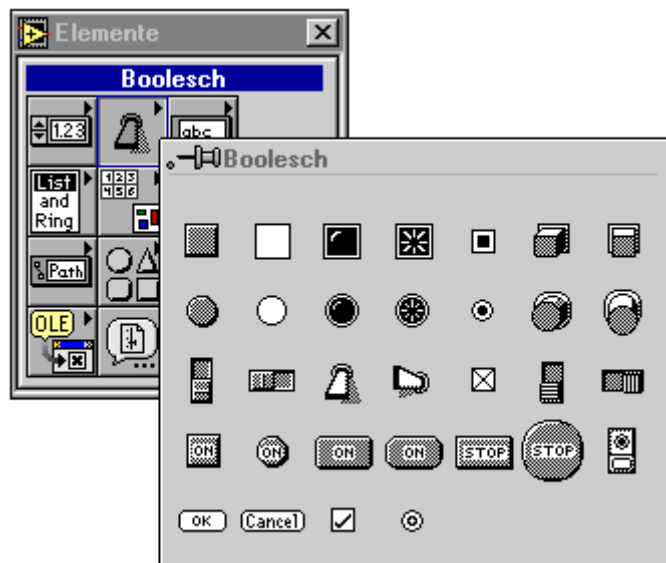
wird, daß der Eingang mit m/s an den Eingang mit $\$1$ und der Eingang mit m/s^2 an den Eingang mit $\$1/s$ angeschlossen wird, wird die Ausgangseinheit als m^2/s^2 berechnet. Wenn dieses VI so verbunden wird, daß der Eingang mit m an den Eingang mit $\$1$ und der Eingang mit kg an den Eingang mit $\$1/s$ angeschlossen wird, wird der SubVI-Aufruf jedoch unterbrochen. Einer der Eingänge gilt als Einheitenkonflikt, und der Ausgang wird (wenn möglich) vom anderen Eingang aus berechnet. Ein polymorphes VI kann über ein polymorphes SubVI verfügen, da die jeweiligen Einheiten separat gehalten werden.

Boolesche Bedien- und Anzeigeelemente

In diesem Kapitel wird beschrieben, wie Boolesche Bedien- und Anzeigeelemente erstellt, bedient und konfiguriert werden.

Erstellen und Bedienen von Booleschen Bedien- und Anzeigeelementen

Ein Boolesches Bedien- oder Anzeigeelement hat zwei Werte—TRUE oder FALSE. Boolesche Bedien- und Anzeigeelemente können aus der Palette **Elemente**»**Boolesch** ausgewählt werden (siehe nachstehende Abbildung).



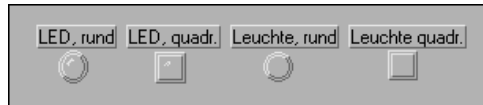
Einige Boolesche Bedienelemente, die mechanische Drucktasten, Kippschalter und Schiebeschalter simulieren, werden in folgender Abbildung dargestellt.



Hinweis

Die Dialogtaste, das Dialoghäkchen und das Dialogoptionsfeld sehen auf jeder Plattform anders aus. Informationen über Dialog-Bedienelemente auf verschiedenen Plattformen finden Sie im Abschnitt [Anpassen von Steuerungen in Dialogfeldern](#) in Kapitel 8, [Einführung in Frontpanel-Objekte](#).

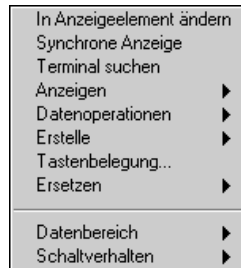
Einige Boolesche Anzeigeelemente, die LEDs und Lampen simulieren, sind in folgender Abbildung dargestellt.



Durch Klicken eines Booleschen Bedienelements mit dem Bedienwerkzeug wird dieses zwischen TRUE (ein) und FALSE (aus) hin- und hergeschaltet. Im Ausführungsmodus hat das Klicken eines Anzeigeelements keine Auswirkung, da Anzeigeelemente nur für Ausgaben gedacht sind. Im Bearbeitungsmodus können Sie sowohl Bedien- als auch Anzeigeelemente bedienen.

Konfigurieren von Booleschen Bedien- und Anzeigeelementen

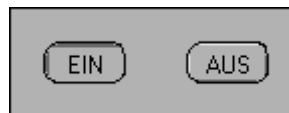
Jedes Boolesche Bedien- oder Anzeigeelement verfügt im dazugehörigen Popup-Menü über verschiedene Optionen. Das Popup-Menü für ein Boolesches Objekt wird in folgender Abbildung dargestellt.



Die Menüpunkte über der Linie im Popup-Menü sind allen Bedien- und Anzeigeelementen gemeinsam und werden im Abschnitt [Optionen für Bedien- und Anzeigeelemente von Frontpanels](#) von Kapitel 8, [Einführung in Frontpanel-Objekte](#) beschrieben. Die Menüpunkte **Datenbereich** und **Schaltverhalten** werden weiter hinten in diesem Kapitel beschrieben.

Ändern der Booleschen Labels

Mit mehreren Bedienelementen in der Palette **Boolesch** wird Text angezeigt; diese Bedienelemente werden als *beschriftete Boolesche Bedienelemente* bezeichnet. Zu Anfang wird im TRUE-Zustand das Wort ON (Ein) und im FALSE-Zustand das Wort OFF (Aus) angezeigt (siehe nachstehende Abbildung). Wenn Sie dann mit dem Bedienwerkzeug auf die Taste klicken, wird das Bedienelement in den anderen Zustand umgeschaltet. Im Bearbeitungsmodus kann der Text in beiden Zuständen mit dem Beschriftungswerkzeug geändert werden, z.B. von EIN auf AUS.



Standardmäßig ist der Text auf der Taste zentriert. Wenn der Text verschoben werden soll, **Text loslassen** aus dem Popup-Menü auswählen.

Zu diesem Zeitpunkt können Sie den Text mit dem Positionierwerkzeug neu positionieren oder den Menüpunkt **Text in Mitte fixiert** auswählen. Den Booleschen Text und dann den Fontring auf der Werkzeugpalette auswählen, um die Schriftart, Größe und Farbe des Textes zu ändern. Außerdem können Sie den Booleschen Text von beiden Zuständen verbergen, indem Sie den Menüpunkt **Boolescher Text** auf dem Untermenü **Anzeigen** umschalten.

Die Booleschen Bedienelemente, die nicht als beschriftete Boolesche Bedienelemente erscheinen, sind standardmäßig unbeschriftet. Sie können jedoch den Menüpunkt **Boolescher Text** aus dem Untermenü **Anzeigen** aufrufen, um sie zu beschriften. Sie können den Booleschen Text in diese Booleschen Bedienelemente übertragen. Dieser Text ist nicht in der Mitte der Taste zentriert, es sei denn, Sie wählen den Menüpunkt **Text in Mitte fixiert** aus dem Popup-Menü.

Wenn Sie die Schriftart des Bezeichnungs-Labels oder des Booleschen Textes ändern möchten, ohne daß sich die Änderung auf beide auswirkt, müssen Sie das zu ändernde Element mit dem Beschriftungswerkzeug auswählen und dann die Menüpunkte im Fontring verwenden, um die gewünschten Änderungen vorzunehmen.

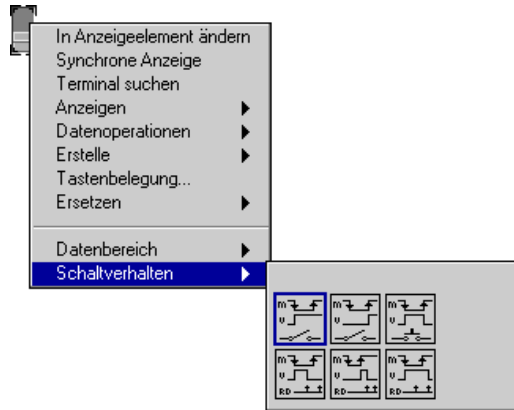
Boolesche Datenbereichsprüfung

Es ist eventuell erforderlich, Fehler in Booleschen Werten aufzufinden. Wenn ein Boolescher Wert erwartungsgemäß stets TRUE ist, den Menüpunkt **Datenbereich»Unterdrücken wenn FALSE** aus dem Untermenü des Booleschen Popup-Menüs auswählen. Wenn der Boolesche Wert erwartungsgemäß stets FALSE ist, den Menüpunkt **Datenbereich»Unterdrücken wenn TRUE** auswählen, um Fehler festzustellen. Wird der unerwartete Boolesche Wert angetroffen, dann wird das VI unterdrückt, bevor oder nachdem es ausgeführt wird, wie für das Element **Unterdrücken** des numerischen Bereichs beschrieben.

Konfigurieren des Schaltverhaltens der Booleschen Bedienelemente

Boolesche Bedienelemente verfügen über sechs Arten von Schaltverhalten. Das für Ihre Anwendung angemessene Schaltverhalten kann aus dem Popup-Menü der Palette **Schaltverhalten** ausgewählt werden, das in folgender Abbildung dargestellt ist. Bei diesen Palettensymbolen stellt M die Bewegung der Maustaste dar, wenn das Bedienelement bedient wird;

V stellt den Ausgangswert des Bedienelements dar, und RD stellt den Zeitpunkt dar, zu dem das VI das Bedienelement liest.



Mit dem Schaltverhalten **Schaltet wenn gedrückt** wird der Wert des Bedienelements jedesmal dann geändert, wenn Sie mit dem Bedienwerkzeug darauf klicken, ähnlich wie bei einem Lichtschalter. Das Schaltverhalten wird nicht davon beeinflusst, wie häufig das VI das Bedienelement liest.



Mit dem Schaltverhalten **Schaltet wenn losgelassen** wird der Wert des Bedienelements nur dann geändert, wenn Sie die Maustaste nach einem Mausklick innerhalb der graphischen Grenze des Bedienelements loslassen. Das Schaltverhalten wird nicht davon beeinflusst, wie häufig das VI das Bedienelement abliest.



Mit dem Schaltverhalten **Schaltet bis losgelassen** wird der Wert des Bedienelement geändert, wenn Sie darauf klicken, und der neue Wert wird beibehalten, bis Sie die Maustaste loslassen. Zu diesem Zeitpunkt kehrt das Bedienelement wieder zum ursprünglichen Wert zurück, ähnlich wie die Funktion eines Türsummers. Das Schaltverhalten wird nicht davon beeinflusst, wie häufig das VI das Bedienelement abliest.



Beim Schaltverhalten **Latch während gedrückt** wird der Wert des Bedienelements geändert, wenn Sie darauf klicken, und der neue Wert wird beibehalten, bis das VI ihn einmal abgelesen hat. Zu diesem Zeitpunkt kehrt das Bedienelement wieder zum Standardwert zurück, ganz unabhängig davon, ob die Maustaste gedrückt wird. Dieses Verhalten ähnelt einem Trennschalter und ist zum Stoppen von While-Schleifen nützlich; außerdem kann man damit ein VI so einstellen, daß es jedesmal, wenn das Bedienelement eingestellt wird, ein bestimmter Vorgang nur einmal durchgeführt wird.



Mit dem Schaltverhalten **Latch wenn losgelassen** wird der Wert des Bedienelements nur dann geändert, wenn Sie die Maustaste innerhalb der graphischen Grenzen des Bedienelements loslassen. Nachdem Ihr VI es einmal gelesen hat, kehrt das Bedienelement zum alten Wert zurück. Dieses Verhalten funktioniert in derselben Weise wie Dialogfeldtasten und Systemtasten.



Mit dem Schaltverhalten **Latch bis zum Loslassen** wird der Wert des Bedienelements geändert, wenn Sie darauf klicken; er wird solange beibehalten, bis er vom VI einmal gelesen wurde oder bis die Maustaste losgelassen wird, je nachdem, welcher Vorgang zuletzt stattfindet.

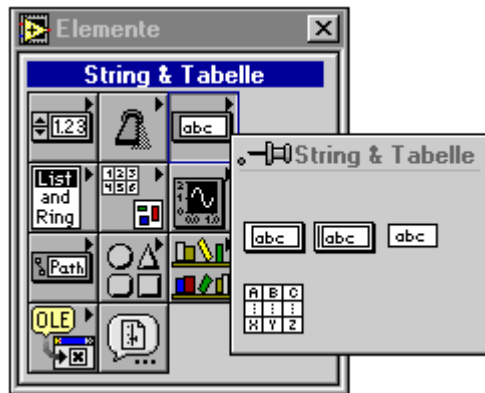
Ein Beispiel mit Booleschen Bedien- und Anzeigeelementen finden Sie in `examples\general\controls\booleans.llb`.

Anpassen von Booleschen Bedien- und Anzeigeelementen mit importierten Bildern

Sie können Ihren eigenen Booleschen Stil entwerfen, indem Sie für ein beliebiges Boolesches Bedien- oder Anzeigeelement Bilder für den TRUE- und FALSE-Zustand importieren. Dieses Verfahren wird im Abschnitt [Verwendung des Bedienelement-Editors](#) von Kapitel 24, [Benutzerspezifische Bedienelemente und Typendefinitionen](#) erklärt.

Bedien- und Anzeigeelemente vom Typ String

In diesem Kapitel wird die Verwendung von Bedien- und Anzeigeelementen vom Typ String und die Tabelle beschrieben. Die Objekte können über die Palette **Elemente»String & Tabelle** aufgerufen werden (siehe nachstehende Abbildung).



Verwendung von Bedien- und Anzeigeelementen vom Typ String

In folgender Abbildung sind ein String-Bedienelement und ein String-Anzeigeelement dargestellt.



Der Text kann mit dem Bedien- oder Beschriftungswerkzeug in das String-Bedienelement eingegeben oder geändert werden. Standardmäßig wird neuer oder geänderter Text erst dann an das Diagramm übergeben, wenn die <Eingabe>-Taste auf dem numerischen Tastenfeld gedrückt oder auf die **Enter**-Taste in der Palette **Werkzeuge** oder außerhalb des Bedienelements geklickt wird, um die Bearbeitungssitzung zu beenden. Wenn Sie jedoch den Menüpunkt **Wert während der Eingabe aktualisieren** aus dem Popup-Menü auswählen, ändert sich der Wert des Bedienelements mit jedem Zeichen. Durch das Drücken der <Return>-Taste auf der alphanumerischen Tastatur wird einen Zeilenrücklauf eingegeben.

Wenn der Text am rechten Rand der Stringanzeige angelangt ist, erfolgt ein Umbruch zur nächsten Zeile, und zwar auf einem natürlichen Begrenzungszeichen wie einem Leerschritt oder einem Tabulatorzeichen.



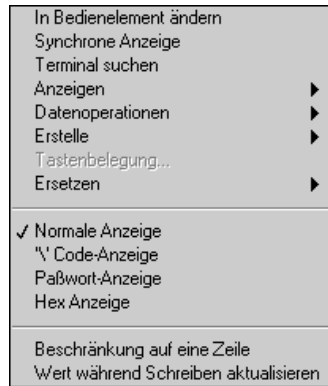
Hinweis

Bei der Ausführung eines VIs wird die <Tab>-Taste dazu verwendet, um zum nächsten Bedienelement vorzurücken. Bei der Bearbeitung eines VIs werden durch das Drücken der <Tab>-Taste Werkzeuge geändert. Zur Eingabe eines Tabulatorzeichens in einen String den Menüpunkt '\t' Code-Anzeige aus dem String-Popup-Menü auswählen und \t eingeben. Zur Eingabe eines Zeilenvorschubs in einen String die <Eingabe>- (Windows und HP-UX) oder <Return>- (Macintosh und Sun) Taste auf der alphanumerischen Tastatur drücken oder den Menüpunkt '\n' Code-Anzeige aus dem String-Popup-Menü auswählen und \n eingeben. Zur Eingabe eines Zeilenrücklaufs in einen String den Menüpunkt '\r' Code-Anzeige aus dem String-Popup-Menü auswählen und \r eingeben. Eine vollständige Liste von '\'-Codes finden Sie in Tabelle 11-1, G '\'-Codes im Abschnitt [Anzeigetypen](#) in diesem Kapitel.

Informationen über das Manipulieren von Strings finden Sie in der **Online-Referenz** unter dem **Thema»Funktion und VI-Referenz»String-Funktionen**. Beispiele dazu finden Sie in `examples\general\strings.llb`.

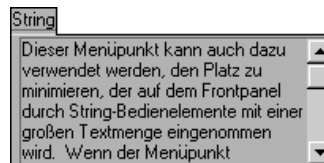
Menüpunkte für Bedien- und Anzeigeelemente vom Typ String

Strings verfügen über besondere Funktionen, die über das String-Popup-Menü ausgerufen werden können (siehe nachstehende Abbildung).



Rollbalken-Menüpunkte

Der Menüpunkt **Rollbalken** im String-Popup-Untermenü **Anzeigen** ist deaktiviert, bis der String so vergrößert wird, daß er auf einen Rollbalken paßt. Wenn dieser Menüpunkt ausgewählt wird, erscheint ein senkrechter Rollbalken auf dem String-Bedien- oder Anzeigeelement (siehe nachstehende Abbildung) so daß Text angezeigt werden kann, der im String-Bedienelement nicht sichtbar ist. Dieser Menüpunkt kann auch dazu verwendet werden, den Platz zu minimieren, der auf dem Frontpanel durch String-Bedienelemente mit einer großen Textmenge eingenommen wird. Wenn der Menüpunkt abgeblendet ist, muß der String höher geschoben werden, damit der Rollbalken bei Auswahl dieses Punkts paßt.



Anzeigetypen

Mit Menüpunkten im mittleren Abschnitt des String-Popup-Menüs können Sie auswählen, ob ein String Daten im Normalmodus anzeigt, umgekehrte Schrägstrich-Codes anstelle von nicht druckbaren Zeichen anzeigt, im Paßwortmodus anzeigt (unter Verwendung des Symbols “*” anstelle einzelner Zeichen) oder Hexadezimalzeichen anzeigt.

Normale Anzeige

Der Menüpunkt **Normale Anzeige** zeigt alle Zeichen so an, wie sie eingegeben wurden (mit der Ausnahme von Zeichen, die mit nicht anzeigbaren Sondertasten wie die <Tab>- oder <Esc>-Taste erstellt wurden).

Umgekehrter Schrägstrich- (‘\’) Code-Anzeige

Durch die Auswahl von ‘\’ **Code-Anzeige** aus dem String-Popup-Menü wird das Softwareprogramm angewiesen, Zeichen, die unmittelbar auf einen umgekehrten Schrägstrich folgen (\), als Code für nicht anzeigbare Zeichen zu interpretieren. Folgende Tabelle zeigt, wie G diese Codes interpretiert.

Tabelle 11-1. G \-Codes

Code	G-Interpretation
\00 - \xFF	Hex-Wert eines 8-Bit-Zeichens; muß in Großbuchstaben eingegeben werden
\b	Rückwärtsschritt (ASCII BS, entspricht \08)
\f	Formularvorschub (ASCII FF, entspricht \0C)
\n	Zeilenvorschub (ASCII LF, entspricht \0A)
\r	Zeilenrücklauf (ASCII CR, entspricht \0D)
\t	Tab (ASCII HT, entspricht \09)
\s	Leerschritt (entspricht \20)
\\	Umgekehrter Schrägstrich (ASCII \, entspricht \5C)

Für Hexadezimalzeichen Großbuchstaben und für Sonderzeichen, wie z.B. Formularvorschub und Rückwärtsschritt, müssen Kleinbuchstaben verwendet werden. Somit interpretiert G die Sequenz \bFare als Hex BF, gefolgt von dem Wort *are*, wobei G \bFare und \bfare als einen

Rückwärtsschritt gefolgt von den Worten *Fare* und *fare* interpretiert. In der Sequenz `\Bfare` ist `\B` nicht der Rückwärtsschritt-Code, und `\Bf` ist kein gültiger Hex-Code. Wenn in einem solchen Fall auf einen umgekehrten Schrägstrich nur ein Teil eines gültigen Hex-Zeichens folgt, nimmt G an, daß eine *0* auf den umgekehrten Schrägstrich folgt; also interpretiert G `\B` als *hex 0B*. Immer dann, wenn auf einen umgekehrten Schrägstrich kein gültiges Hex-Zeichen folgt, erkennt G das Zeichen “Umgekehrter Schrägstrich” nicht.

Einige nicht anzeigbare Zeichen, wie Zeilenrücklauf, können von der Tastatur in ein String-Bedienelement eingegeben werden, unabhängig davon, ob ‘\’ **Code-Anzeige** ausgewählt wird. Falls Sie jedoch den umgekehrten Schrägstrichmodus aktivieren, wenn das Anzeigefenster Text enthält, zeichnet G die Anzeige neu, um die umgekehrte Schrägstrichdarstellung eventuell vorhandener, nicht anzeigbaren Zeichen sowie das `\` Zeichen selbst anzuzeigen.

Angenommen, der Modus ist auf Normalanzeige eingestellt, und es wird folgender String eingegeben.

```
left
\righ\3F
```

Wenn der Modus aktiviert wird, erscheint folgender String, da der Zeilenrücklauf nach `left` und die darauffolgenden umgekehrten Schrägstrichzeichen in umgekehrter Schrägstrichform als `\n\\` angezeigt werden.

```
left\n\\right\\3F
```

Angenommen, es wird ‘\’ **Code-Anzeige** ausgewählt und folgender String eingegeben.

```
left
\righ\3F
```

Wenn der Modus deaktiviert wird, erscheint folgender String, da G `\r` ursprünglich als Zeilenrücklauf interpretiert hat und jetzt dieses Zeichen druckt. `\3F` ist die Sonderdarstellung des Fragezeichens (?) und wird folgendermaßen gedruckt.

```
left
ight?
```


Wenn jetzt ‘\’ **Code-Anzeige** erneut gewählt wird, erscheint folgender String.

```
left\n\righ?
```

Anzeigeelemente weisen das gleiche Verhalten auf.

In diesen Beispielen ändern sich die Daten im String nicht vom einen Modus zum anderen. Nur die angezeigte Darstellung bestimmter Zeichen ist anders.

Der umgekehrte Schrägstrichmodus ist nützlich für das Debugging von Programmen und zum Senden von nicht druckbaren Zeichen an Instrumente, serielle Schnittstellen und andere Geräte.

Anzeige des Paßwortes

Mit dem Menüpunkt **Paßwort-Anzeige** zeigt das String-Bedienelement das symbol “*” für jedes Zeichen an, das dort eingegeben wird. Wenn Sie jedoch die Daten des Strings vom Blockdiagramm ablesen, lesen Sie die vom Anwender eingegebenen tatsächlichen Daten. Wenn versucht wird, Daten vom Bedienelement zu kopieren, werden nur die *-Zeichen kopiert.

Hex-Anzeige

Mit dem Menüpunkt **Hex Anzeige** kann der String als Hex-Zeichen anstelle von alphanumerischen Zeichen angezeigt werden. Wie der Menüpunkt ‘\’ **Code-Anzeige** ist **Hex Anzeige** nützlich für das Debugging und bei der Kommunikation von Instrumenten.

Auf eine Zeile begrenzen

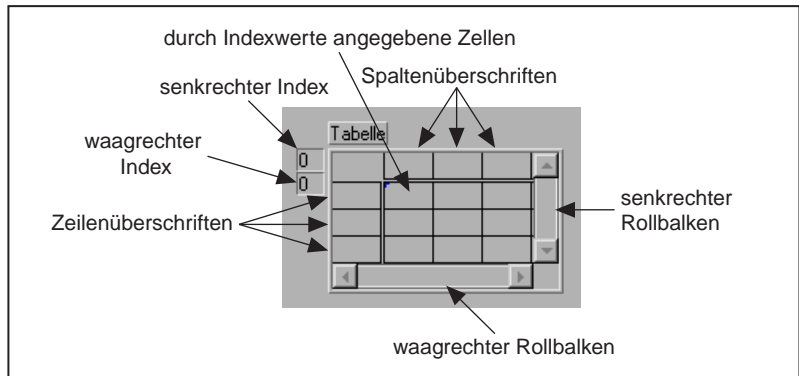
Der Menüpunkt **Auf eine Zeile begrenzen** hindert Sie daran, beim Schreiben in einen String ein Zeilenrücklaufzeichen einzugeben.

Wert während der Eingabe aktualisieren

Durch die Auswahl des Menüpunkts **Wert während der Eingabe aktualisieren** kann sich der Wert eines Bedienelements ändern, während Zeichen eingegeben werden, so daß mit der Änderung des Wertes eines Bedienelements nicht gewartet werden muß, bis der Anwender auf die Eingabetaste klickt oder die Bearbeitungssitzung anderweitig beendet. Dieses Verhalten ist nützlich zur Prüfung der Richtigkeit der Eingabe, zur Begrenzung der Eingabe oder um dem Anwender Feedback zur Verfügung zu stellen. Es ist z.B. möglich, ein String-Bedienelement zu erstellen, bei dem die Eingabe auf alphanumerische Zeichen beschränkt ist.

Tabellen

Eine Tabelle ist ein 2D-Array von Strings. Folgende Abbildung ist ein Beispiel einer Tabelle, in der alle Funktionen angezeigt sind.

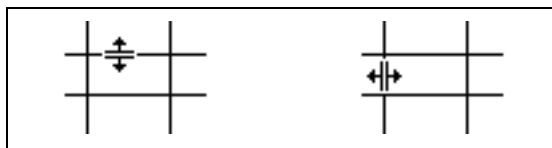


Ein Tabelle verfügt über Zeilen- und Spaltenüberschriften, die durch einen dünnen, offenen Rand von den Daten abgetrennt sind. Wenn die Tabelle auf das Frontpanel platziert wird, können Überschriften eingegeben werden, die mit dem Bedien- oder Beschriftungswerkzeug auch geändert werden können. Die Überschriften können mit dem Attributknoten aktualisiert oder gelesen werden.

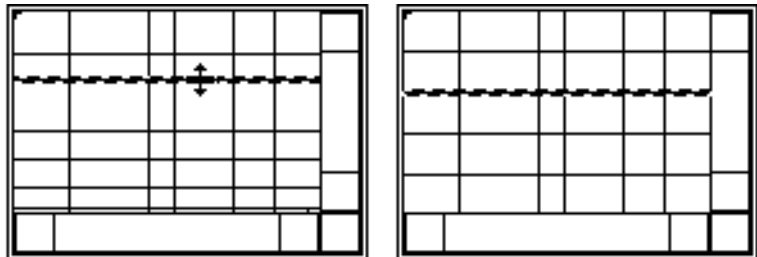
Die Indexanzeige gibt an, welche Zelle in der oberen linken Ecke der Tabelle sichtbar ist. Diese Indizes können wie auf einem Array bedient werden.

Ändern der Größe von Tabellen, Zeilen und Spalten

Die Größe der Tabelle kann von jeder Ecke aus mit dem Größenänderungswerkzeug geändert werden. Die Größe einzelner Zeilen und Spalten in einer Tabelle kann geändert werden, indem eine der Randlinien mit dem Positionierwerkzeug gezogen wird. Wenn das Werkzeug zur Platzierung einer Linie richtig platziert wird, erscheint ein Zieh-Cursor (siehe nachstehende Abbildung). Auf eine Linie klicken und sie ziehen, um die Größe der Zeile oder Spalte zu ändern.



Beim Ziehen einer Randlinie können Sie die <Umschalt>-Taste drücken, um mehrere Zeilen oder Spalten so zu ändern, daß sie alle dieselbe Größe haben. Wenn die Zeile oder Spalte, dessen Größe sie ändern, sich innerhalb der ausgewählten Tabelle befindet (in blau oder Fettdruck umrissen), werden alle Zeilen oder Spalten in der Tabelle auf die gleiche Größe geändert. Dadurch erhalten die ungleichmäßigen waagerechten Zeilen (siehe Abbildung links) einen gleichmäßigen Abstand (siehe Abbildung rechts), weil die <Umschalt>-Taste verwendet wurde.



Eingeben und Auswählen von Datentabellen

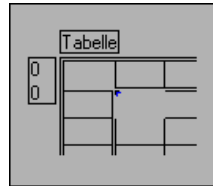
Zur schnellen Eingabe von Daten in eine Tabelle kann die Tastatur verwendet werden. Mit dem Bedien- oder Beschriftungswerkzeug in eine Zelle klicken und die Daten eingeben.

Die <Return>-Taste auf der alphanumerischen Tastatur gibt Text ein und bewegt den Cursor in die Zelle darunter. Die <Eingabe>-Taste auf dem numerischen Tastenfeld gibt Daten ein und beendet den Eintrag. Durch das Drücken der <Umschalt>-Taste bei gleichzeitigem Drücken der Pfeiltasten wird der Eingabecursor zu angrenzenden Zellen verschoben.

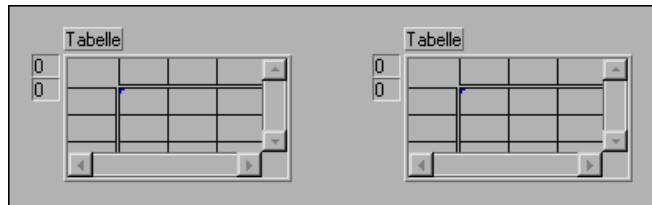
Individuelle Zellen können mit dem Bedienwerkzeug oder durch Doppelklicken gewählt werden, und Sie können die Auswahl erweitern, indem Sie bei niedergehaltener Maustaste die <Umschalt>-Taste drücken und dann ziehen.

Durch das Verschieben außerhalb des aktuellen Inhalts der Tabelle wird die Tabelle gerollt, während gleichzeitig die Auswahl erweitert wird. Eine Umrandung erscheint um ausgewählte Zellen und zeigt an, daß sie ausgewählt wurden. Das Rollen kann ein- oder ausgeschaltet werden, indem der Menüpunkt **Auswahl Rollen** im Popup-Menü der Tabelle verwendet wird.

Außerdem können alle Daten in einer Tabelle gewählt werden oder alle Daten in einer Zeile oder Spalte. Das Bedienwerkzeug in die obere linke Ecke der Tabelle setzen, um alle Daten auszuwählen. Ein besonderer Doppelpfeil-Cursor erscheint, wenn das Werkzeug richtig positioniert wird (siehe nachstehende Abbildung). Auf den Bereich klicken, um die Daten auszuwählen.



Zur Auswahl einer ganzen Zeile oder Spalte mit Daten, das Bedienwerkzeug auf den linken Rand einer Zeile oder ganz oben auf eine Spalte setzen. Es erscheint erneut ein besonderer Doppelpfeil-Cursor, wenn das Werkzeug richtig positioniert wird (siehe nachstehende Abbildung). Zur Auswahl der Daten klicken und über die Breite der Spalte oder die Höhe der Zeile ziehen.

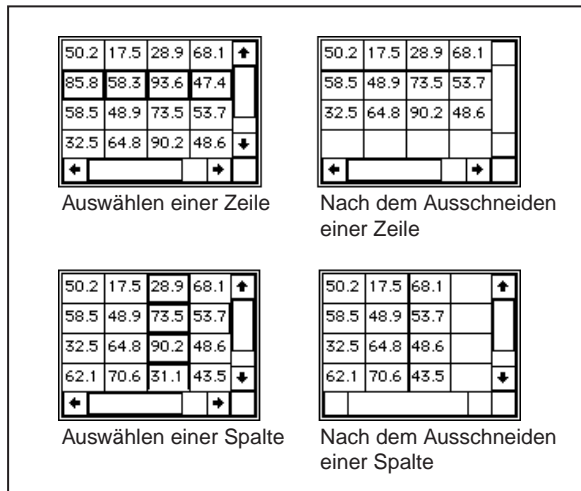


Daten können kopiert, ausgeschnitten und eingefügt werden, und zwar mit den Menüpunkten **Daten kopieren**, **Daten ausschneiden** und **Daten einfügen** im Untermenü **Datenoperationen** des Pop-up-Menüs.

Wenn Sie eine Zeile oder Zeilen mit Daten ausschneiden, werden alle Zeilen darunter nach oben verschoben, wie im oberen Teil der folgenden Abbildung dargestellt. Wenn Sie Spalten mit Daten ausschneiden, werden alle Spalten rechts davon nach links verschoben, wie im unteren Teil der folgenden Abbildung dargestellt.

Wenn Sie einen Teil einer Zeile oder Spalte ausschneiden, wird die gesamte Zeile bzw. Spalte aus der Tabelle entfernt; es werden jedoch nur die ausgewählten Daten in die Zwischenablage kopiert.

Beim Einfügen ohne vorheriges Auswählen einer Zielstelle werden Zeilen und Spalten der Tabelle nach Bedarf hinzugefügt.



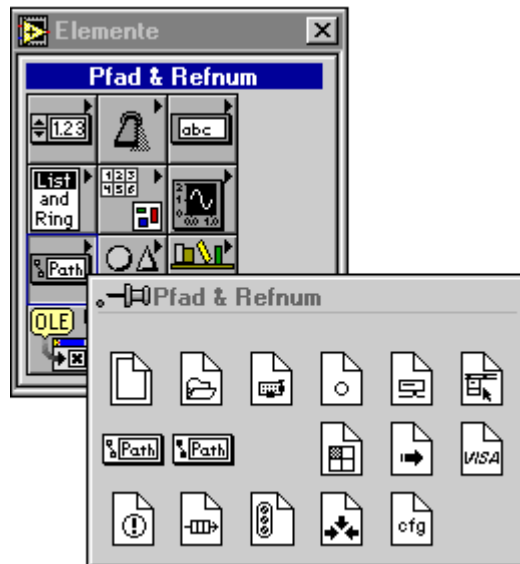
Zum Anzeigen eines ausgewählten Datenbereichs können Sie den Punkt ein- oder ausschalten, indem Sie **Datenoperationen»Auswahl anzeigen** aus- oder abwählen.

Die Zeilen- und Spaltenüberschriften einer Tabelle sind nicht Teil der Tabellendaten. Tabellenüberschriften werden als separate Dateninformationen betrachtet, und sie können mit dem Attributknoten gelesen und eingestellt werden.

Die Tabelle entspricht in bezug auf das Blockdiagramm einem zweidimensionalen Array mit Strings. Deshalb können String-Funktionen Tabellen manipulieren. Informationen über die Verwendung dieser Funktionen finden Sie unter **Online-Referenz** unter dem **Thema»Funktion und VI-Referenz»String-Funktionen**.

Bedien- und Anzeigeelemente vom Typ Pfad und Refnum

In diesem Kapitel wird beschrieben, wie Dateipfad-Bedienelemente und Refnums verwendet werden, die aus der **Elemente»Pfad & Refnum**-Palette ausgewählt werden können (siehe nachfolgende Abbildung).



Bedien- und Anzeigeelemente vom Typ Pfad

Ein Bedienelement und Anzeigeelement vom Typ Pfad sind in folgender Abbildung dargestellt.





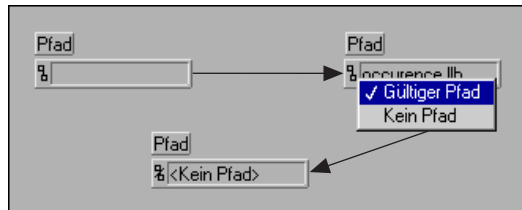
Pfad-Symbol



Kein Pfad-Symbol

Mit der standardmäßigen Syntax für eine gegebene Plattform können Pfad-Bedien- und Anzeigeelemente zur Eingabe bzw. Anzeige der Speicherstelle einer Datei oder eines Verzeichnisses in einem Dateisystem verwendet werden. Wenn eine Funktion, die einen Pfad zurückgeben soll, fehlschlägt, wird ein ungültiger Pfad zurückgegeben. Wenn ein Pfad- Bedien- oder Anzeigeelement einen ungültigen Pfad anzeigt, ändert sich das Pfad-Symbol in ein Kein Pfad-Symbol, und `<Kein Pfad>` erscheint im Textfeld und zeigt an, daß der Pfad ungültig ist.

Der Wert eines Pfad-Bedienelements kann von einem gültigen Pfad zu einem ungültigen Pfad geändert werden, indem Sie auf das Pfad-Symbol klicken und den Menüpunkt **Kein Pfad** aus dem Menü auswählen. Dies wird in folgender Abbildung dargestellt. In derselben Weise kann der Wert eines Pfad-Bedien- oder Anzeigeelementes von einem ungültigen Pfad zu einem gültigen Pfad geändert werden, indem Sie auf das Kein Pfad-Symbol klicken und den Menüpunkt **Gültiger Pfad** aus dem Menü auswählen.



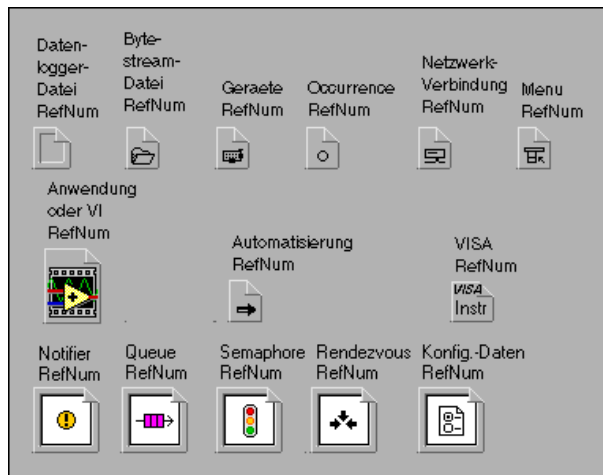
Dieser Wert **Kein Pfad** kann als Standardwert für ein Pfad-Bedienelement auf Ihrem VI verwendet werden. Auf diese Weise können Sie feststellen, wenn der Anwender keinen Pfad eingegeben hat. In diesem Fall das Dialogfeld zur Auswahl eines Pfades wählen.

Ein leerer Pfad in einem Pfad-Bedienelement erscheint als leerer String im Bedienelement. Wenn dieser leere Pfad mit den Datei I/O-Funktionen verbunden wird, insbesondere mit dem Datei/Verzeichnis-Info-VI und dem Listenverzeichnis-VI, bezieht sich der leere Pfad auf der Windows-Plattform auf die Liste von Laufwerken, die Ihrem Computer zugeordnet sind. Auf der Mac-Plattform bezieht sich der leere Pfad auf die Dateien auf Ihrem Desktop. Auf UNIX-Plattformen bezieht sich der leere Pfad auf das Hauptverzeichnis.

Bedien- und Anzeigeelemente vom Typ Refnum

Die Palette **Pfad & Refnums** enthält mehrere Refnum-Bedienelemente (siehe nachfolgende Abbildung). Eine Refnum ist ein eindeutiger Bezeichner, der zur Identifizierung eines Objektes wie einer Datei, einem Gerät oder selbst einer Netzwerkverbindung mit einem anderen Gerät dient. Zahlreiche Refnums werden zur Bezeichnung des Objektes verwendet, auf das ein I/O-Vorgang reagieren soll, wie z.B. die zu lesende Datei bei der Funktion Datei lesen. Ein Refnum-Bedienelement übergibt eine Refnum in ein VI, und ein Refnum-Anzeigeelement übergibt ein Refnum aus einem VI. Für jedes Refnum-Bedienelement gibt es ein zugeordnetes Anzeigeelement. Ein Refnum-Anzeigeelement kann erstellt werden, indem zuerst ein Refnum-Bedienelement angelegt wird. Dann das Bedienelement mit der rechten Maustaste anklicken und den Menüpunkt **In Anzeigeelement ändern** aus dem Menü auswählen.

Folgende Abbildung zeigt die verschiedenen Arten von Refnums.



Wenn Sie eine Datei öffnen, müssen Sie den Pfad der zu öffnenden Datei angeben. Eine *Refnum* wird zurückgegeben, die diese Datei bezeichnet. Diese Refnum kann in allen darauffolgenden Vorgängen verwendet werden, die sich auf diese Datei beziehen. Diese Refnum ist sozusagen eine eindeutige Nummer, die jedesmal dann erzeugt wird, wenn Sie eine Datei zur Bezeichnung. Wenn Sie die Datei schließen, wird der Bezug der Refnum von der Datei getrennt. Nachfolgend sind Beschreibungen der Refnums in der oben dargestellten Abbildung aufgeführt, die in der Palette **Elemente»Pfad & Refnum** zu finden sind.

Es gibt zwei Arten von Datei-Refnums, eine für Datenprotokoll-Dateien und eine für Byte-Stream-Dateien.

- **Datenprotokoll-Datei-RefNum**—Da Datenprotokoll-Dateien über eine inhärente Struktur verfügen, wird mit der Option “Datenprotokoll-Datei-RefNum” die Refnum sowie eine Beschreibung des Dateityps an (oder von) aufrufenden VIs übergeben. Die Größe der “Datenprotokoll-Datei-RefNum” kann wie bei einem Cluster angepaßt werden. Sie plazieren ein Bedienelement in die Refnum, womit die Struktur der Datei definiert wird. Für eine Datei mit Zahlen können Sie eine Datenprotokoll- Refnum mit einer Zahl erstellen. Wenn jeder Datensatz in der Datei ein Zahlenpaar enthält, plazieren Sie einen Cluster in die Refnum und anschließend zwei numerische Bedienelemente in den Cluster.
- **Byte-Stream-Datei-RefNum**—Die Byte Stream File RefNum wird mit Byte-Stream-Dateien verwendet; dabei handelt es sich entweder um Text- oder binäre Dateien. Normalerweise wird sie dann verwendet, wenn Sie eine Datei in einem VI öffnen oder erstellen, jedoch I/O auf der Datei in einem anderen VI durchführen möchten. Sie müssen ein Refnum-Bedienelement auf dem Frontpanel des VIs haben, das I/O durchführt, und ein Refnum-Anzeigeelement auf dem Frontpanel des VIs, das die Datei öffnet oder erstellt.

Die verbleibenden Refnums sind leicht zu verwenden, da sie über keine Optionen verfügen.

- **Geräte-RefNum**—Die Geräte-RefNum wird mit den Geräte-I/O-Funktionen verwendet, die für den Macintosh zur Verfügung stehen. Sie wird nur dann verwendet, wenn Sie ein Gerät in einem VI öffnen, I/O jedoch auf dem Gerät in einem anderen VI durchführen möchten.



Hinweis

Ein Geräte-RefNum-Bedienelement auf Ihrem Frontpanel ist nur selten notwendig. Diese Bedienelemente werden in Geräte-I/O-VIs in `vi.lib` für den Zugriff auf die seriellen Gerätetreiber von G verwendet. Ein Geräte- RefNum-Bedienelement ist nur zum Schreiben eines speziellen G-Gerätetreibers für den Macintosh erforderlich.

- **Occurrence-RefNum**—Die Occurrence-RefNum wird mit den Occurrence-Funktionen verwendet. Sie kann nur verwendet werden, wenn Sie ein Ereignis in einem VI erzeugen, das Ereignis jedoch in einem anderen VI eingestellt oder auftreten soll.
- **Netzwerkverbindung-RefNum**—Die Netzwerkverbindung-RefNum wird mit den TCP/IP-VIs verwendet. Sie wird im allgemeinen dann verwendet, wenn Sie eine Netzwerkverbindung in

einem VI öffnen möchten, I/O jedoch bei der Netzwerkverbindung in einem anderen VI durchführen möchten.

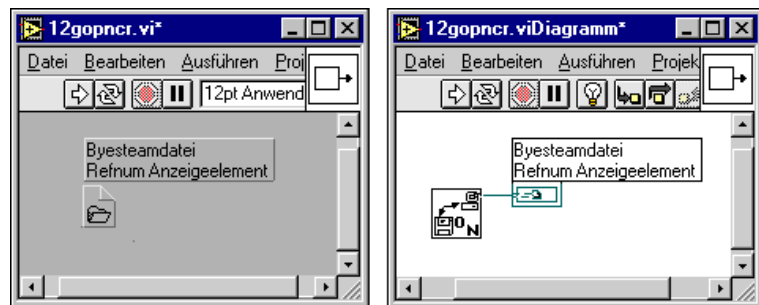
- **Anwendungs- oder VI-Refnum**—Das Anwendungs- oder VI-Refnum-Bedienelement wird mit den VI-Serverfunktionen verwendet. Dieses Refnum-Bedienelement wird verwendet, wenn Sie eine Referenz entweder zu einer LabVIEW-Anwendung oder einem VI innerhalb einer LabVIEW-Anwendung öffnen und die Referenz als Parameter an ein anderes VI übergeben werden soll. Durch die Übergabe einer dieser Refnums an eine der VI-Server-Funktionen können Sie das Verhalten der Anwendung und bestimmter VIs steuern. Durch die Auswahl von **VI Server-Klasse auswählen** aus dem Refnum-Bedienelement-Untermenü können Sie den Datentyp des Bedienelements als Anwendungs-Refnum, VI-Refnum oder als strikte Type VI-Refnum festlegen. Strikte Typen-VI-Refnums verfügen über Datentypeninformationen, die das Anschlußfeld des VIs umfassen und daher zum Aufrufen eines dynamisch geladenen VIs mit der Funktion Aufruf nach Referenz verwendet werden. Mit dem Menüpunkt **VI Server-Klasse auswählen»Durchsuchen...** können Sie ein VI von der Festplatte auswählen, das über das gewünschte Anschlußfeld verfügt. Außerdem können Sie ein VI-Icon/Anschlußfeld oder ein SubVI-Icon von einem Diagramm oder dem Hierarchie-Fenster ziehen und auf ein VI-Refnum-Bedienelement fallenlassen, um den Typ des strikten Typ-VI-Refnum-Bedienelements anzugeben.

Während VI-Refnum-Bedienelemente normalerweise zur Übergabe einer VI-Refnum von einem VI an ein anderes verwendet werden, ist ein strikter Typ-VI-Refnum-Bedienelement außerdem als Eingabe zur Spezifizierung des Typs für die Funktion “Open VI Reference” erforderlich, wenn eine strikter Typ-Referenz zu einem VI gewünscht wird. In diesem Fall ist der Wert des Refnum-Bedienelements unwichtig. Nur der Typ wird durch die Funktion verwendet.

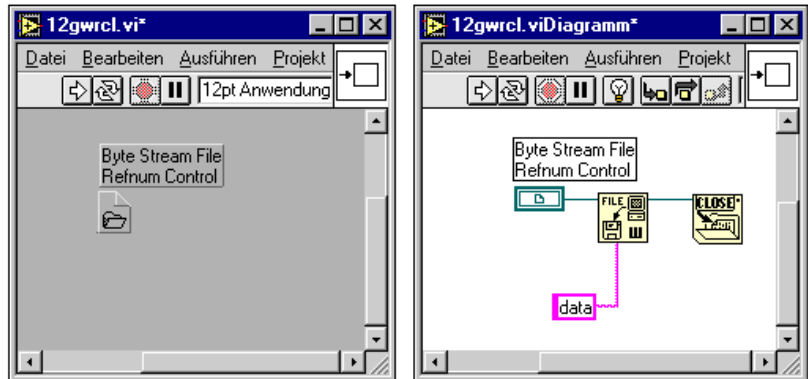
- **Automation RefNum**—Die Automation-Refnum ist eine Referenz zu einem Automation-Objekt.
- **VISA RefNum**—Eine VISA-Refnum ist eine Referenz zu einer VISA-Ressource.
- **Notifier RefNum**—Die Notifier RefNum wird bei Meldungs-VIs verwendet. Sie wird im allgemeinen dann verwendet, wenn Sie einen Benachrichtiger in einem VI erstellen, jedoch auf die Benachrichtigung warten möchten oder eine Benachrichtigung in ein anderes VI senden möchten.

- **Queue RefNum**—Die Queue RefNum wird bei Warteschlangen-VIs verwendet. Sie wird im allgemeinen dann verwendet, wenn Sie eine Warteschlange in einem VI erstellen, jedoch Warteschlangenelemente in ein anderes VI einfügen oder daraus entfernen möchten.
- **Semaphore RefNum**—Die Semaphore RefNum wird bei Semaphore VIs verwendet. Sie wird im allgemeinen dann verwendet, wenn Sie ein Semaphor in einem VI erstellen, jedoch das Semaphor in einem anderen VI erfassen oder freigeben möchten.
- **Rendezvous RefNum**—Die Rendezvous RefNum wird mit Rendezvous-VIs verwendet. Sie wird im allgemeinen dann verwendet, wenn Sie ein Rendezvous in einem VI erstellen, jedoch beim Rendezvous in einem anderen VI warten möchten.

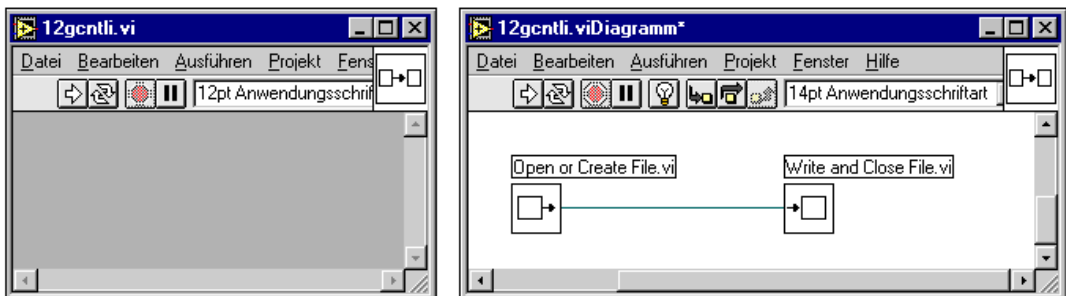
Folgende Abbildung zeigt das Frontpanel und Blockdiagramm eines VIs, das eine Datei öffnet oder erstellt und eine Byte-Stream-Datei-Refnum ausgibt, die zum Aufrufen der Datei in anderen VIs verwendet werden kann.




Folgende Abbildung zeigt das Frontpanel und Blockdiagramm eines VIs, das Daten in eine Datei schreibt und sie dann schließt; diese Datei wird durch die Byte-Stream-Datei-Refnum spezifiziert, die über ein Byte-Stream-Datei-Refnum-Bedienelement in das VI übergeben wurde.



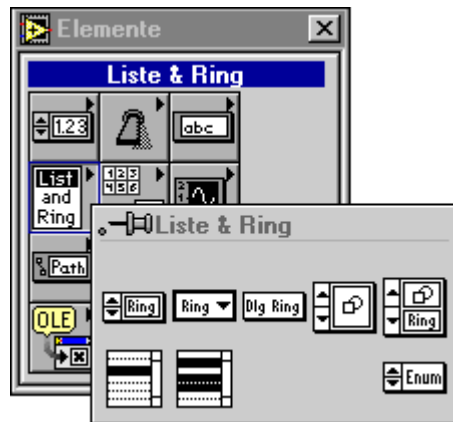
Folgende Abbildung zeigt das Frontpanel und Blockdiagramm eines VIs, das eine Byte-Stream-Datei-Refnum, die zum Aufrufen einer Datei verwendet wird, die durch ein SubVI geöffnet oder erstellt wurde, an ein anderes SubVI übergibt, das Daten in die Datei schreibt und sie dann schließt.



 **Hinweis** *Verwenden Sie ein Refnum-Bedien- oder Anzeigeelement, wenn Sie eine Refnum in ein oder aus einem VI übergeben möchten.*

Bedien- und Anzeigeelemente vom Typ Liste und Ring

In diesem Kapitel werden die Bedien- und Anzeigeelemente vom Typ Listenfeld und Ring beschrieben, die in der Palette **Elemente» Liste & Ring** zur Verfügung stehen (siehe nachstehende Abbildung).



Die Palette **Liste & Ring** enthält acht Bedienelemente—Textring, Menüring, Dialogring, Bildring, Text- & Bildring, Einzelauswahl-Listenfeld, Mehrfachauswahl-Listenfeld und Enumerated Type. In diesem Kapitel werden die Bedienelemente vom Typ Ring, Listenfeld und Enumerated Type behandelt, und zwar in dieser Reihenfolge. Informationen über die Attribute Bedienelemente vom Typ Listenfeld und Ring finden Sie in Kapitel 22, [Attributknoten](#).

Bedienelemente vom Typ Listenfeld

Bedienelemente vom Typ Listenfeld bieten dem Anwender eine Liste von Menüpunkten. Ein Beispiel wird in folgender Abbildung gezeigt. Es gibt zwei Arten von Bedienelementen vom Typ Listenfeld—Einzelauswahl zur Auswahl nur eines Punkts und Mehrfachauswahl zur Auswahl von einem oder mehreren Objekten.



Es wird nicht nur eine Liste von Textobjekten angezeigt; darüber hinaus wird neben jedem Objekt eines von mehreren verschiedenen Symbolen gezeichnet (wie im Dialogfeld **Speichern**, in dem Verzeichnisse und Dateien über unterschiedliche Symbole verfügen). Individuelle Objekte können außerdem deaktiviert werden, und zwischen Objekten können Trennlinien eingestellt werden. Das (die) aktuell ausgewählte(n) Objekt(e) können durch Ablesen des Wertes des Bedienelements erkannt werden. Durch die Verwendung eines Attributknotens können Sie außerdem feststellen, auf welches Objekt der Anwender gegebenenfalls doppelgeklickt hat. Und schließlich können Sie die Objekt-Strings und -Symbole einstellen und ob während der Verwendung des Attributknotens die Objekte deaktiviert werden.

Erstellen einer Liste mit Objekten

Wenn Sie ein Listenfeld auf ein Frontpanel plazieren, enthält es keine Objekte. Eine Liste mit Objekten kann im Bearbeitungsmodus per Eingabe mit dem Beschriftungswerkzeug erstellt werden bzw. im Ausführungsmodus durch die Verwendung des Attributknotens. Die einzelnen Objekte können durch Eingabe eines Wagenrücklaufzeichens (CR) voneinander getrennt werden. Wenn Sie die Objekte schon während der Erstellung des VIs bestimmen können, ist es ratsam, im Bearbeitungsmodus Eingaben zu machen. Wenn die Objekte zum Ausführungszeitpunkt bestimmt werden (z.B. eine Liste mit Dateien), sollte der Attributknoten verwendet werden.

Auswählen von Listenfeld-Objekten

Objekte in einem Listenfeld können auf dreierlei Weise ausgewählt werden—durch Verwendung der Maus, durch Verwendung der Pfeiltasten oder durch Eingabe eines Teils des Namens des gewünschten Objekts.

- Maus—Zum Auswählen von Objekten das Bedienwerkzeug verwenden. Mit dem Mehrfachauswahl-Listenfeld können Sie mehr als ein Objekt auswählen, indem Sie auf zusätzliche Objekte klicken und die <Umschalt>-Taste gedrückt halten.
- Pfeiltasten—Mit den Pfeiltasten können Objekte gewählt werden. Zum Hinzufügen des momentan hervorgehobenen Objekts zur Liste ausgewählter Objekte die Leertaste drücken. Die Auswahl eines momentan ausgewählten Objekts kann aufgehoben werden, indem Sie mit den Pfeiltasten zum Objekt gehen und anschließend die Leertaste drücken.
- Eingabe eines Teils des Namens des gewünschten Objekts—Zum Auswählen des Objekts den Namen des Objekts eingeben. Außerdem können Sie die Nach-Links-Taste oder Nach-Rechts-Taste verwenden, um zum vorherigen oder nächsten Objekt zu gehen, das mit den eingegebenen Buchstaben übereinstimmt.

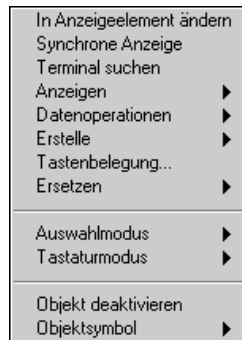
Listenfeld-Datentypen

Der Datentyp für ein Einzelauswahl-Listenfeld ist `Int32`. Der Datenwert für das Einzelauswahl-Listenfeld ist eine Zahl, die das momentan ausgewählte Objekt darstellt, wobei das erste Objekt einen Wert von Null hat. Ist kein Objekt ausgewählt, ist der Wert `-1`.

Das Mehrfachauswahl-Listenfeld ist ein Array von `Int32s`, wobei der (die) Wert(e) im Array das (die) momentan ausgewählte(n) Objekt(e) darstellt (darstellen). Ist kein Objekt ausgewählt, ist der Wert ein leeres Array.

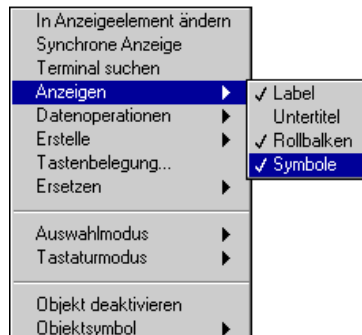
Popup-Menüobjekte des Listenfeldes

Das Popup-Menü des Listenfeldes enthält die in folgender Abbildung dargestellten Objekte. In diesem Abschnitt werden die Objekte beschrieben, die eindeutig den Listenfeld-Bedienelementen zugeordnet sind.

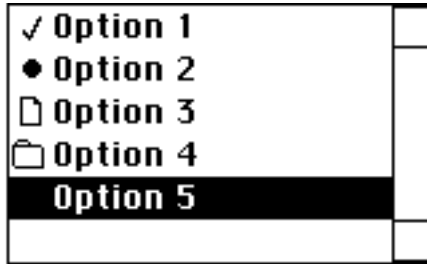


Anzeigen

Der Menüpunkt **Anzeigen** aus dem in folgender Abbildung dargestellten Popup-Menü wird zur Auswahl der Komponenten verwendet, die im Listenfeld erscheinen sollen. Außerdem können der Label, der Rollbalken und die Symbole angezeigt oder verborgen werden.



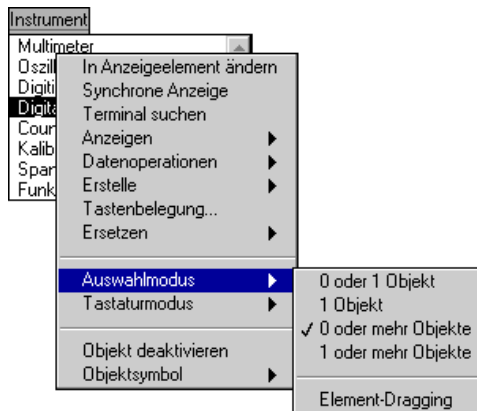
Wenn **Anzeigen»Symbole** ausgewählt wird, fügt das Listenfeld eine zusätzliche Spalte links von der Liste für Symbole hinzu. Ein Beispiel hierfür wird in folgender Abbildung dargestellt. Anfänglich verfügen keine der Objekte über Symbole. Mit dem Menüpunkt **Objektsymbol** oder **Attributknoten** können Symbole hinzugefügt werden.



Auswahlmodus

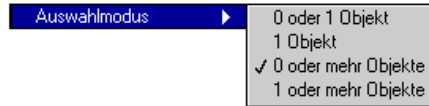
Der Menüpunkt **Auswahlmodus** im Popup-Menü wird zur Anzeige dessen verwendet, wie viele Objekte auf einmal ausgewählt werden können.

Für ein Einzelauswahl-Listenfeld wird das Listenfeld durch Auswahlmodus-Objekte so konfiguriert, daß entweder Null oder ein Element unterstützt werden (siehe nachstehende Abbildung).



Wenn **1 Objekt** ausgewählt wird (die Standardeinstellung), wird ein Objekt im Listenfeld stets ausgewählt (hervorgehoben). Wenn Sie ein anderes Objekt anklicken, wird das neue Objekt ausgewählt (hervorgehoben), und die Auswahl des vorherigen Objekts wird aufgehoben. Wenn Sie **0 oder 1 Objekt** auswählen, hat dies das gleiche Verhalten zur Folge mit der Ausnahme, daß ein Anwender die Auswahl des aktuellen Objekts aufheben kann, indem er darauf <Umschalt>-klickt, so daß keine Objekte ausgewählt sind. Mit dem Menüpunkt **Element-Drugging** können Sie Objekte im Listenfeld verschieben, indem sie mit der Maus gezogen werden.

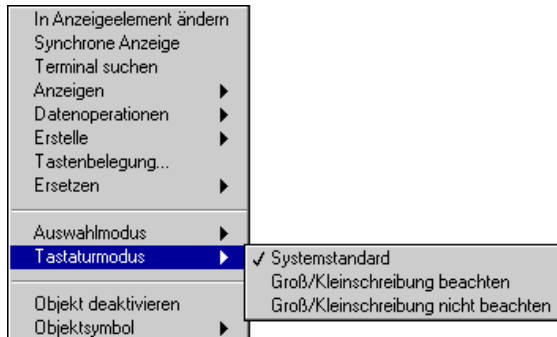
In einem Mehrfachauswahl-Listefeld finden Sie dieselben Auswahlmodus-Objekte wie im Einzelauswahl-Listefeld. Darüber hinaus können Sie mehrere Objekte zur Auswahl markieren (siehe nachstehende Abbildung).



Standardmäßig verfügt das Mehrfachauswahl-Listefeld über einen Auswahlmodus, in dem der Anwender **0 or mehr Objekte** auswählen kann. Außerdem können Sie **1 oder mehr objekte** auswählen, um einen Mehrfachauswahlmodus aufzurufen, wobei mindestens ein Objekt gewählt sein muß.

Tastatur-Modus

Sie können **Tastaturmodus** im Popup-Menü verwenden (siehe nachstehende Abbildung), um anzugeben, wie im Listefeld mit Groß- und Kleinbuchstaben verfahren werden soll, wenn Sie Objekte durch Eingabe ihrer Anfangsbuchstaben auswählen.

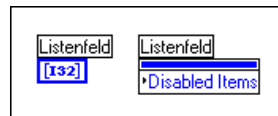


Wenn Sie **Groß/Kleinschreibung beachten** auswählen, wird ein Objekt nur dann ausgewählt, wenn die Groß- und Kleinbuchstaben der Anwendereingabe genau mit dem Objekt übereinstimmen. Wenn **Groß/Kleinschreibung nicht beachten** gewählt wird, wird die Eingabe nicht nach Groß- und Kleinschreibung unterschieden. Wenn **Systemstandard** ausgewählt wird (die Standardeinstellung für das Listefeld), verhält sich das Listefeld genauso wie andere Listfelder einer gegebenen Plattform. So wird in Windows und auf dem Macintosh im Listefeld nicht nach Groß- und Kleinschreibung unterschieden. In UNIX wird im Listefeld nach der Groß- und Kleinschreibung unterschieden.

Objekt deaktivieren

Ein Listenfeld-Objekt kann deaktiviert werden, indem Sie das Popup-Menü des Objekts aufrufen und **Objekt deaktivieren** auswählen. Zum Aktivieren eines deaktivierten Objekts das Popup-Menü des Objekts aufrufen und **Objekt aktivieren** auswählen.

Zum programmatischen Aktivieren/Deaktivieren eines Objekts den Attributknoten **Disabled Items** verwenden (ein Array von Objekten), wie in folgender Abbildung dargestellt wird.

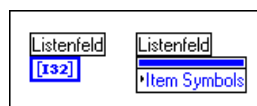


Objektsymbol und Trennlinie

Ein Symbol eines Listenfeldes kann eingestellt werden, indem Sie das Popup-Menü des Objekts aufrufen und eines der Objekte aus dem Untermenü **Objektsymbol** auswählen. Durch die Auswahl des Objekts in der unteren rechten Ecke wird das Listenobjekt durch eine Trennlinie ersetzt.



Zur programmatischen Einstellung von Listenobjekten den Attributknoten **Objektsymbole** verwenden (ein Array von Objektsymbolen), wie in folgender Abbildung dargestellt.

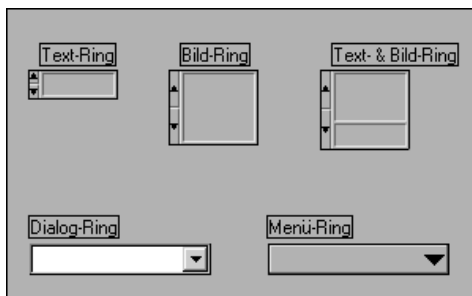


Mit der Konstante **Listenfeld Symbolring** in der Palette **Funktionen» Numerisch»Zusätzliche Num Konstanten** können Sie Objektsymbole angeben.

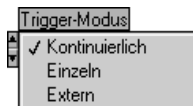
Wenn Sie einen Wert von -1 für das Symbol eines Objekts verwenden, zeichnet das Listenfeld statt des Objektname eine als nicht bedienbar markierte Linie.

Bedienelemente vom Typ Ring

Ringe sind besondere numerische Objekte, die numerische Werte mit Strings, Bildern oder beiden assoziieren. Die unterschiedlichen Ringstilarten werden in folgender Abbildung dargestellt.



Ringe sind besonders nützlich zur Auswahl von sich gegenseitig ausschließenden Objekten, wie etwa Trigger-Modi. Anwender sollen z.B. zwischen kontinuierlicher, einzelner und externer Triggerung wählen können (siehe nachstehende Abbildung).



Im vorangegangenen Beispiel hat der mit **Trigger-Modus** beschriftete Ring drei Modus-Deskriptoren nacheinander in der Textanzeige des Rings. G ordnet die Objekte in einer ringförmigen Liste an (wie ein Rolodex), wobei immer nur je ein Objekt auf einmal sichtbar ist. Jedes Objekt verfügt über einen numerischen Wert, der von Null bis $n - 1$ reicht, wobei n die Anzahl der Objekte darstellt (in diesem Beispiel drei). Der Wert des Rings jedoch kann ein beliebiger Wert im entsprechenden numerischen Datenbereich sein. Das letzte Objekt (**Extern**, s.o.) für einen beliebigen Wert größer als oder gleich zwei und das erste Objekt (**Kontinuierlich**,

s.o.) für einen beliebigen Wert kleiner als oder gleich Null werden dort angezeigt.

Sie können das kontextsensitive Menü aufrufen und **Anzeigen»Digitale Anzeige** auswählen, um den numerischen Wert auszuwählen, der mit dem aktuellen Objekt eines Rings assoziiert wird.

Anwender können diesen Ring zur Auswahl eines Objekts aus einer leicht verständlichen Liste verwenden, ohne daß sie genau wissen müssen, welchen Wert ein Objekt darstellt. Der mit dem ausgewählten Objekt assoziierte Wert wird an das Blockdiagramm übergeben, wo Sie z.B. einen Case aus einer Case-Struktur (bedingter Code) auswählen können, der das ausgewählte Objekt durchführt.

Ein Objekt in einem Bedienelement vom Typ Ring kann auf zweierlei Arten ausgewählt werden. Mit den Inkrement-Tasten zum nächsten oder vorherigen Objekt gehen. Die Inkrementierung wird kreisförmig durch die Liste der Objekte fortgeführt, solange die Maustaste gedrückt bleibt. Außerdem können Sie ein beliebiges Objekt direkt auswählen, indem Sie auf den Ring mit dem Bedienwerkzeug klicken und dann das gewünschte Objekt aus dem aufgerufenen Menü auswählen. Der Menü-Ring sieht jedoch genauso aus wie ein Pulldown-Menü und läßt sich auch so bedienen. Er hat keine Inkrement-Tasten. Ein Objekt kann durch Klicken auf den Ring und Auswahl des Objekts aus dem aufgerufenen Menü gewählt werden.

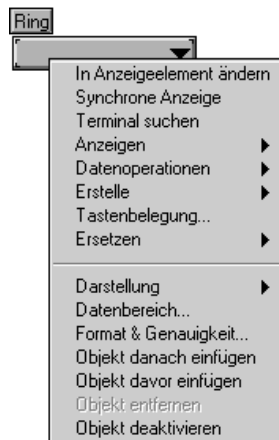
Hinzufügen von Textobjekten zu Ringen

Ein neuer Textring hat ein Objekt mit einem Wert von Null und eine Anzeige mit einem leeren String. Die Eingabe bzw. Änderung von Text im Ring-Textbereich erfolgt wie bei Labels, und zwar mit dem Beschriftungswerkzeug. Die <Eingabe>- (**Windows** und **HP-UX**) oder <Return>- (**Macintosh** und **Sun**) Taste drücken oder außerhalb des Textbereichs klicken, um die Eingabe zu beenden. Mit dem Menüpunkt **Objekt danach einfügen** aus dem Popup-Menü des Textrings wird ein neues leeres Objekt erstellt, das auf das aktuelle Objekt folgt. Mit dem Menüpunkt **Objekt davor einfügen** wird ein neues Objekt vor dem aktuellen Objekt hinzugefügt. Wenn Sie z.B. Objekt 0 bearbeiten und dann **Objekt danach einfügen** auswählen, wird Objekt 1 erstellt, und Sie können Text für das neue Objekt eingeben. Sie können außerdem <Umschalt-Eingabe> (**Windows** und **HP-UX**) oder <Umschalt-Return> (**Macintosh** und **Sun**) drücken, wenn Sie die Eingabe von Text in einem Objekt abgeschlossen haben und zu einem neuen Objekt vorrücken möchten.

Für jeden Ring erhöhen sich die Objektwerte über die Einfügemarke um eins, um die entsprechende Anpassung an das neue Objekt vorzunehmen. Wenn Sie z.B. ein Objekt nach Objekt 4 einfügen, wird das neue Objekt zu Objekt 5, das vorherige Objekt 5 wird zu Objekt 6, 6 wird zu 7 und so weiter. Wenn Sie ein Objekt vor Objekt 4 einfügen, wird das neue Objekt zu Objekt 4, das vorherige Objekt 4 wird zu Objekt 5, 5 wird zu 6 und so weiter.

Mit dem Befehl **Objekt entfernen** im Popup-Menü des Rings können Sie ein beliebiges Objekt löschen. Wie bei den Befehlen zum Hinzufügen von Objekten werden die numerischen Werte der Objekte automatisch angepaßt.

Zum Deaktivieren von Ringobjekten während der Bearbeitung das Popup-Menü eines Bedienelements vom Typ Ring aufrufen und **Objekt deaktivieren** auswählen (siehe nachstehende Abbildung). Anschließend das Objekt auswählen, das deaktiviert werden soll.

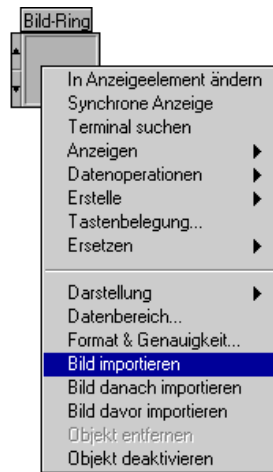


Wenn Sie während der Bearbeitung ein deaktiviertes Objekt aktivieren möchten, das kontextsensitive Menü aufrufen und **Objekt aktivieren** auswählen. Zum programmatischen Deaktivieren/Aktivieren eines Ringobjekts den Attributknoten **Disabled Items** auswählen (ein Array von Indizes), wie in folgender Abbildung dargestellt.



Hinzufügen von Bildobjekten zu Ringen

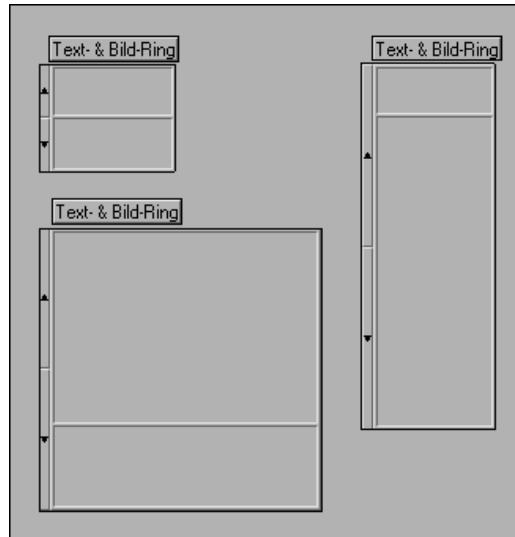
Ein neuer Bild-Ring oder Text- & Bild-Ring hat ein Objekt mit einer leeren Bildanzeige. Ein Bild muß zuerst in die Zwischenablage kopiert werden, bevor es in einen Bild-Ring importiert werden kann. Wenn Sie ein Bild in der Zwischenablage haben, das Popup-Menü des Bild-Rings aufrufen und **Bild importieren**. Zum Hinzufügen eines weiteren Bildes, dieses in die Zwischenablage kopieren, dann das Popup-Menü des Bild-Rings aufrufen und **Bild davor importieren** oder **Bild danach importieren auswählen** (siehe nachstehende Abbildung).



Ändern von Größe und Text von Text- & Bild-Ringen

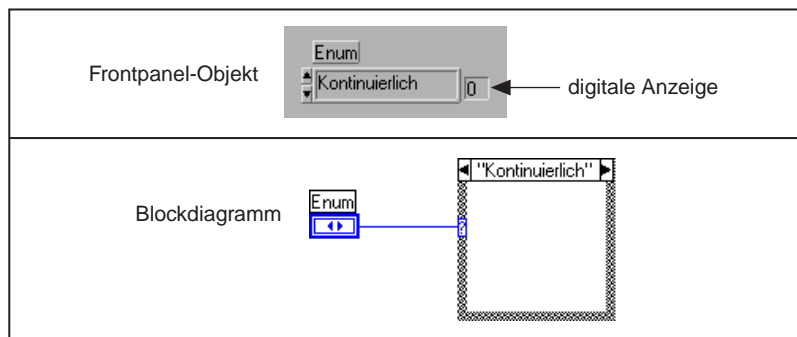
Der Text- & Bild-Ring kann vergrößert werden, um mehr Platz für den Inhalt zu schaffen, so wie alle Bedienelemente vergrößert werden können. Außerdem können Sie den Ring verkleinern. Wenn Sie die Größe des Text- & Bild-Rings von einer der unteren Ecken aus ändern, verändert sich die Höhe des Textbereichs. Wenn Sie den Text- & Bild-Ring von einer der

oberen Ecken aus ändern, verändert sich die Höhe des Bildbereiches. Die Funktionalität zur Größenänderung wird in folgender Abbildung dargestellt.



Enumerated Type-Bedienelemente

Ein Enumerated Type-Bedienelement gleicht einem Text-Ring-Bedienelement. Wenn jedoch Daten von einem Enumerated Type mit einer Case-Struktur verbunden werden, zeigt der Case den String oder Text des Objekts anstelle einer Zahl an. Dieser Enumerated Type-Datentyp ist ein vorzeichenloser Byte-, vorzeichenloser Wort- oder vorzeichenloser Long-Integer-Wert, der aus der Palette **Darstellung** auswählbar ist. Ein Beispiel wird in folgender Abbildung dargestellt.



Ein Vorteil der Verwendung von Enumerated Type anstelle von Bedienelementen vom Typ Ring ist folgender: wenn Sie einen Enumerated Type auf dem Anschlußfeld eines SubVIs verwenden und dann zur Erstellung einer Konstante, eines Bedien- oder Anzeigeelements das kontextsensitive Menü aufrufen, erstellen Sie eine Aufzählung mit den richtigen Strings.

Objekte können in einen Enumerated Type auf dieselbe Weise eingegeben werden wie in einen Ring, und zwar unter Verwendung von **Objekt danach einfügen** oder **Objekt davor einfügen**. Oder Sie können das Beschriftungswerkzeug verwenden. Drücken Sie <Umschalt-Eingabe> (**Windows und HP-UX**) oder <Umschalt-Return> (**Macintosh und Sun**), um ein neues Objekt einzugeben. Nach Abschluß der Eingabe von Objekten auf einen beliebigen Bereich außerhalb des Enumerated Type klicken.

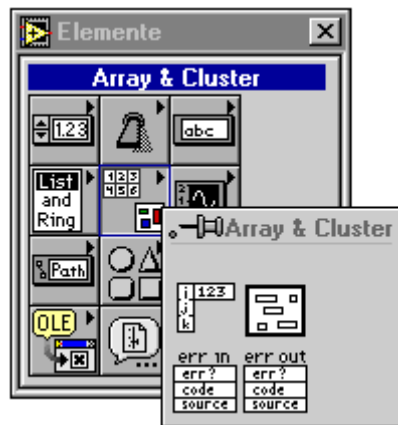
Wenn ein Enumerated Type mit einer Case-Struktur verbunden wird, muß die Case-Struktur über Cases verfügen, die jedes Objekt im Enumerated Type verarbeiten. Wenn die Case-Struktur über einen Standard-Case verfügt, verarbeitet es Objekte, die nicht explizit von anderen Cases verarbeitet werden.

Alle arithmetischen Operationen mit Ausnahme von "Eins addieren" und "Eins subtrahieren" behandeln den Enumerated Type wie einen vorzeichenlosen numerischen Wert. Mit "Eins addieren" wird die letzte Aufzählung zur ersten hochgezählt und mit "Eins subtrahieren" wird die erste Aufzählung zur letzten heruntergezählt. In allen anderen Fällen wird der Enumerated Type als vorzeichenloser Wert behandelt. Wenn ein vorzeichenbehafteter Integer-Wert in das Format eines Enumerated Type umgewandelt wird, werden negative Zahlen an die erste Aufzählung geheftet und positive Zahlen außerhalb des Bereichs an die letzte Aufzählung geheftet. Vorzeichenlose Integer-Werte werden stets an den letzten Wert der Aufzählung geheftet.

Wenn ein numerischer Wert mit einem Enumerated Type-Anzeigeelement verbunden wird, wird die Zahl in das nächste Enum-Objekt konvertiert, wobei Zahlen außerhalb des Bereichs so wie oben beschrieben behandelt werden. Wenn Sie ein Enum-Bedienelement mit einem numerischen Wert verbinden, ist der Wert der Enumerated Type-Index. Zur Verbindung eines Enumeration Type mit einem Enum-Anzeigeelement müssen die Enum-Objekte übereinstimmen, auch wenn das Anzeigeelement Objekte haben könnte, die zusätzlich zu den Objekten im Typ zur Verfügung stehen.

Bedien- und Anzeigeelemente vom Typ Array und Cluster

Dieses Kapitel beschreibt die Verwendungsweise von Arrays und Clustern. Der Zugriff auf Arrays und Cluster erfolgt über die Palette **Elemente» Array & Cluster** (siehe nachstehende Abbildung).



Die angeführten Beispiele finden Sie in `examples\general\arrays.llb`.

Beschreibungen von G-Funktionen, die neben Skalaren Arrays ausführen, finden Sie unter **Hilfe»Online-Referenz»Funktion und VI Referenz**. Die Themen *Array-Funktionen* und *Cluster-Funktionen* beschreiben Funktionen, die ausschließlich für Array- und Cluster-Operationen entwickelt wurden.

Arrays

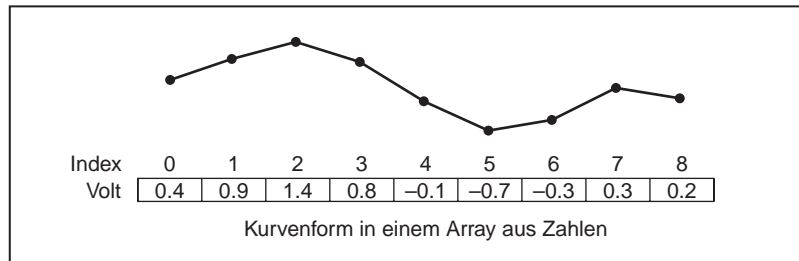
Im Gegensatz zu einem Cluster, bei dem es sich um eine Sammlung fester Größen mit Datenelementen verschiedener Typen handelt, ist ein Array eine größenveränderliche Sammlung von Datenelementen desselben Typs. Arrayelemente werden angeordnet, und der Zugriff auf ein einzelnes Arrayelement wird durch die *Indizierung* des Arrays ermöglicht. Der *Index* ist nullbasiert, das heißt, er liegt im Bereich zwischen Null und $n - 1$, wobei n die Anzahl der Elemente im Array ist (siehe nachstehende Abbildung).

Index	11-Element-Array
0	Ursula
1	Dieter
2	Heinrich
3	Tim
4	Thomas
5	Phillip
6	Wolfgang
7	Ludwig
8	Uschi
9	Claudia
10	Friedrich

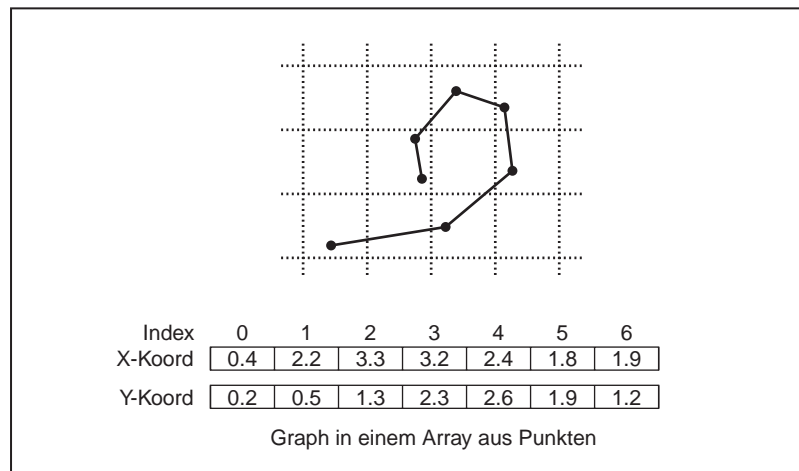
Ein Namensverzeichnis, welches in G als Array aus Strings (siehe nachstehende Abbildung) dargestellt wird, ist ein einfaches Beispiel eines Arrays.

Ute	Paul	Steffi	Dirk	Lutz	Uwe	Klaus	Ernst	Grit	Tim	Erich
Namensverzeichnis in einem Array aus Strings										

Ein anderes Beispiel ist eine Kurve, die als numerisches Array dargestellt wird, bei dem jedes nachfolgende Element den Spannungswert in sukzessiven Zeitintervallen ist (siehe nachstehende Abbildung).

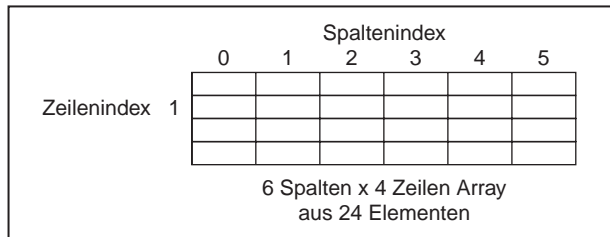


Ein komplexeres Beispiel ist ein Graph, der als Array von Punkten dargestellt wird, wobei jeder Punkt ein Zahlenpaar ist, welches die X- und Y-Koordinaten darstellt (siehe nachstehende Abbildung).



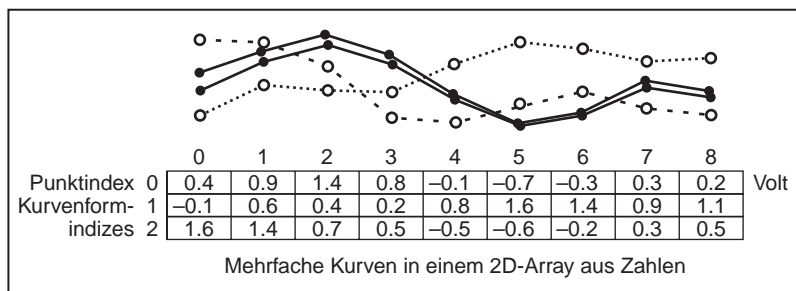
Die vorstehenden Beispiele sind *eindimensionale* (1D) Arrays. Ein *zweidimensionales* (2D) Array erfordert zum Lokalisieren eines Elements zwei Indizes—einen Spaltenindex und einen Zeilenindex, die

beide nullbasiert sind. Man spricht in diesem Fall von einem Array mit N Spalten mal M Zeilen, der N mal M Elemente umfaßt (siehe nachstehende Abbildung).



Ein einfaches Beispiel hierfür ist ein Schachbrett. Es besteht aus acht Spalten und acht Zeilen bzw. insgesamt 64 Feldern, die entweder frei oder besetzt sein können. Sie können ein Schachbrett als zweidimensionales Array aus Strings darstellen. Jeder String hat den Namen einer Figur, die eine entsprechende Position auf dem Brett einnimmt, oder einen leeren String, falls das Feld frei ist. Andere vertraute Beispiele sind u.a. Kalender, Fahrpläne und sogar Fernsehbilder, die als zweidimensionale Arrays aus Zahlen dargestellt werden, die jedem Bildpunkt eine Lichtintensität zuweisen. Computerbenutzer sind mit Tabellenkalkulationsprogrammen vertraut, die Zeilen und Spalten aus Zahlen, Formeln und Text enthalten.

Alle eindimensionalen Arraybeispiele können zu zwei Dimensionen verallgemeinert werden. Die nachstehende Abbildung zeigt eine Sammlung von Kurven, die als zweidimensionales Array aus Zahlen dargestellt wird. Der Zeilenindex wählt die Kurve, und der Spaltenindex wählt den Punkt auf der Kurve.



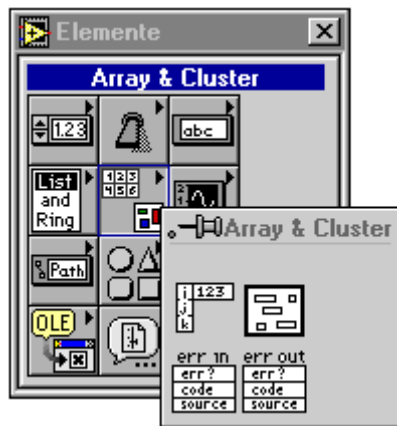
Arrays können eine arbiträre Anzahl von Dimensionen aufweisen, aber zum Lokalisieren eines Elements ist ein Index pro Dimension erforderlich. Der Datentyp der Arrayelemente kann ein Cluster sein, der ein Sortiment von Typen enthält, etwa Arrays, deren Elementtyp ein Cluster ist usw. Ein

Array aus Arrays ist nicht möglich. Benutzen Sie statt dessen ein multidimensionales Array oder ein Array aus Arrayclustern.

Ziehen Sie beim Einsatz einer Sammlung ähnlicher Daten die Verwendung von Arrays in Betracht. Arrays sind oftmals nützlich, wenn Sie sich wiederholende Berechnungen oder I/O-Funktionen durchführen. Durch Arrays kann Ihre Anwendung aufgrund der großen Anzahl von Arrayfunktionen und VIs in G kompakter, schneller und leichter zu entwickeln werden.

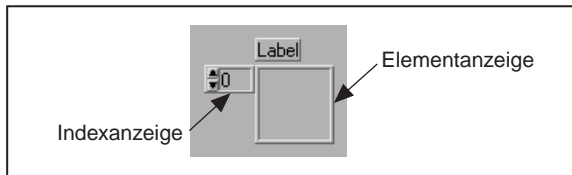
Erstellen von Array-Bedienelementen

Zur Erstellung eines Array-Bedienelements wählen Sie zunächst ein Array aus der Palette **Elemente»Array & Cluster** (siehe nachstehende Abbildung).



Sie erstellen ein Bedien- oder Anzeigeelement vom Typ Array durch Kombinieren einer *Array-Shell* aus der Palette **Array & Cluster** (siehe vorstehende Abbildung) mit einem gültigen *Element*, welches ein numerisches oder Boolesches Element, ein String, Pfad, eine Refnum oder ein Cluster sein kann. Das Element kann kein anderes Array oder Diagramm sein. Handelt es sich bei dem Element ferner um einen Graphen, sind nur die Datentypen des Graphen gültig, die auf höchster Ebene Cluster statt Arrays enthalten.

Durch Auswählen eines Arrays aus der Palette **Array & Cluster** wird eine Array-Shell auf das Frontpanel plaziert (siehe nachstehende Abbildung).



Eine neue Array-Shell besitzt eine *Indexanzeige*, eine leere *Elementanzeige* und ein optionales Label.

Ein Arraytyp kann auf zweierlei Art definiert werden. Ziehen Sie ein Bedien- oder Anzeigeelement des gewünschten Typs ins Elementanzeigefenster, oder platzieren Sie das Bedien- oder Anzeigeelement direkt, indem Sie das Popup-Menü des Frontpanels aufrufen, ein Bedienelement auswählen und in die Elementanzeige ziehen. Bei beiden Methoden füllt das eingefügte Bedien- oder Anzeigeelement die leere Anzeige aus.

Wenn Sie beispielsweise ein Array aus Booleschen Elementen definieren wollen, rufen Sie das Popup-Menü auf, wählen Sie ein Boolesches Bedienelement, und ziehen Sie es in die Elementanzeige (siehe Abbildungen 14-1 bis 14-4).

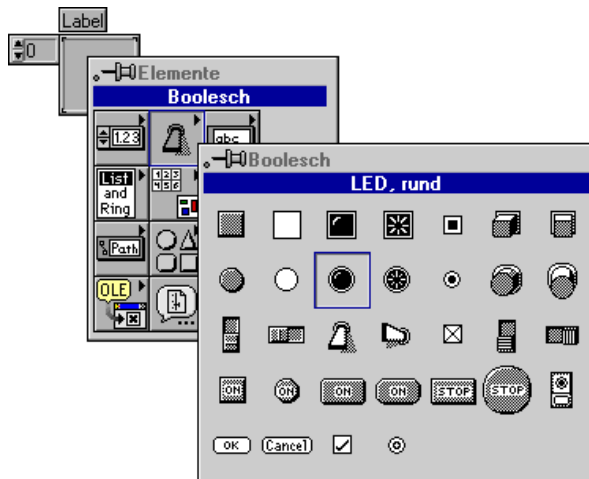


Abbildung 14-1. Popup-Menü in der Elementanzeige aufrufen und ein Boolesches Bedienelement auswählen.

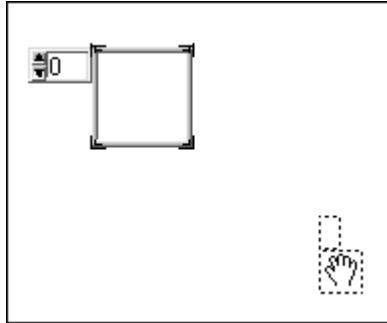


Abbildung 14-2. Maustaste loslassen. Das Menü wird ausgeblendet, und der Cursor nimmt die Gestalt des Icons “Fenster rollen” an.

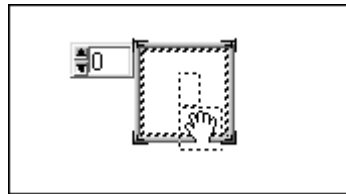


Abbildung 14-3. Ziehen Sie das Icon “Fenster rollen” in die Elementanzeige.

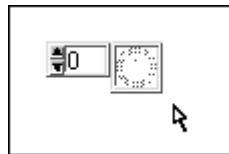
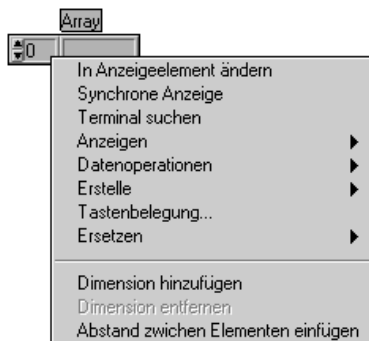


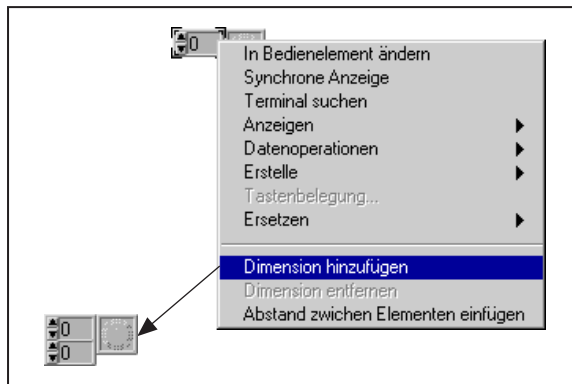
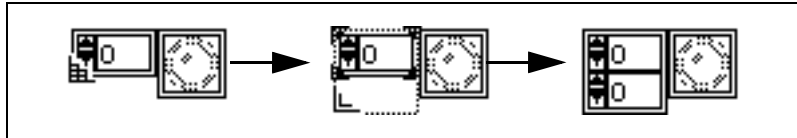
Abbildung 14-4. Klicken Sie die Maustaste. Das Bedienelement wird innerhalb der Anzeige platziert.

Die folgende Abbildung zeigt das Popup-Menü der Indexanzeige.



Array-Dimensionen

Ein neues Array besitzt eine Dimension und eine Indexanzeige. Zum Hinzufügen einer Dimension passen Sie die Indexanzeige entweder vertikal an oder wählen den Menüpunkt **Dimension hinzufügen** aus dem Popup-Menü der Indexanzeige. Zum Löschen einer Dimension verkleinern Sie den Index vertikal oder wählen den Menüpunkt **Dimension entfernen**. Für jede hinzugefügte Dimension erscheint eine zusätzliche Indexanzeige. Die folgende Abbildung zeigt zwei Methoden der Größenanpassung.

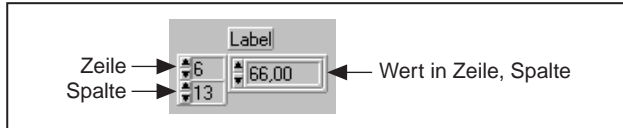


Die Dimension eines Arrays bestimmt die Anzahl der Identifikatoren, die zum Lokalisieren eines Datenobjekts erforderlich sind. Um beispielsweise ein Wort in einem Buch zu finden, müssen Sie eventuell zum sechsten Wort in Zeile 28 auf Seite 192 gehen. Das Wort wird also durch drei Indizes spezifiziert, und zwar 6, 28 und 192, so daß ein dreidimensionales (3D) Array aus Wörtern eine mögliche Darstellung dieses Buchs ist. Um ein Buch in einer Bibliothek zu lokalisieren, geben Sie dessen Position im Regalfach, das Fach, das Regal, den Gang und das Stockwerk an. Wenn Sie dies in einem Array darstellen wollten, wären dazu fünf Dimensionen erforderlich. Die Position eines Wortes in einer Bibliothek erfordert somit acht Indizes.

Beachten Sie, daß sich der Stil, die Stärke und Farbe der Verbindung ändern, wenn Sie einem Array eine Dimension hinzufügen. Weitere Informationen hierzu finden Sie in Kapitel 18, [Verbinden des Blockdiagramms](#).

Array-Indexanzeige

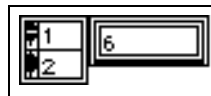
Wenn Sie sich ein zweidimensionales Array als Zeilen und Spalten vorstellen, ist die obere Darstellung der Zeilenindex und die untere Darstellung der Spaltenindex. Die kombinierte Darstellung rechts zeigt den Wert an der angegebenen Position (siehe nachstehende Abbildung).



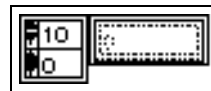
Nehmen wir z.B. an, daß Sie ein aus 3 Zeilen und 4 Spalten bestehendes Array haben, dessen Werte in der nachstehenden Abbildung dargestellt sind.

0	1	2	3
4	5	6	7
8	9	10	11

Zeilen und Spalten sind nullbasiert, d.h. die erste Spalte ist Spalte 0, die zweite Spalte 1 usw. Durch Ändern der Indexanzeige in Zeile 1 wird in Spalte 2 der Wert 6 angezeigt (siehe nachstehende Abbildung).



Wenn Sie versuchen, einen Wert außerhalb des Bereichs darzustellen, erscheint die Wertdarstellung grau, um anzugeben, daß gegenwärtig kein Wert definiert ist. Wenn Sie beispielsweise versuchen, ein Element in Zeile 10 des obigen Arrays darzustellen, erscheint die Darstellung, wie in der nachstehenden Abbildung gezeigt, grau.





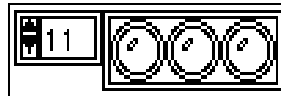
Array-Größenänderungswerkzeug

Darstellung von Arrays in Einzelelement- oder Tabellenform

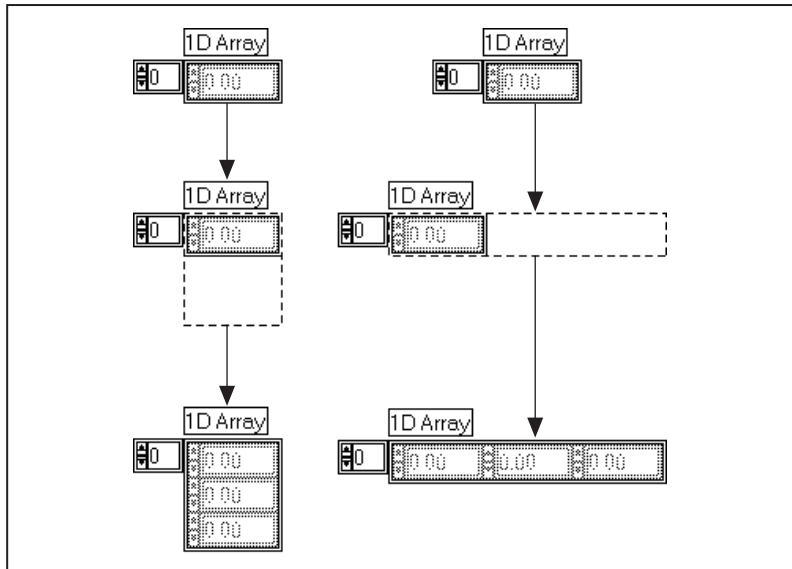
Ein neues Array erscheint in Einzelelementform. Das Array stellt den Wert eines einzelnen Elements dar—und zwar desjenigen, auf das die Indexanzeige verweist. Sie können das Array auch als Tabelle aus Elementen darstellen, indem Sie die Array-Shell an einer der vier das Element umgebenden Ecken in der Größe ändern. Das Größenänderungswerkzeug unterscheidet sich etwas vom normalen Größenänderungswerkzeug, wenn es sich über den Größenänderungshandles der Array-Shell befindet, und der Cursor nimmt die Gestalt des normalen Symbols an, wenn Sie mit der Größenänderung beginnen. Die nachstehende Abbildung zeigt ein Beispiel.



Gewönl. Größenänderungswerkzeug



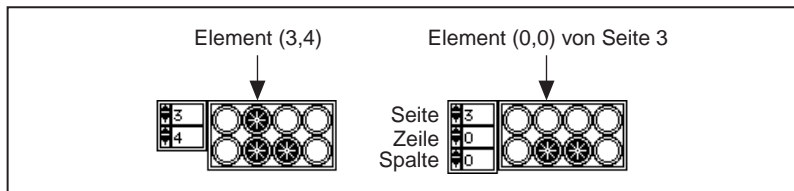
Die folgende Abbildung veranschaulicht das Vorgehen bei der vertikalen oder horizontalen Größenanpassung eines eindimensionalen Arrays zu dem Zweck, mehrere Elemente gleichzeitig anzuzeigen.



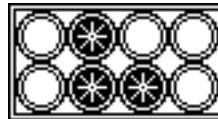
Die Indexdarstellung enthält den Index des Elements in der linken bzw. linken oberen Ecke der Tabelle. Durch Ändern des Indizes können Sie verschiedene Abschnitte eines großen oder multidimensionalen Arrays

darstellen. Bei eindimensionalen Arrays identifiziert der Index die Spalte des letzten links sichtbaren Elements. Bei zwei- oder höherdimensionalen Arrays identifizieren die unteren zwei Indizes die Koordinaten des oben links sichtbaren Elements. Im linken Abschnitt der folgenden Abbildung ist dieses Element [3,4].

Wenn Sie ein dreidimensionales Array als Buch mit Seiten betrachten, das aus Zeilen (auch Zeilen in diesem Sinne) von Zeichen (Spalten) besteht, stellt die Tabelle eine Seite teilweise oder ganz dar. In der nachstehenden Abbildung werden rechts die ersten vier Spalten der ersten beiden Zeilen von Seite 3 des Arrays gezeigt.



Wenn Sie die Arrayindizes verbergen wollen, rufen Sie das Popup-Menü des äußeren Rahmens auf, und schalten Sie den Menüpunkt **Anzeigen» Anzeige indexieren** aus dem Popup-Menü des Arrays aus. Die folgende Abbildung zeigt ein Beispiel dafür, wie ein tabellarisches Array ohne die Indexdarstellung aussieht.



Wenn Sie ein Array mit einer Ausrichtung darstellen wollen, die sich vom tabellarischen Arrayformat unterscheidet, stellen Sie die Elemente in einem Cluster auf dem Frontpanel dar, und verarbeiten Sie sie in einem Array. Verwenden Sie hierzu die Funktionen "Array in Cluster" und "Cluster in Array", die unter dem Thema **Online-Referenz»Funktion und VI Referenz»Cluster-Funktionen** erläutert werden.

Ausführen von Arrays

Ein Element des Arrays wird auf dieselbe Weise ausgeführt wie eines, das nicht Bestandteil des Arrays ist. Die Indexdarstellung wird auf dieselbe Weise wie ein digitales Bedienelement ausgeführt. Um den Wert eines Array-Bedienelements einzustellen, bringen Sie das Element mit der

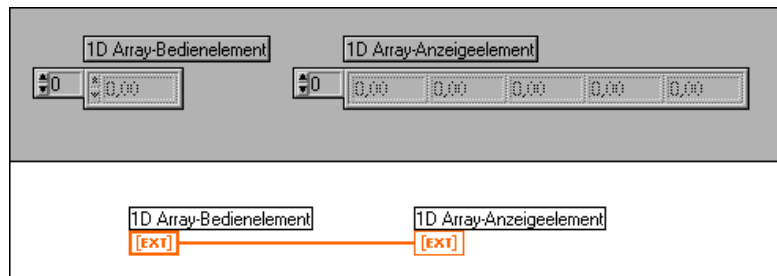
Indexdarstellung in die Ansicht, und stellen Sie dann den Wert des Elements ein.

Standardgrößen und -werte von Arrays

Sie können die Größe eines Arrays nicht durch eine feste Anzahl von Elementen einschränken. Wenn Sie jedoch die Standardwerte eines Array-Bedienelements einstellen, können Sie auch die Standardgröße einstellen. *Die Standardgröße sollte nicht größer als notwendig sein.* Wenn Sie ein Array auf eine große Standardgröße einstellen, werden neben dem VI alle Standarddaten gespeichert und somit die Größe Ihrer Datei erhöht.

Eine neue Array-Shell ohne Element ist *undefiniert*. Sie besitzt weder einen Datentyp noch Elemente und kann auch nicht in einem Programm verwendet werden. Sobald Sie ein Bedienelement ins Array einfügen, ist dieses ein *leeres Array* (der Länge 0) mit dem definierten Typ des eingefügten Bedienelements. Obgleich es keine definierten Elemente besitzt, können Sie es im VI verwenden. Das Array-Bedienelement markiert alle Elemente als nicht bedienbar, womit angezeigt wird, daß sie nicht definiert sind.

Die folgenden Abbildungen zeigen ein Array-Bedienelement in Einzelelementform, das mit einem Array-Anzeigeelement mit fünf Elementen in Tabellenform verbunden ist. In der ersten Abbildung sind beide Arrays leer (Elementwerte sind undefiniert).



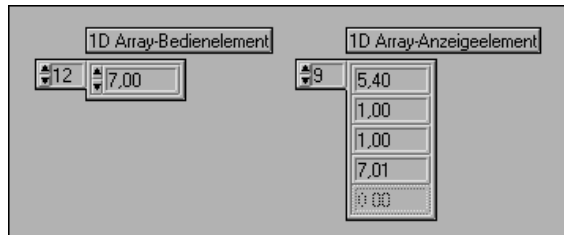
Wenn Sie die Indexanzeige des Array-Bedienelements auf 9 und den Elementwert beim Index 9 auf 5,40 einstellen, werden die Daten im Array auf 10 Elemente (0 bis 9) erweitert. Der Wert von Element 9 ist der von Ihnen eingestellte Wert, in diesem Beispiel 5,40. G weist den anderen Elementen den Standardwert des digitalen Bedienelements zu, nämlich 0,00. Wenn Sie nun **Aktuellen Wert als Standard** entweder aus der Shell oder aus dem Popup-Menü des Array-Bedienelements wählen, bestehen die Standarddaten des Arrays nunmehr aus den angegebenen

10 Elementen. Nach der Ausführung des VIs zeigt das Anzeigeelement dieselben Werte an.



Durch Wählen von **Rücksetzen auf Standard** für ein Element wird lediglich dieses Element auf seinen Standardwert zurückgesetzt.

Nun angenommen, Sie ändern den Standardwert eines numerischen Elements von 0,0 in 1,0. Ändern Sie hierzu den Wert im Anzeigeelement mit dem Bedienwerkzeug, rufen Sie dann das Popup-Menü des Elements auf, und wählen Sie **Aktuellen Wert als Standard**. Nehmen wir weiter an, Sie stellen den Wert von Element 12 auf 7,0 ein. Das Array besteht nun aus 13 Elementen. Die Werte der ersten 10 Elemente ändern sich nicht, aber die Elemente 10 und 11 haben den neuen Standardwert 1,0 und Element 12 hat den zugewiesenen Wert 7,0. Dieses Beispiel ist in der nachstehenden Abbildung gezeigt.



Wenn Sie nun aus den Popup-Menüs der Shells oder Indizes **Rücksetzen auf Standard** wählen, kehrt das Array zu dem im obigen Beispiel gezeigten 10-Element-Array zurück.

Um die Größe eines Arrays von $[N_i \text{ mal } N_j \text{ mal } N_k \dots]$ auf $[M_i \text{ mal } M_j \text{ mal } M_k \dots]$ zu erhöhen, weisen Sie dem neuen letzten Element $[M_i - 1, M_j - 1, M_k - 1, \dots]$ einen Wert zu. Um die Größe zu verringern, führen Sie zunächst den Befehl **Datenoperationen»Leeres Array** aus dem Popup-Menü des Indexarrays aus, stellen Sie dann das Array und die Werte auf die gewünschte Größe ein, oder wählen Sie das unerwünschte Subset des Arrays und anschließend den Menüpunkt **Datenoperationen»Daten ausschneiden** aus dem Popup-Menü. Nähere Einzelheiten über die verschiedenen Verfahren erhalten Sie in den folgenden Abschnitten.

Arrayelemente

Benutzen Sie im Ausführungsmodus die <Tab>-Taste, um den Fokus entweder zwischen Bedienelementen eines Frontpanels oder zwischen Elementen innerhalb eines einzelnen Arrays zu bewegen. Zunächst bewegt die <Tab>-Taste den Tastenfokus zwischen Bedienelementen des Frontpanels. Um den Fokus zwischen den Elementen innerhalb eines bestimmten Arrays zu bewegen, wechseln Sie zunächst mit Hilfe der <Tab>-Taste zum betreffenden Array. Benutzen Sie dann die <Strg>-Taste (**Windows**), <Befehlstaste> (**Macintosh**); <Meta>-Taste (**Sun**) oder <Alt>-Taste (**HP-UX**) und die Nach-unten-Pfeiltaste, um den Tastenfokus in das Array zu bewegen. Sie können nun die <Tab>-Taste benutzen, um die Arrayelemente und -indizes nacheinander durchzugehen. Um zur Bedienelementebene zurückzukehren, benutzen Sie die <Strg>-Taste (**Windows**), <Befehlstaste> (**Macintosh**), <Meta-Taste> (**Sun**) oder <Alt>-Taste (**HP-UX**) und den Nach-oben-Pfeil.

Feststellen der Größe von Arrays

Um die Größe eines Bedien- oder Anzeigeelements eines Arrays festzustellen, wählen Sie **Datenoperationen»Letztes Element anzeigen** aus dem Popup-Menü des Indexarrays.

Verschieben oder Ändern der Größe von Arrays

Verschieben Sie das Array stets durch Klicken auf den Shell-Rahmen oder die Indexdarstellung und Ziehen des Objekts. Wenn Sie das Element im Array ziehen, wird es vom Array getrennt, da die Shell und die Elemente nicht direkt miteinander verbunden sind. Wenn Sie versehentlich das Element ziehen, während Sie lediglich die Shell ziehen wollten, brechen Sie die Operation ab, indem Sie das Element wieder in die Shell ziehen oder über den Fensterrand des Frontpanels hinwegziehen, oder indem Sie vor Freigeben der Maustaste die <Esc>-Taste drücken. Eine Größenänderungsoperation kann dadurch abgebrochen werden, daß die Umrandung über das Fenster des Frontpanels hinaus gezogen wird, bevor die Maustaste freigegeben wird. Ist das Array Teil der aktuellen Auswahl, wird das Array durch Anfassen des Elements mitgezogen.

Der Array-Index kann von einer beliebigen Ecke aus vertikal oder von links aus horizontal in der Größe geändert werden.

Auswählen von Array-Zellen

Um Daten aus Arrays zu kopieren oder darin einzufügen, müssen Sie zunächst einen Bereich des Arrays auswählen. Gehen Sie zum Kopieren oder Einfügen folgendermaßen vor.

1. Um die Daten auszuwählen, stellen Sie den Array-Index auf das erste Element in dem Datensatz ein, den Sie kopieren wollen.
2. Wählen Sie dann den Menüpunkt **Datenoperationen»Auswahl beginnen** aus dem Popup-Menü des Arrays.
3. Als nächstes stellen Sie den Array-Index auf das letzte Element in dem Datensatz ein, den Sie kopieren wollen.
4. Wählen Sie dann den Menüpunkt **Datenoperationen»Auswahl beenden** aus dem Popup-Menü des Arrays.
5. Wählen Sie schließlich **Datenoperationen»Kopieren** oder **Datenoperationen»Einfügen**, um die gewählten Objekte zu kopieren oder einzufügen.

Diese Menüpunkte sind im Ausführungsmodus direkt verfügbar, wenn Sie das Popup-Menü eines Arrays aufrufen. Der Auswahlprozeß wird im folgenden Beispiel ausführlich erläutert.

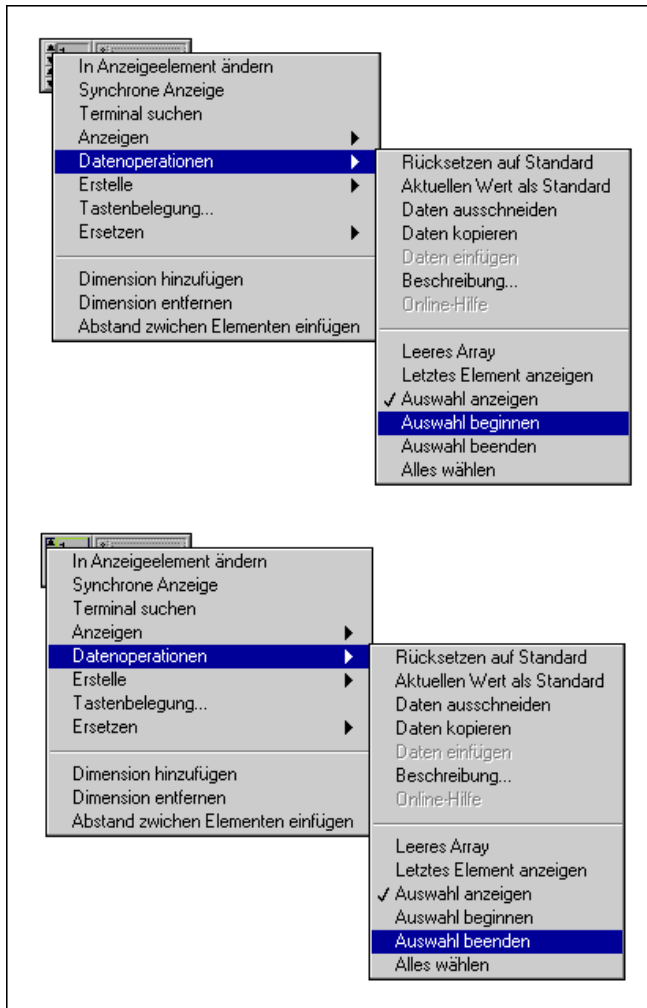
Ein Beispiel für die Auswahl von Arrayzellen

Beginnen Sie damit, daß Sie **Auswahl anzeigen** aus dem Popup-Menü wählen. Aktivieren Sie **Auswahl anzeigen** zur Anzeige sichtbar markierter Zellen.

Die gewählten Zellen sind von einem Rand umgeben. Wählen Sie **Datenoperationen»Abstand zwischen Elementen einfügen** aus dem Popup-Menü. Die Zellen werden somit durch einen dünnen freien Raum getrennt. Obgleich das Einfügen des Abstandes zwischen den Elementen nicht notwendig ist, ändert dieser offene Rand das Aussehen des Arrays und kann es womöglich verbessern. Wenn Sie Zellen auswählen, wird dieser Freiraum durch einen dicken Rand gefüllt, der die ausgewählten Zellen kennzeichnet. Wenn Sie keinen Elementabstand einfügen, erscheint der Auswahlrahmen über der Kante der ausgewählten Zelle.

Sie definieren die Zellen, die mit dem Array-Index ausgewählt werden sollen. Im folgenden Beispiel wird ein zweidimensionales Array verwendet.

Stellen Sie den Array-Index auf 1, 1 ein. Wählen Sie **Auswahl beginnen** aus dem Popup-Menü. Stellen Sie den Array-Index auf 3, 3 ein. Wählen Sie **Auswahl beenden** aus dem Popup-Menü.



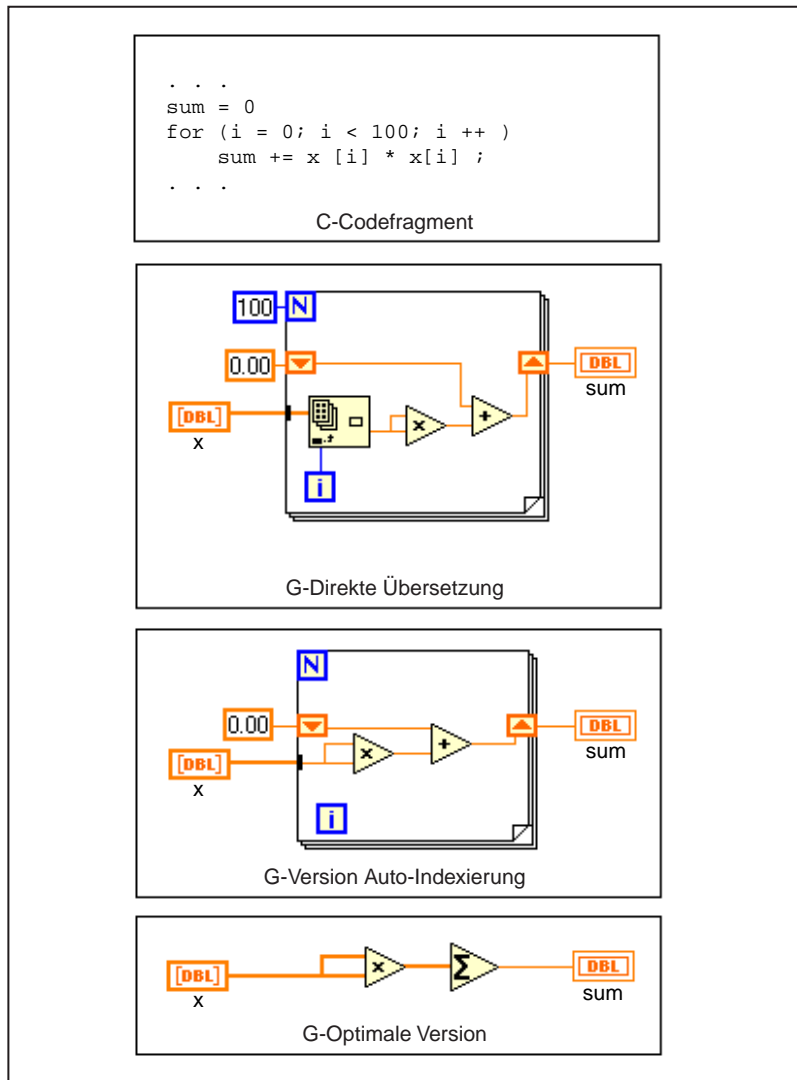
Die Auswahl umfaßt die durch die kleineren Indexziffern bezeichneten Zellen, nicht aber die durch die größeren Indexziffern bezeichneten Zellen. Wenn Sie die Zellen von einschließlich 1, 1 bis 3, 3 auswählen wollen, stellen Sie den Index auf 4, 4, bevor Sie **Auswahl beenden** auswählen. Beachten Sie, daß beim Ausschneiden der gewählten Objekte eigentlich der gesamte Satz von Zeilen und Spalten in der Auswahl entfernt wird, aber nur der gewählte Bereich in die Zwischenablage übertragen wird.

Nachdem Sie Zellen auswählen, fügen Sie die Daten durch Ausschneiden oder Kopieren in andere Zellen ein. Wenn Sie fertig sind, werden Sie feststellen, daß eine Rahmenlinie hervorgehoben bleibt. Dies ist ein Einfügepunkt, der im Array bestehen bleibt. Um diese Linie zu verbergen, deaktivieren Sie den Menüpunkt **Auswahl anzeigen** aus dem Popup-Menü. Sie können diese Linie beseitigen, ohne **Auswahl anzeigen** zu deaktivieren, indem Sie die Auswahl löschen. Stellen Sie den Index in allen Dimensionen auf 0, und wählen Sie dann **Auswahl beginnen** und **Auswahl beenden**. Die Auswahl erstreckt sich damit in allen Dimensionen von 0 bis 0, was dem Zustand der Auswahl entspricht, wenn sie anfangs auf das Frontpanel plaziert wird.

G-Arrays und Arrays in anderen Systemen

Die meisten Programmiersprachen verfügen über Arrays, obgleich nur wenige über die Anzahl von eingebauten Arrayfunktionen verfügen, die G besitzt. Andere Sprachen erschöpfen sich gewöhnlich in grundlegenden Array-Operationen, wie z.B. dem Extrahieren oder Ersetzen eines Elements, und überlassen es dem Benutzer, komplexere Operationen zu erstellen. G verfügt über diese Grundoperatoren, damit Sie ein Programm mit einer anderen Sprache direkt einem VI in G zuordnen können. Sie können jedoch in der Regel ein einfacheres und kleineres Diagramm erstellen, wenn Sie in G neu anfangen und die Arrayfunktionen höherer Ebenen verwenden. Das folgende Beispiel zeigt einen Ausschnitt eines C-Programms, das die Quadrate der Elemente eines Arrays addiert, die

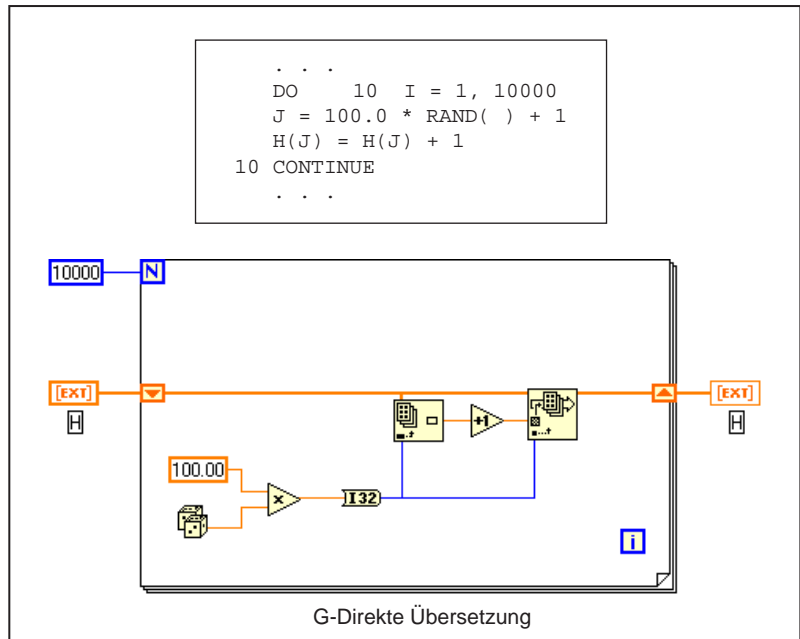
direkte Umsetzung in ein G-Blockdiagramm und zwei neu verfaßte Versionen, die wirkungsvoller als die Umsetzung sind.



Die Indizierungoperation zum Extrahieren eines Elements wird in C durch rechteckige Klammern dargestellt [], die dem Arraynamen folgen und den Index einschließen. In G wird diese Operation durch das Icon der Funktion "Index-Array" dargestellt, das hier links zu sehen ist. Das Array ist mit dem linken oberen Terminal verbunden, der Index ist mit dem linken

unteren Terminal verbunden, und der Wert des Arrayelements ist mit dem rechten Terminal verbunden.

Das folgende Beispiel zeigt die grundlegende Operation des Ersetzens eines Arrayelements. Der Ausschnitt des FORTRAN-Codes generiert ebenso wie die G-Version ein Histogramm von Zufallszahlen (der Generator von Zufallszahlen erzeugt Werte zwischen 0 und 1). Ignorieren Sie die Operation +1 in der FORTRAN-Berechnung von J; diese Addition ist deshalb notwendig, weil die Indizierung in FORTRAN-Arrays nicht bei Null, sondern bei 1 anfängt.



Die Indizierungsoperation zum Einfügen oder Ersetzen eines Elements wird in FORTRAN durch Klammern () auf der linken Seite des Gleichheitszeichens dargestellt, die dem Arraynamen folgen und den Index einschließen. In G wird diese Operation durch das Icon der Funktion “Arrayelement ersetzen” dargestellt (siehe Abbildung links). Das Array ist mit dem linken oberen Terminal verbunden, der neue Wert ist mit dem linken mittleren Terminal verbunden, der Index ist mit dem linken unteren Terminal verbunden, und das aktualisierte Array ist vom rechten Terminal aus verbunden.



Arrayelement ersetzen (Funktion)

In den meisten Programmiersprachen müssen Sie ein Array zuerst deklarieren und in manchen Fällen sogar initialisieren, bevor Sie es

verwenden können. In G entspricht die Erstellung eines Bedien- oder Anzeigeelements vom Typ 'Array' der Deklaration eines Arrays, und die Definition seines Standardwerts entspricht seiner Initialisierung.

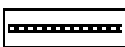
Die meisten Sprachen erfordern, daß Sie als Bestandteil der Deklaration eine maximale Größe für ein Array angeben. Die Information über die Größe eines Arrays wird nicht als integraler Bestandteil des Arrays behandelt, was bedeutet, daß Sie die Größe selbst im Auge behalten müssen. Bei G ist es nicht notwendig, die maximale Größe für Arrays zu deklarieren, da G die Größeninformation zu dem Array automatisch aufbewahrt. Ferner können Sie bei G keinen Wert außerhalb der Grenzen eines Arrays speichern. In vielen herkömmlichen Programmiersprachen gibt es keinen solchen Prüfmechanismus, wodurch Sie den Speicher versehentlich zerstören können.

Cluster

Ein G-Cluster ist eine geordnete Sammlung eines oder mehrerer Elemente und ist Strukturen in C und anderen Sprachen ähnlich. Im Gegensatz zu Arrays können Cluster eine beliebige Kombination von G-Datentypen enthalten. Obgleich sowohl Cluster- als auch Arrayelemente geordnet sind, erfolgt der Zugriff auf Cluster-Elemente statt durch Indizierung einzelner Elemente durch *Aufschlüsseln* aller Elemente auf einmal. Cluster unterscheiden sich auch insofern von Arrays, als sie eine feste Größe aufweisen. Ebenso wie ein Array ist ein Cluster entweder ein Bedien- oder Anzeigeelement; ein Cluster kann nicht aus einer Kombination von Bedien- und Anzeigeelementen bestehen.

Sie können Cluster zum Gruppieren von in Beziehung stehenden Datenelementen verwenden, die im Diagramm an mehreren Stellen erscheinen, was die Übersichtlichkeit der Verbindungen erhöht und die Anzahl der von SubVIs benötigten Anschlußterminals senkt.

Die meisten Cluster im Blockdiagramm weisen ein allgemeines Verbindungsmuster auf, obgleich Cluster aus Zahlen (die manchmal als *Punkte* bezeichnet werden) ein spezielles Verbindungsmuster aufweisen.



Allgemeines Verbindungsmuster für Cluster



Verbindungsmuster für Cluster aus Zahlen

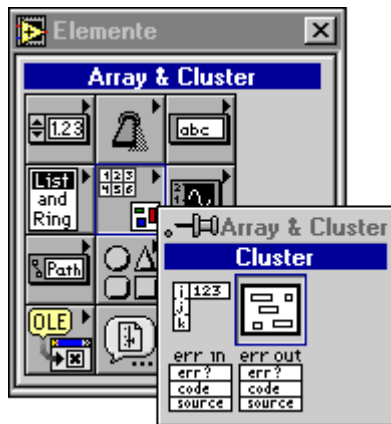
Sie können Terminals nur dann miteinander verbinden, wenn sie denselben Typ aufweisen. Bei Clustern bedeutet dies, daß beide Cluster dieselbe Anzahl von Elementen aufweisen müssen und die Typen einander entsprechender Elemente — durch die Cluster-Anordnung bestimmt — übereinstimmen müssen. G wandelt unterschiedliche Darstellungen von Zahlen in denselben Typ um.

Die Cluster-Anordnung ist eine numerische Reihenfolge, die Elementen in einem Cluster zugewiesen wird, wie im Abschnitt *Reihenfolge von Cluster-Elementen* beschrieben.

Informationen über Funktionen, die Cluster manipulieren, finden Sie unter dem Thema **Online-Referenz»Funktion und VI Referenz»Cluster-Funktionen**.

Erstellen von Clustern

Sie erstellen ein Bedien- oder Anzeigeelement vom Typ “Cluster” durch Installieren einer beliebigen Kombination von Booleschen Elementen, Strings, Diagrammen, Graph-Skalaren, Arrays oder sogar anderen Clustern in einer *Cluster-Shell*. Zugriff auf die Cluster-Shell besteht über die Palette **Elemente** (siehe nachstehende Abbildung).



Eine neue Cluster-Shell besitzt einen in der Größe veränderbaren Rahmen und ein optionales Label.

Wenn Sie im freien Elementbereich das Popup-Menü aufrufen, erscheint die Palette **Elemente**. Sie können ein beliebiges Element aus der Palette **Elemente** in einen Cluster plazieren. Sie können bestehende Objekte in die Cluster-Shell ziehen oder dort an Ort und Stelle erstellen und ablegen, indem Sie sie aus dem Popup-Menü des Bedienelements auswählen und in die Cluster-Shell ziehen. Das Cluster nimmt ebenso wie später hinzugefügte Objekte die Datenrichtung (Bedien- oder Anzeigeelement) desjenigen Elements an, das als erstes in den Cluster plaziert wird. Durch Auswählen von **In Bedienelement ändern** oder **In Anzeigeelement ändern** aus dem Popup-Menü oder eines beliebigen anderen Cluster-Elements werden der Cluster und alle seine Elemente von Anzeige- in Bedienelemente bzw. von Bedien- in Anzeigeelemente umgewandelt.

Ausführen und Konfigurieren von Cluster-Elementen

Mit Ausnahme der Einstellung von Standardwerten und der Menüpunkte **In Bedienelement ändern** und **In Anzeigeelement ändern** erfolgt die Konfiguration von Bedien- und Anzeigeelementen auf dieselbe Weise wie bei Bedien- und Anzeigeelementen, die nicht in einem Cluster angeordnet sind.

Cluster-Elemente

Im Ausführungsmodus können Sie die <Tab>-Taste benutzen, um den Fokus entweder zwischen Bedienelementen eines Frontpanels oder zwischen Elementen innerhalb eines einzelnen Clusters zu bewegen. Zunächst bewegt die <Tab>-Taste den Fokus zwischen Bedienelementen des Frontpanels. Um den Fokus zwischen den Elementen innerhalb eines bestimmten Clusters zu bewegen, wechseln Sie zunächst mit Hilfe der <Tab>-Taste zum betreffenden Cluster. Benutzen Sie dann die <Strg>-Taste (**Windows**), <Befehlstaste> (**Macintosh**); <Meta>-Taste (**Sun**) oder <Alt>-Taste (**HP-UX**) und die Nach-unten-Pfeiltaste, um sich zwischen den Cluster-Elementen hin- und herzubewegen. Um zur Bedienelementebene zurückzukehren, benutzen Sie die <Strg>-Taste (**Windows**), <Befehlstaste> (**Macintosh**), <Meta>-Taste (**Sun**) oder <Alt>-Taste (**HP-UX**) und den Nach-oben-Pfeil.

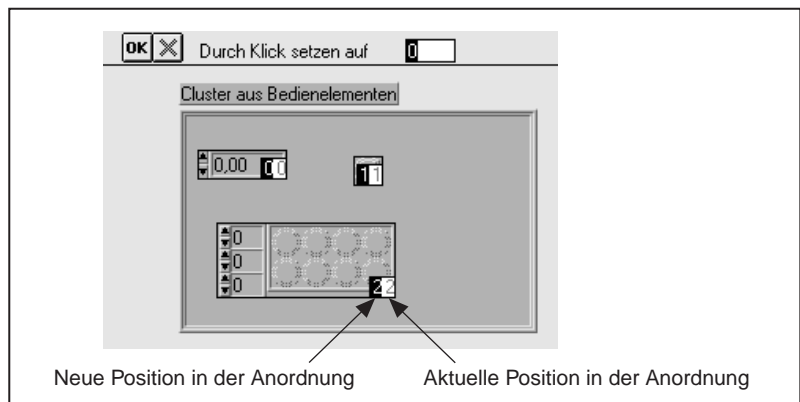
Standardwerte von Clustern

Um den Standardwert des gesamten Clusters auf den aktuellen Wert jedes einzelnen Elements einzustellen, rufen Sie das Popup-Menü des Clusterrahmens auf, und wählen Sie den Befehl **Datenoperationen»Aktuellen Wert als Standard** aus dem Popup-Menü des Clusters. Um alle Elemente auf ihre individuellen Standardkonfigurationen zurückzusetzen, wählen Sie den Befehl **Datenoperationen»Rücksetzen auf Standard** aus dem Popup-Menü des Clusters.

Um den Standardwert eines einzelnen Elements in einem Cluster zu ändern, rufen Sie das Popup-Menü des betreffenden Elements auf, und wählen Sie **Aktuellen Wert als Standard**.

Reihenfolge von Cluster-Elementen

Cluster-Elemente haben eine logische Reihenfolge, die nicht mit ihrer Position innerhalb der Shell in Beziehung steht. Das erste Objekt, das Sie ins Cluster einfügen, ist Element 0, das zweite Element 1 und so weiter. Wenn Sie ein Element löschen, wird die Reihenfolge automatisch dementsprechend angepaßt. Sie können die aktuelle Reihenfolge durch Wählen des Menüpunkts **Cluster-Anordnung...** aus dem Popup-Menü des Clusters ändern. Das Aussehen des Elements ändert sich entsprechend der nachstehenden Abbildung.





Enter-Taste



X-Taste

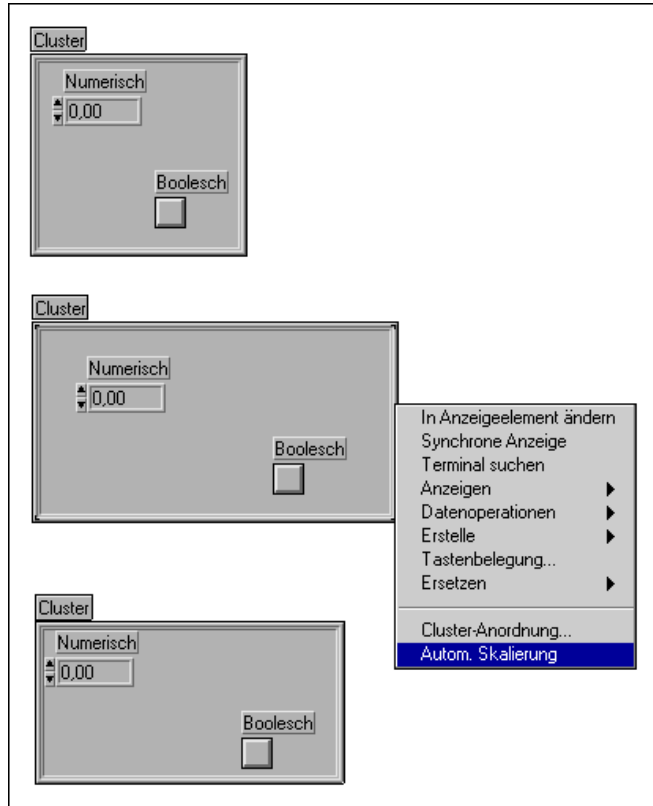
Die weißen Felder auf den Elementen zeigen die aktuellen Positionen in der Cluster-Anordnung an. Die schwarzen Felder zeigen die neue Position in der Anordnung an. Durch Klicken auf ein Element mit dem Cursor für die Cluster-Anordnung wird die Position des Elements in der Cluster-Anordnung auf die in der Symbolleiste angezeigte Zahl eingestellt. Sie ändern diese Anordnung, indem Sie eine neue Zahl in das Feld eingeben und auf das betreffende Bedienelement klicken. Wenn die Anordnung Ihren Wünschen entspricht, klicken Sie auf die Enter-Schaltfläche, um die Einstellung vorzunehmen und den Bearbeitungsmodus für die Cluster-Anordnung zu beenden. Klicken Sie auf die X-Schaltfläche, um die alte Anordnung wiederherzustellen.

Die Cluster-Anordnung bestimmt die Reihenfolge, in der die Elemente als Terminals in den Funktionen Bündeln und Aufschlüsseln im Blockdiagramm erscheinen. Weitere Informationen über diese Funktionen finden Sie unter dem Thema **Online-Referenz» Funktion und VI Referenz» Cluster-Funktionen**.

Verschieben und Ändern der Größe von Clustern

Clusterelemente sind keine festen Komponenten von Clustern. Das bedeutet, daß Sie Elemente unabhängig verschieben können, auch wenn sich diese in der Shell befinden. Um zu vermeiden, daß sie versehentlich aus der Shell gezogen werden, klicken Sie auf die Shell, wenn Sie einen Cluster verschieben wollen, und ändern Sie die Größe von Clustern über den Shell-Rahmen. Wenn Sie versehentlich das Element ziehen, während Sie den ganzen Cluster ziehen wollten, können Sie die Operation abbrechen, indem Sie das Element wieder in die Shell ziehen oder über den Fensterrand des Frontpanels hinwegziehen, oder indem Sie vor Freigeben der Maustaste die <ESC>-Taste drücken. Eine Größenänderungsoperation kann auch dadurch abgebrochen werden, daß die Umrandung vor Freigeben der Maustaste über das Fenster des Frontpanels hinaus gezogen wird, oder durch Betätigen der <ESC>-Taste vor Freigeben der Maustaste.

Sie können die Größe von Clustern durch Wählen von **Autom. Skalierung** aus dem Popup-Menü an ihren Inhalt anpassen (siehe nachstehende Abbildung).



Zusammensetzen von Clustern

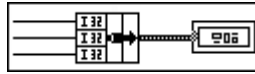
G verfügt über drei Funktionen zum Zusammensetzen oder Erstellen von Clustern. Die Funktionen “Bündeln” und “Nach Namen bündeln” setzen einen Cluster aus einzelnen Elementen zusammen oder ersetzen einzelne Elemente mit Elementen desselben Typs. Die Funktion “Array in Cluster” konvertiert ein Array aus Elementen in einen Cluster aus Elementen.

Die Funktion “Bündeln”

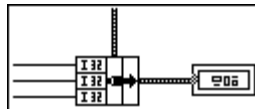


Die Funktion “Bündeln”, die Sie über die Palette **Array & Cluster** der Palette **Funktionen** erhalten, erscheint im Blockdiagramm mit zwei Elementeingabeterminals auf der linken Seite. Ändern Sie die Größe des Icons vertikal, um so viele Terminals wie nötig zu erstellen. Das Element, das Sie mit dem oberen Terminal verbinden, wird im Ausgabecluster zu Element 0, das Element, das Sie mit dem zweiten Terminal verbinden, wird im Ausgabecluster zu Element 1 usw.

Für jedes Element, das Sie mit einem Eingangsterminal verbinden, erscheint am zuvor leeren Terminal ein Symbol, das den Datentyp des verbundenen Elements darstellt (siehe nachstehende Abbildung). Sie müssen alle Eingänge, die Sie erstellen, verbinden.



Zusätzlich zu Eingangsterminals für Elemente auf der linken Seite verfügt die Funktion “Bündeln” über ein Eingangsterminal für Cluster in der Mitte (siehe nachstehende Abbildung). Mit diesem Terminal können ein oder mehrere Elemente einer bestehenden Clusterverbindung ohne Auswirkung auf die anderen Elemente ersetzt werden.



Im Abschnitt *Ersetzen von Cluster-Elementen* in diesem Kapitel finden Sie ein Anwendungsbeispiel dieser Funktion zum Ersetzen von Elementen in einem Cluster.

Die Funktion “Nach Namen bündeln”



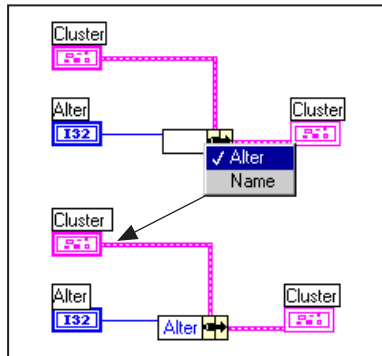
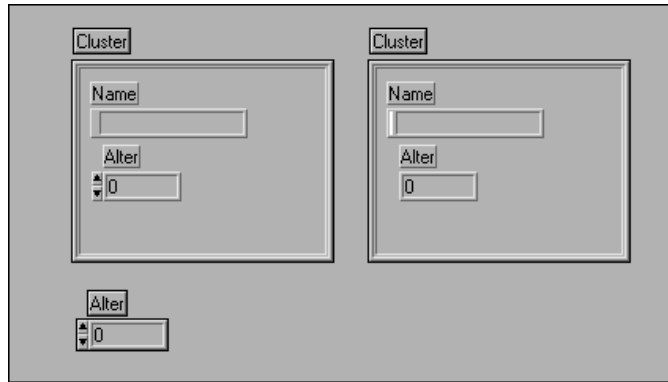
Nach Namen bündeln

Die Funktion “Nach Namen bündeln”, die ebenfalls in der Palette **Funktionen»Cluster** zu finden ist, ist der Funktion “Bündeln” ähnlich. Anstatt jedoch auf Felder nach der Position Bezug zu nehmen, erfolgt die Bezugnahme über den Namen. Im Gegensatz zur Funktion “Bündeln” können Sie nur die Elemente anzeigen, auf die Sie zugreifen wollen. Für jedes Element, auf das Sie zugreifen wollen, müssen Sie einen Eingang für die Funktion hinzufügen.

Da der Name keine Position innerhalb des Clusters bezeichnet, müssen Sie einen Cluster ebenfalls mit dem mittleren Terminal verbinden. Sie können mit der Funktion “Nach Namen bündeln” nur Elemente nach dem Namen ersetzen; Sie können damit keinen neuen Cluster erstellen. Sie könnten jedoch einen Datentyp-Cluster mit dem mittleren Terminal verbinden und dann alle neuen Werte als Namen verbinden, um dieselbe Verhaltensweise zu erzeugen.

Nehmen wir z.B. an, ein Cluster beinhaltet einen String namens “Name” und eine Zahl namens “Alter”. Nach dem Platzieren der Funktion “Nach Namen bündeln” müssen Sie zunächst einen Eingabecluster mit dem mittleren Terminal der Funktion “Nach Namen bündeln” verbinden.

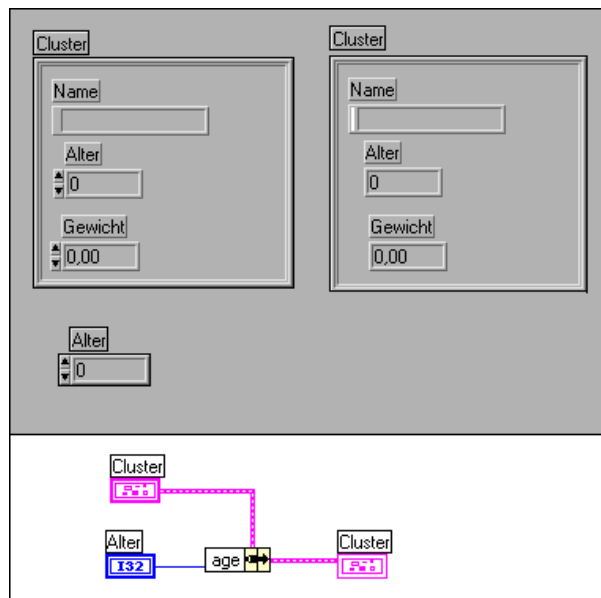
Nachdem Sie das linke Terminal verbunden haben, wählen Sie die zu modifizierenden Elemente, indem Sie das Pop-up-Menü eines der linken Terminals der Funktion “Nach Namen bündeln” aufrufen. Sie werden eine Liste mit den Namen der Elemente aus dem Quellcluster sehen. Sobald Sie einen Namen auswählen, können Sie einen neuen Wert mit diesem Terminal verbinden, um dem betreffenden Wert des Clusters einen neuen Wert zuzuweisen. Die nachstehenden Abbildungen veranschaulichen dies anhand eines Beispiels.



Ändern Sie die Größe der Funktion “Nach Namen bündeln”, um so viele oder so wenige Elemente wie nötig anzuzeigen. Dies unterscheidet sich von der Funktion “Bündeln”, bei der die Größenänderung dieselbe Anzahl von Elementen wie der resultierende Cluster ergeben muß.

Wenn der Cluster einen weiteren Cluster enthält und Sie das Pop-up-Menü eines einzelnen Elements der Funktion “Nach Namen bündeln” aufrufen, erhalten Sie ein seitlich ausziehbares Untermenü, **Objekte wählen**, mit dem Sie auf einzelne Elemente des eingebetteten Clusters nach dem Namen zugreifen oder alle Elemente auswählen können.

Die Funktion “Nach Namen bündeln” ist nützlich, wenn Sie mit Datenstrukturen arbeiten, die sich höchstwahrscheinlich im Verlauf des Entwicklungsprozesses ändern werden. Umfaßt die Modifikation das Hinzufügen eines neuen Elements oder die Neuordnung des Clusters, ist hierzu nicht die Änderung der Funktion “Nach Namen bündeln” erforderlich, da die Namen weiterhin gültig sind. Wenn Sie beispielsweise das vorherige Beispiel abändern, indem Sie dem Quell- und Zielcluster das neue Element und Gewicht hinzufügen, ist das VI noch richtig. Die nachstehende Abbildung veranschaulicht dieses Beispiel.



Die Funktion “Nach Namen bündeln” ist insbesondere in umfangreicheren Anwendungen in Verbindung mit Typendefinitionen nützlich. Wird ein Cluster, der sich wahrscheinlich ändert, als Typendefinition definiert, kann jedes VI, das dieses Bedienelement enthält, geändert werden, um einen neuen Datentyp zu reflektieren, indem die Datei der Typendefinition aktualisiert wird. Wenn die VIs nur “Nach Namen bündeln” und “Nach Namen aufschlüsseln” verwenden, um auf die Elemente des Clusters zuzugreifen, können Sie die Einführung von Fehlern in den VIs vermeiden, die den Cluster verwenden.

Weitere Informationen über Typendefinitionen finden Sie im Abschnitt *Typendefinitionen* in Kapitel 24, *Benutzerspezifische Bedienelemente und Typendefinitionen*. Informationen finden Sie auch im Abschnitt *Die Funktion “Nach Namen aufschlüsseln”* in diesem Kapitel.



Array in Cluster

Die Funktion “Array in Cluster”

Die Funktion “Array in Cluster”, die Sie aus der Palette **Cluster** oder **Array** der Palette **Funktionen** wählen, konvertiert die Elemente eines eindimensionalen Arrays in Elemente eines Clusters. Diese Funktion ist nützlich, wenn Sie Elemente desselben Typs innerhalb eines Cluster-Anzeigeelements eines Frontpanels anzeigen wollen, aber die Elemente auch nach dem Indexwert im Blockdiagramm manipulieren wollen.

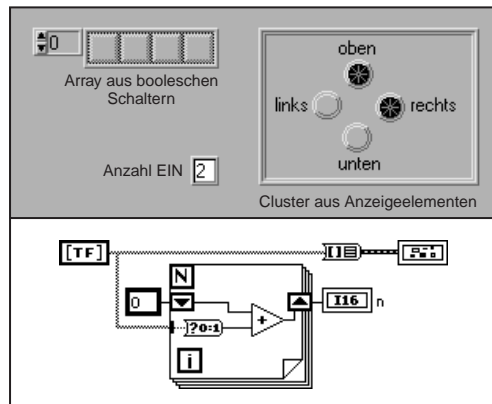
Benutzen Sie den Menüpunkt **Cluster-Größe...** aus dem Pop-up-Menü der Funktion, um die Anzahl der Elemente im Cluster zu konfigurieren. Das daraufhin angezeigte Dialogfeld wird in der nachstehenden Abbildung gezeigt.



Die Funktion weist dem Null-Arrayelement das Null-Cluster-Element zu usw. Enthält das Array mehr Elemente als der Cluster, ignoriert die Funktion die übrigen Elemente. Wenn das Array weniger Elemente als der Cluster enthält, weist die Funktion den zusätzlichen Cluster-Elementen den Standardwert für den Elementdatentyp zu.

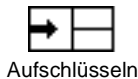
Die folgende Abbildung zeigt, wie ein Array aus Daten in einem Cluster-Anzeigeelement eines Frontpanels dargestellt wird. Mit diesem Verfahren können Sie die Arrayelemente auf dem Frontpanel für Ihre Anwendung organisieren. Sie könnten statt dessen ein tabellarisches Array-Anzeigeelement verwenden, dies beschränkt Sie jedoch auf ein festes Darstellungsformat — horizontal mit dem niedrigsten Element links und

einer angehängten Indexanzeige. Die gezeigte Schleife zählt lediglich die Anzahl der Elemente im Array, die auf TRUE eingestellt sind.



Zerlegen von Clustern

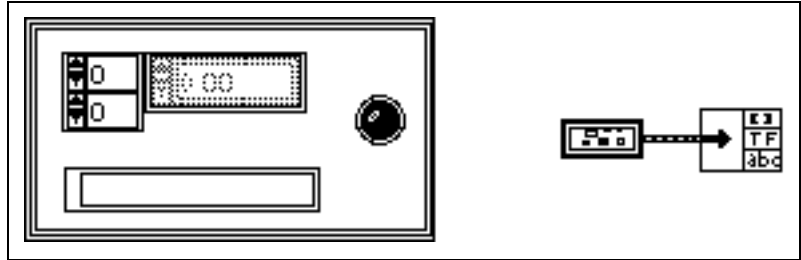
G verfügt über drei Funktionen zum Zerlegen von Clustern. Die Funktionen “Aufschlüsseln” und “Nach Namen aufschlüsseln” zerlegen einen Cluster in einzelne Elemente. Die Funktion “Cluster in Array” konvertiert Elemente identischen Typs in ein Array des betreffenden Typs.



Die Funktion “Aufschlüsseln”

Die Funktion “Aufschlüsseln”, die Sie über die Palette **Array & Cluster** der Palette **Funktionen** erhalten, verfügt über zwei Element-Ausgangsterminals auf der rechten Seite. Sie passen die Anzahl der Terminals mit dem Größenänderungswerkzeug in derselben Weise an, in der Sie die Funktion “Bündeln” anpassen. Element 0 in der Cluster-Anordnung wird an das obere Ausgangsterminal übergeben, Element 1 wird an das zweite Terminal übergeben usw.

Die Clusterverbindung bleibt solange unterbrochen, bis Sie die richtige Anzahl von Ausgangsterminals erstellen. Sobald die Anzahl der Terminals stimmt, ändert sich die Verbindung in eine Vollinie. Die Terminalsymbole stellen die Elementdatentypen dar (siehe nachstehende Abbildung).



Die Funktion “Nach Namen aufschlüsseln”



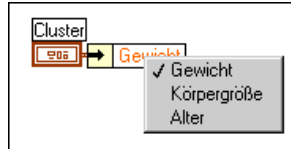
Nach Namen aufschlüsseln

Die Funktion “Nach Namen aufschlüsseln”, die ebenfalls in der Palette **Funktionen»Array & Cluster** zu finden ist, ist der Funktion “Aufschlüsseln” ähnlich. Anstatt jedoch auf Felder nach der Position Bezug zu nehmen, erfolgt die Bezugnahme über den Namen. Im Gegensatz zur Funktion “Aufschlüsseln” können Sie nur die Elemente anzeigen, die Sie lesen wollen. Es muß nicht für jedes Element ein Ausgang gegeben sein.

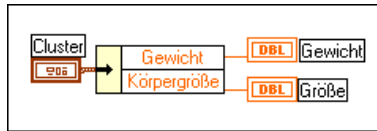
Die folgende Abbildung zeigt beispielsweise ein Panel mit einem Cluster aus den drei Elementen Gewicht, Größe und Alter.



Wenn Sie die Funktion “Nach Namen aufschlüsseln” verbinden, können Sie das Popup-Menü eines Ausgangsterminals der Funktion aufrufen und den Menüpunkt **Objekt wählen** wählen, um ein Element auszuwählen, das Sie aus den Komponentennamen des Clusters lesen wollen (siehe nachstehende Abbildung).

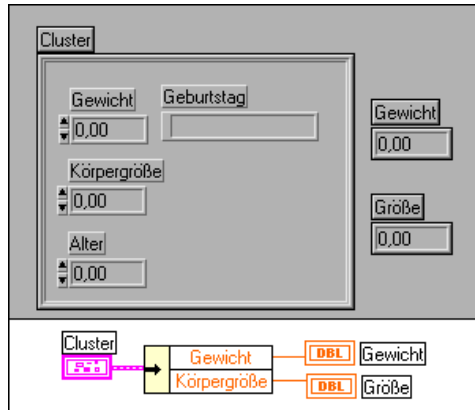


Ändern Sie die Größe der Funktion zum Lesen anderer Elemente (siehe nachstehende Abbildung). Beachten Sie, daß Sie nicht alle Elemente lesen müssen, und daß Sie sie in beliebiger Reihenfolge lesen können.



Einer der Vorteile der Funktion “Nach Namen aufschlüsseln” gegenüber der Funktion “Aufschlüsseln” besteht darin, daß sie nicht so eng mit der Datenstruktur des Clusters verbunden ist. Die Funktion “Bündeln” muß stets genauso viele Terminals wie Elemente aufweisen. Wenn Sie ein Element in einem Cluster hinzufügen oder entfernen, der mit der Funktion “Aufschlüsseln” verbunden ist, erhalten Sie dadurch unterbrochene Verbindungen. Mit der Funktion “Nach Namen aufschlüsseln” können Sie Elemente zum Cluster hinzufügen oder daraus entfernen, ohne die VI-Verbindungen zu trennen, solange nicht vom Diagramm aus darauf Bezug genommen wurde.

Wenn Sie im Beispiel des vorherigen Clusters einen String für Geburtstag hinzufügen, ist das Diagramm noch richtig. Dieses Beispiel wird in der nachstehenden Abbildung gezeigt.



Ebenso wie die Funktion “Nach Namen bündeln” ist die Funktion “Nach Namen aufschlüsseln” insbesondere bei umfangreicheren Anwendungen nützlich in Verbindung mit Typendefinitionen. Weitere Informationen hierzu finden Sie im Abschnitt *Die Funktion “Nach Namen bündeln”* in diesem Kapitel und im Abschnitt *Typendefinitionen* in Kapitel 24, *Benutzerspezifische Bedienelemente und Typendefinitionen*.

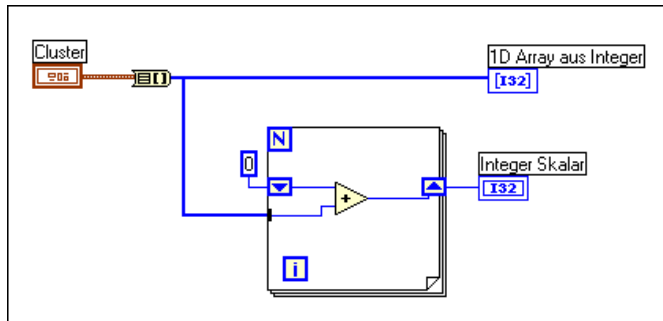
Die Funktion “Cluster in Array”



Die Funktion “Cluster in Array”, die Sie über die Palette **Array** der Palette **Funktionen** erhalten, konvertiert die Elemente eines Clusters in ein eindimensionales Array derselben Elemente. Die Cluster-Elemente müssen denselben Typ aufweisen. Das Array enthält dieselbe Anzahl von Elementen wie der Cluster.

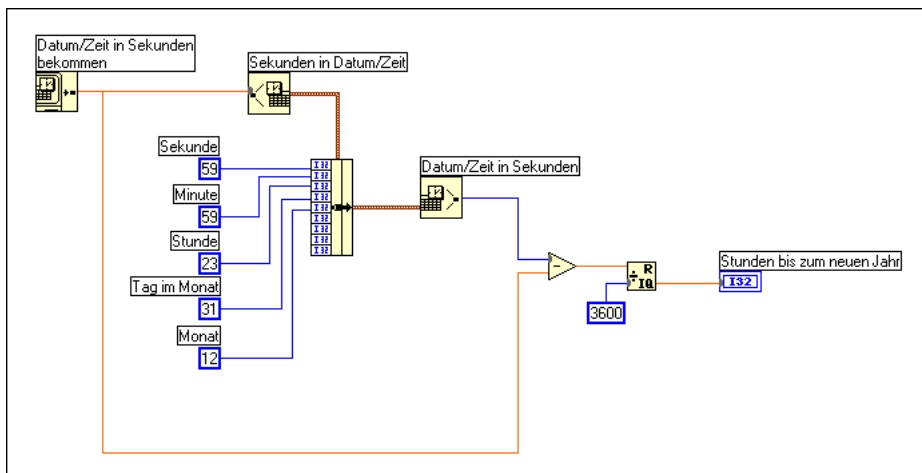
Die Funktion “Cluster in Array” ist nützlich, wenn Sie einen Satz von Bedienelementen eines Frontpanels innerhalb des Clusters in einer

bestimmten Reihenfolge gruppieren wollen, sie im Blockdiagramm aber als Array bearbeiten wollen (siehe nachstehende Abbildung).



Ersetzen von Cluster-Elementen

Sie werden manchmal den Wert eines oder zweier Elemente eines Clusters ersetzen oder ändern wollen, ohne daß sich dies auf andere Elemente auswirkt. Hierzu müssen Sie ein Cluster aufschlüsseln und die unveränderten Elemente samt den Ersatzwerten des anderen Elements direkt mit einer Bündelfunktion verbinden. Das folgende Beispiel veranschaulicht eine bequemere Methode. Mit Hilfe des Datum/Zeit-Clusters berechnet es die Anzahl der Stunden bis zum neuen Jahr. Da das Cluster-Eingangsterminal (mittleres Terminal) der Funktion “Bündeln” verbunden ist, müssen lediglich die Element-Eingangsterminals verbunden werden, deren Werte ersetzt werden.

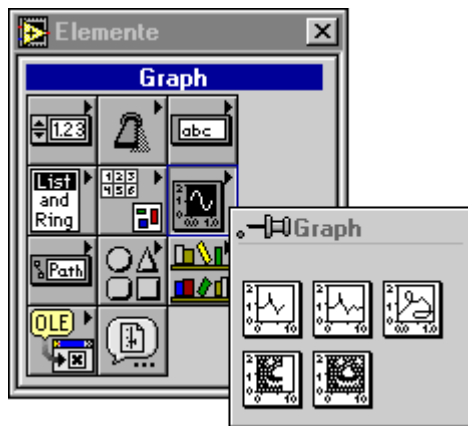


Bedien- und Anzeigeelemente vom Typ Graph und Diagramm

Dieses Kapitel beschreibt die Erstellung und Verwendung von Anzeigeelementen in Form von Graphen und Diagrammen mit Hilfe der Palette **Elemente**»Graph.

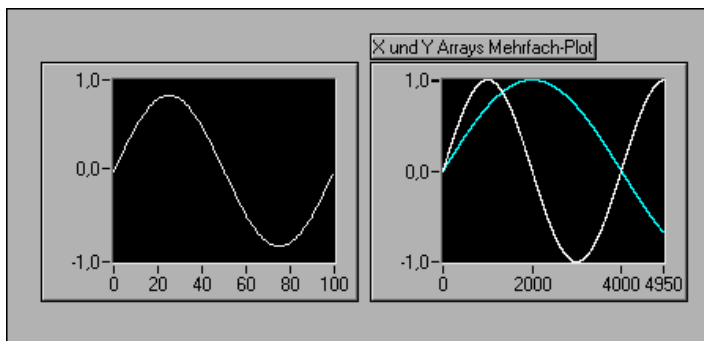
Ein Anzeigeelement vom Typ Graph ist eine zweidimensionale (2D) Darstellung eines oder mehrerer *Plots*. Der Graph empfängt und plottet Daten als Block. Ein Diagramm stellt ebenfalls Plots dar, aber es empfängt die Daten und aktualisiert die Darstellung Punkt für Punkt oder Array für Array, wobei eine bestimmte Anzahl von abgearbeiteten Punkten zu Darstellungszwecken in einem Puffer gespeichert werden. Zusätzlich zu den Informationen in diesem Kapitel finden Sie es möglicherweise hilfreich, sich die Graphen- und Diagrammbeispiele anzusehen. Diese Beispiele sind unter `examples\general\graphs` zu finden.

G verfügt über drei Arten von Graphen und zwei Arten von Diagrammen. Sie sind in der nachstehenden Abbildung zu sehen, und zwar oben von links nach rechts — Kurvendiagramm, Kurvengraph, *xy*-Graph, Intensitätsdiagramm und Intensitätsgraph.



Erstellen von Kurven- und xy-Graphen

Kurvengraphen dienen zur Darstellung von gleichmäßig abgetasteten Maßen. *xy-Graphen* können einen beliebigen Satz von Punkten darstellen, ob gleichmäßig abgetastet oder nicht. Sie erhalten einen Kurvengraphen und *xy-Graphen* aus der Palette **Elemente»Graph**. Beispiele dieser Graphen sind in der nachstehenden Abbildung gezeigt.



Der Kurvengraph plottet nur Einzelwertfunktionen mit Punkten, die entlang der x -Achse verteilt sind, wie z.B. erfasste zeitabhängige Kurvenformen. Der xy -Graph ist ein kartesisches Allzweck-Graphierungsobjekt, mit dem Sie mehrwertige Funktionen wie z.B. kreisförmige Vielecke oder Kurenformen mit veränderlicher Zeitbasis plotten können. Beide Graphen können eine beliebige Anzahl von Plots darstellen.

Jeder dieser Graphen nimmt mehrere Datentypen an. Dies reduziert das Maß, zu dem Sie die Daten vor der Darstellung manipulieren müssen, auf ein Minimum. Diese Datentypen werden im nächsten Abschnitt beschrieben.

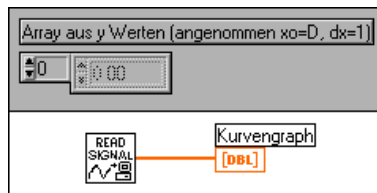
Dem Abschnitt über Datentypen für Graphen folgen Abschnitte, die einige komplexere Optionen des Graphen beschreiben. Hierzu gehören u.a. die Anpassung der Erscheinungsform des Graphen, die Darstellung und Manipulation von Cursors und die Darstellung einer Legende für den Graphen.

Plotten von Einzelplot-Graphen

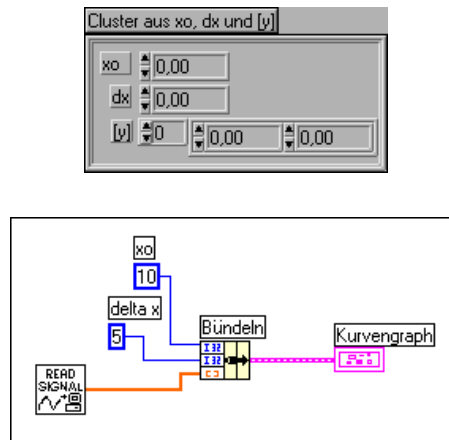
Datentypen von Kurvengraphen

Für Einzelplot-Graphen nimmt der Kurvengraph zwei Datentypen an. Diese Datentypen werden in den nachfolgenden Absätzen beschrieben.

Der erste Datentyp, den der Kurvengraph annimmt, ist ein Einzelarray aus Werten. Der Graph interpretiert die Daten als Punkte und erhöht sie, angefangen bei $x = 0$, um eins. Das folgende Diagramm veranschaulicht das Vorgehen beim Erstellen derartiger Daten.



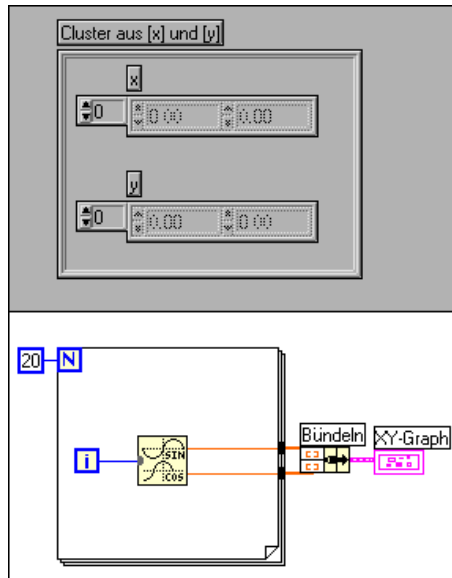
Der zweite Datentyp ist ein Cluster aus einem x -Ausgangswert, einem Δx -Wert und einem Array aus y -Daten. Das folgende Diagramm veranschaulicht das Vorgehen beim Erstellen derartiger Daten.



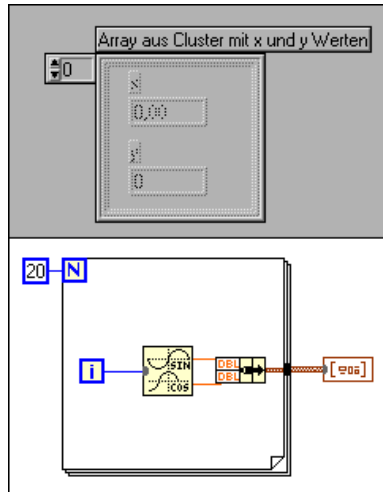
Datentypen für xy-Graphen

Der xy -Graph nimmt für einen Einzelplot-Graphen zwei Datentypen an. Diese Datentypen werden in den nachfolgenden Absätzen beschrieben.

Der erste Datentyp, den der xy -Graph annimmt, ist ein aus einem x -Array und einem y -Datenarray bestehender Cluster. Das folgende Diagramm veranschaulicht das Vorgehen beim Erstellen derartiger Daten.



Der zweite Datentyp ist ein Array aus *Punkten*, wobei ein Punkt ein Cluster aus einem x -Wert und einem y -Wert ist. Das folgende Diagramm veranschaulicht das Vorgehen beim Erstellen derartiger Daten.



Plotten von Mehrplot-Graphen

Sie können mehrere Plots in einem Kurven- oder xy -Graphen darstellen. Im Großen und Ganzen werden Arrays aus den im vorstehenden Abschnitt beschriebenen Datentypen verwendet, um mehrere Plots zur Darstellung in einem einzelnen Graphen zu beschreiben. Da G keine Arrays aus Arrays annimmt, können Sie in einem Fall, in dem durch die Erstellung eines Arrays aus einem Einzelplot-Datentyp ein Array aus Arrays erzeugt wird, entweder 2D-Arrays oder Arrays aus Arrayclustern verwenden.

Datentypen für Kurvengraphen

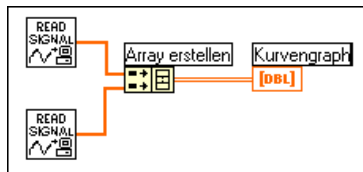
Der Multiplot-Kurvengraph nimmt fünf Datentypen an, die in den folgenden Absätzen beschrieben werden.

Der erste Datentyp, den ein Multiplot-Kurvengraph annimmt, ist ein zweidimensionales Array aus Werten, wobei jede Zeile der Daten ein einzelner Plot ist. Der Graph interpretiert diese Daten als Punkte und erhöht sie, angefangen bei $x = 0$, um eins.

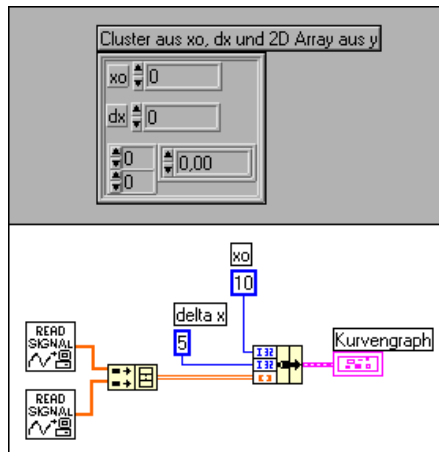


Wenn Sie den Menüpunkt **Array transponieren** aus dem Popup-Menü des Graphen wählen, wird jede Datenspalte als ein Plot behandelt. Dies ist insbesondere dann nützlich, wenn mehrere Kanäle einer Datenerfassungskarte abgetastet werden, da diese Daten als 2D-Arrays wiedergegeben werden, wobei jeder Kanal in einer separaten Spalte gespeichert wird.

Das folgende Beispiel zeigt, wie zwei Signale mit einem Graphen dargestellt werden, wobei jedes Signal eine separate Zeile eines zweidimensionalen Arrays ist.



Der zweite Datentyp ist ein aus einem x -Wert, einem Δx -Wert und einem zweidimensionalen Array aus y -Daten bestehender Cluster. Die y -Daten werden wie beim vorherigen Datentyp beschrieben interpretiert. Dieser Datentyp ist nützlich, um mehrere Signale darzustellen, die bei derselben normalen Abtastgeschwindigkeit gewonnen werden. Das folgende Diagramm veranschaulicht das Vorgehen beim Erstellen derartiger Daten.

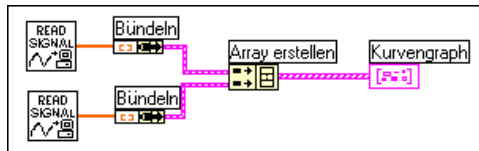


Der dritte Datentyp ist ein Array aus Clustern eines Arrays aus y -Daten. Das innere Array beschreibt die Punkte in einem Plot, und das äußere Array besitzt ein Element für jeden Plot.



Benutzen Sie diese Datenstruktur anstelle eines zweidimensionalen Arrays, wenn die Anzahl der Elemente in jedem Plot unterschiedlich ist, z.B. zum Abtasten von Daten aus mehreren Kanälen mit verschiedenen Abtastzeiten pro Kanal. Benutzen Sie diese Datenstruktur, da jede Zeile eines zweidimensionalen Arrays zwar dieselbe Anzahl von Elementen haben muß, aber die Anzahl der Elemente in den inneren Arrays eines Arrays aus Clustern verschieden sein kann.

Die folgende Abbildung zeigt das Vorgehen beim Erstellen der entsprechenden Datenstruktur aus zwei Arrays.

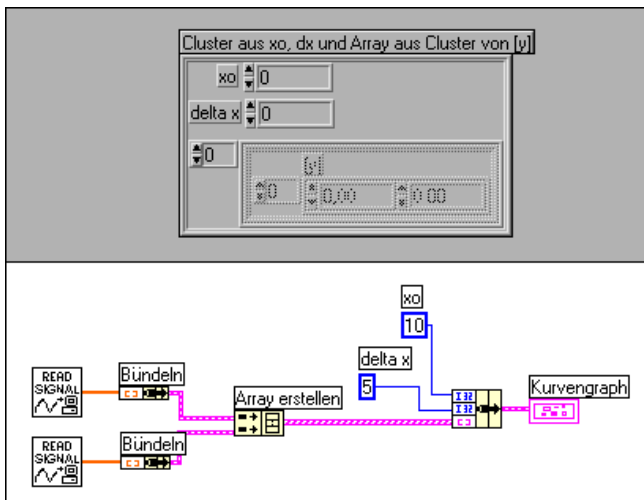


Das VI eines Kurvengraphen im Verzeichnis `examples\general\graphs\gengraph.llb` ist ein Beispiel für dieses Konzept.

Eine andere Art, Arrays in Elemente eines Arrays aus Clustern zu konvertieren, ist die Funktion “Cluster-Array erstellen”.

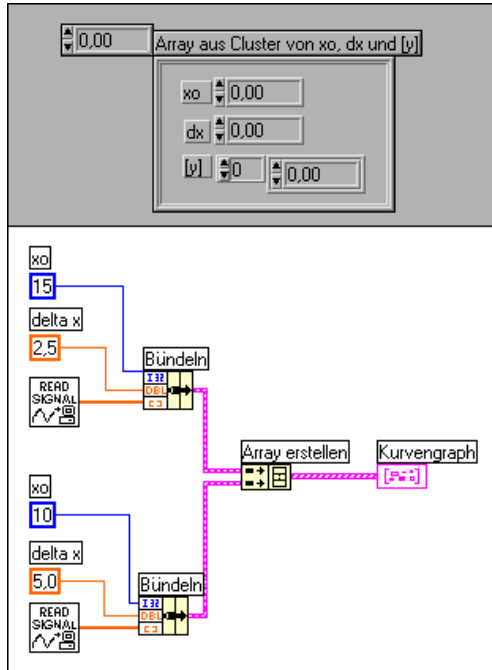
Der vierte Datentyp ist ein Cluster aus einem x -Ausgangswert, einem Δx -Wert und einem Array aus Clustern eines Arrays aus y -Daten.

Benutzen Sie diese Datenstruktur anstelle eines zweidimensionalen Arrays, wenn die Anzahl der Elemente in jedem Plot unterschiedlich ist, z.B. zum Abtasten von Daten aus mehreren Kanälen mit verschiedenen Abtastzeiten pro Kanal. Benutzen Sie diese Datenstruktur, da jede Zeile eines zweidimensionalen Arrays zwar dieselbe Anzahl von Elementen haben muß, aber die Anzahl der Elemente in den inneren Arrays eines Arrays aus Clustern verschieden sein kann. Das folgende Diagramm veranschaulicht das Vorgehen beim Erstellen derartiger Daten.



Beachten Sie, daß die Arrays mit Hilfe der Funktion "Bündeln" zu Clustern gebündelt werden und mit der Funktion "Array erstellen" ein Array aus den resultierenden Clustern erstellt wird. Sie können die Funktion "Cluster-Array erstellen" verwenden, die statt dessen Arrays aus Clustern mit angegebenen Eingaben erstellt.

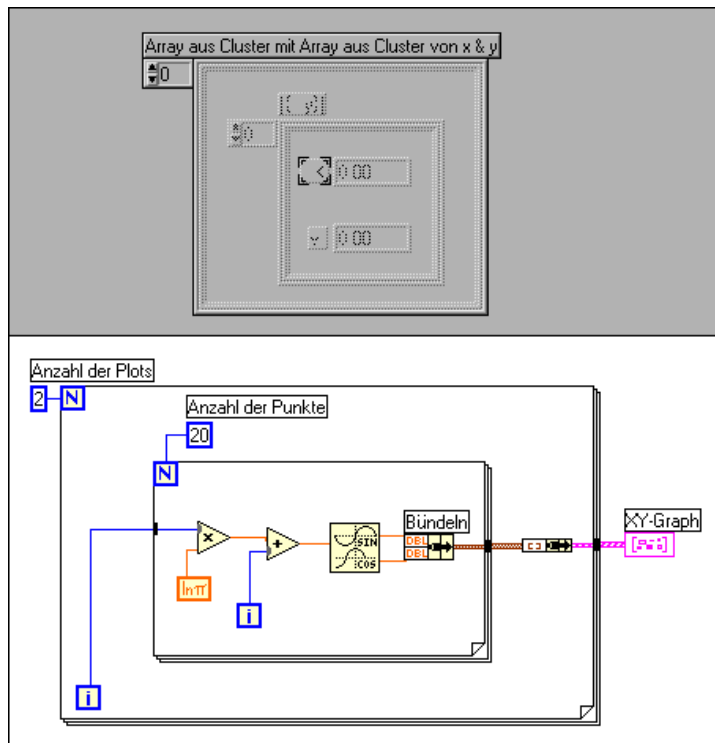
Der fünfte Datentyp ist ein Array aus Clustern, die aus einem x -Wert, einem Δx -Wert und einem Array aus y -Daten bestehen. Dies ist der allgemeinste Multiplot-Datentyp in Form eines Kurvengraphen, da Sie einen eindeutigen Startpunkt angeben können und entlang der x -Achse jedes Plots um eins weiterzählen können. Das folgende Diagramm veranschaulicht das Vorgehen zum Erstellen derartiger Daten.



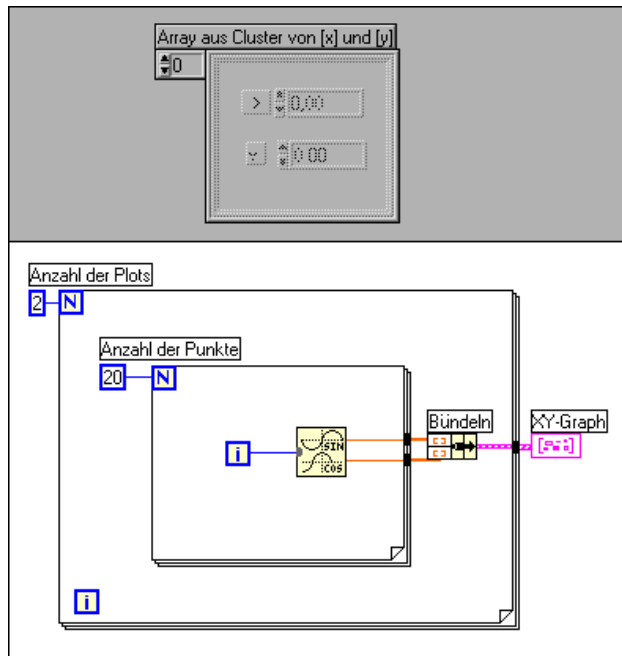
Datentypen für xy-Graphen

Der xy-Graph nimmt zwei Multiplot-Datentypen an, die in den folgenden Absätzen beschrieben werden. Beide Datentypen sind Arrays aus Clustern, die aus den zuvor beschriebenen Einzelplot-Datentypen bestehen.

Der erste Datentyp ist ein Array aus Plot-Clustern, wobei ein Plot ein Array aus Punkten ist. Ein Punkt ist als ein Cluster definiert, der einen x- und einen y-Wert besitzt. Das folgende Diagramm veranschaulicht das Vorgehen beim Erstellen derartiger Daten.



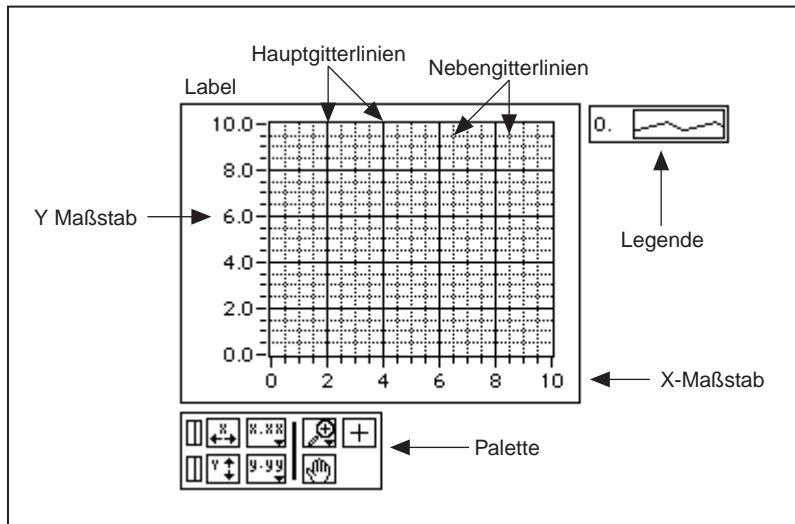
Der zweite Datentyp ist ein Array aus Plots, wobei ein Plot ein aus einem x -Array und einem y -Array bestehender Cluster ist. Das folgende Diagramm veranschaulicht das Vorgehen beim Erstellen derartiger Daten.



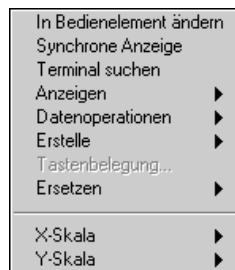
Einstellen benutzerspezifischer Optionen bei einem Graphen

Beide Graphen besitzen optionale Bestandteile, die mit dem Untermenü **Anzeigen** aus dem Popup-Menü des Graphen angezeigt oder verborgen werden können. Zu diesen Menüpunkten zählen eine Legende, mit der Sie die Farbe und den Stil eines bestimmten Plots definieren können, eine Palette, mit der Maßstab- und Formatoptionen während der Ausführung des VIs geändert werden, sowie eine Palette namens **Cursor**, mit der mehrfache Cursor angezeigt werden können. Das folgende Bild zeigt einen

Graphen mit allen optionalen Komponenten mit Ausnahme der Palette **Cursor**, die im Abschnitt *Cursor des Graphen* abgebildet ist.



Graphen verfügen über zahlreiche Optionen, mit denen die Darstellung der Daten angepaßt werden kann. Die folgende Abbildung zeigt das Pop-up-Menü für Graphen. Der Menüpunkt **Array transponieren** ist nur beim Kurvengraphen verfügbar.



Maßstaboptionen

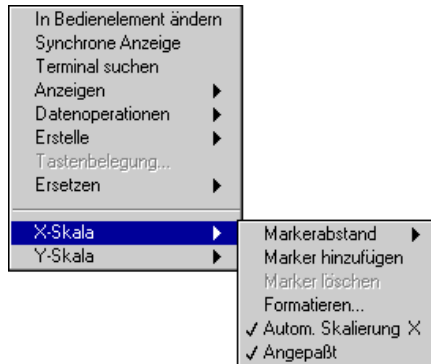
Graphen können ihre vertikale und horizontale Achse automatisch anpassen, um die Daten zu reflektieren, die damit verbunden sind. Diese automatische Skalierungsfunktion wird mit den Menüpunkten **Autom. Skalierung X** und **Autom. Skalierung Y** aus den Untermenüs **Datenoperationen** oder **X-Achse/Y-Achse** des Pop-up-Menüs des Graphen ein- und ausgeschaltet. Sie können diese automatischen Skalierungsoptionen, wie weiter unten in diesem Kapitel beschrieben, auch

über die Palette für den Graphen steuern. Die automatische Skalierung ist bei Graphen standardmäßig aktiviert, sie kann jedoch dazu führen, daß der Graph langsamer abläuft.

Sie können die horizontale oder vertikale Achse, wie bei jedem anderen Bedien- oder Anzeigeelement auch, mit dem Bedien- oder Beschriftungswerkzeug direkt ändern.

Das Untermenü **Datenoperationen** des Popup-Menüs für einen Graphen beinhaltet einen Menüpunkt namens **Graphik glätten**, der einen Offscreen-Puffer verwendet, um das Flackern auf ein Minimum zu reduzieren. Diese Funktion kann dazu führen, daß der Graph je nach dem verwendeten Computer- und Graphiksystem langsamer abläuft.

Die x - und y -Achse besitzen jeweils ein eigenes Untermenü (siehe nachstehende Abbildung).



Markierungsabstand

Standardmäßig sind Markierungswerte für die x - und y -Achse gleichmäßig verteilt. Wenn Sie dies vorziehen, können Sie Markierungen an beliebigen Positionen auf der x - oder y -Achse angeben. Dies ist nützlich, um einige bestimmte Punkte in einem Graphen (wie z.B. einen Bezugspunkt oder Schwellwert) zu markieren.

Wenn Sie eine ungleichmäßige Verteilung der Markierungen wünschen, wählen Sie aus dem Popup-Menü der jeweiligen Achse **X- oder Y-Achse»Markerabstand»Arbiträr**. Nachdem Sie die Auswahl getroffen haben, nimmt der Cursor die Form des Doppelpfeilcursors (siehe nachstehende Abbildung) an, wenn Sie das Bedienwerkzeug über einen Tickmarker bewegen. Sie können dann einen neuen Marker erstellen, indem Sie den bestehenden Tickmarker bei gedrückter <Strg>-Taste

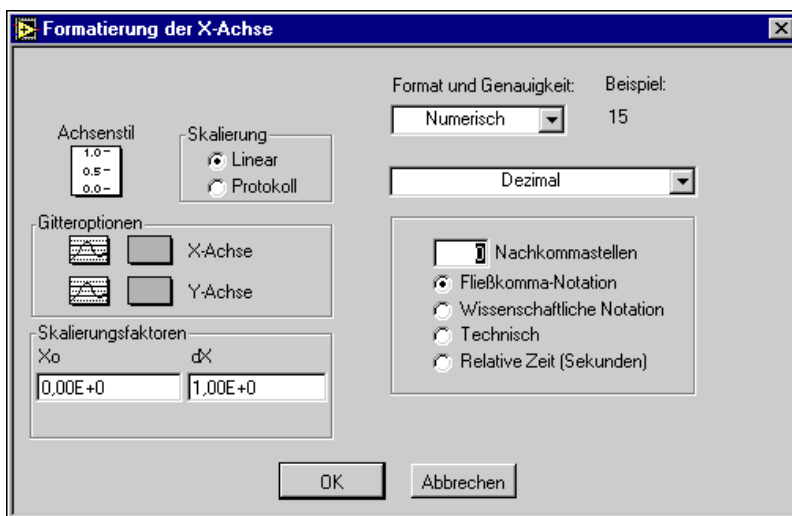
(Windows), <Optionstaste> (Macintosh), <Meta>-Taste (Sun) oder <Alt>-Taste (HP-UX) ziehen, oder Sie können den bestehenden Tickmarker durch Ziehen an eine beliebige Stelle auf der Achse verschieben.



Sie können einen Marker hinzufügen oder löschen, indem Sie das Pop-up-Menü des Graphen oder der Achse aufrufen und **Marker hinzufügen** oder **Marker löschen** wählen. Wenn ein Marker erstellt wird, geben Sie eine Zahl in den Marker ein, um dessen Position zu ändern.

Formatieren

Klicken Sie auf **Formatieren...**, um das in der folgenden Abbildung gezeigte Dialogfeld aufzurufen.



Wählen Sie die Optionen in diesem Dialogfeld zum Einstellen der folgenden Eigenschaften des Graphen.

Achsenstil — Mit dieser Option werden die Haupt- und Nebentickmarker der Achse festgelegt. Haupttickmarker sind Punkte, die Labeln der Achse entsprechen, und Nebentickmarker sind Punkte zwischen Labeln. Sie können diese Palette auch verwenden, um zu entscheiden, ob die Marker einer bestimmten Achse sichtbar sein sollen.

Skalierung — Benutzen Sie diese Option der Maßstabmenüs, um zwischen linearer und logarithmischer Achsenskalierung zu wählen.

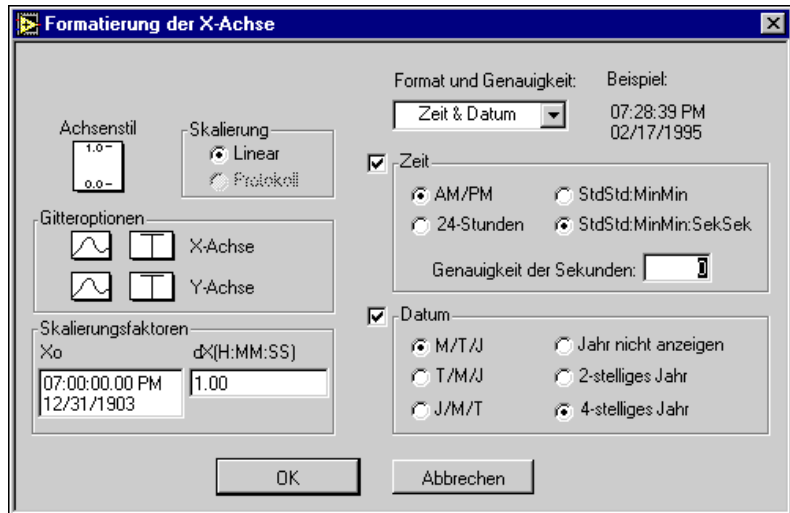
Gitteroptionen — Durch Klicken auf diese Optionen wird eine Palette mit Gittertypen aufgerufen, in der Sie auswählen können, ob keine Gitterlinien, nur an den Stellen der Haupttickmarker (Punkte, die den Achsenlabeln entsprechen) oder an den Haupt- und Nebentickmarkern (Nebentickmarker sind die Punkte zwischen Labeln) Gitterlinien angezeigt werden. Die Steuerung neben dem Popup-Menü des Gitters ist ein Farbmnü, über das Sie die Farbe für die Gitterlinien wählen können.

Skalierungsfaktoren — Die Skalierungsfaktoren geben den Anfangswert und die Abstände zwischen Punkten auf einem Kurvendiagramm oder einem Kurvengraph oder entlang der Achsen eines Intensitätsdiagramms oder eines Intensitätsgraphen an. Sie können diese Skalierungsfaktoren benutzen, um Ihre Daten zur Darstellung zu skalieren. Wenn Ihre Daten beispielsweise in einem Bereich zwischen $-2,048$ und $2,047$ binär abgetastet werden, skalieren Sie diese Daten auf die entsprechenden Spannungswerte zwischen 0 und 5, indem Sie einen Zusatzoffset von 2,5 und einen Skalierungsfaktor von $5/4095 = 0,001221$ angeben.

Format & Genauigkeit — Das Format der x - und y -Achse in numerischer oder Zeit- und Datumsangabe wird über einen Menüring oben im Dialogfeld ausgewählt.

Wenn Sie die Formatierung **Numerisch** wählen, können Sie zwischen Fließkomma-, wissenschaftlicher, technischer oder relativer Bezeichnung der Zeit in Sekunden wählen; Sie können ferner zwischen dezimaler, vorzeichenloser dezimaler, hexadezimaler, oktaler oder binärer Schreibweise wählen; Sie können auch die Genauigkeit (wie viele Ziffern rechts vom Dezimalkomma angezeigt werden) zwischen 0 und 20 wählen. Die Genauigkeit, die Sie wählen, wirkt sich nur auf die Anzeige des Werts aus; die interne Genauigkeit hängt dennoch von der Darstellung ab. Beispiele werden in dem Dialogfeld angezeigt, während Sie eine Kombination von Optionen auswählen.

Wenn Sie die Formatierung **Zeit & Datum** wählen, ändert sich das Dialogfeld der nachfolgenden Abbildung entsprechend.



Sie können entweder die Zeit oder das Datum oder beide absolut formatieren. Wenn Sie eine Achse bearbeiten und nur die Zeit oder nur das Datum eingeben, werden die nichtangegebenen Komponenten erschlossen. Wenn Sie bei der Bearbeitung keine Zeit eingeben, wird 12:00 (0:00) Uhr angenommen. Wenn Sie kein Datum eingeben, wird der vorherige Datumswert angenommen. Wenn Sie ein Datum eingeben, die Skalierung aber nicht im Datumsformat ist, basiert die Anordnung von Monat, Tag und Jahr auf den Einstellungen im Dialogfeld **Voreinstellungen**. Wenn Sie lediglich zwei Ziffern für das Jahr eingeben, wird folgendes vorausgesetzt: eine beliebige Zahl, die kleiner als 38 ist, bezieht sich auf das 21. Jahrhundert, alle anderen Zahlen auf das 20. Jahrhundert.

Ogleich die absolute Zeit als String aus Zeit und Datum angezeigt wird, wird sie intern als die Anzahl von Sekunden seit 0:00 Uhr, dem 1. Januar 1904 (UTC - Universal Time Coordinated) dargestellt, die früher als Greenwich Mean Time (GMT) bezeichnet wurde.

Beachten Sie die Beispiele rechts oben im Dialogfeld, die sich jeweils ändern, wenn Sie eine Auswahl treffen.

Der gültige Bereich für die Zeit und das Datum unterscheidet sich wie folgt je nach Plattform.

- **(Windows)** 0:00 Uhr, 2. Jan. 1970 – 0:00 Uhr, 3. Jan. 2040
- **(Windows NT)** 0:00 Uhr, 1. Jan. 1904 – 0:00 Uhr, 3. Jan. 2040

- **(Macintosh)** 0:00 Uhr, 2. Jan. 1904 – 0:00 Uhr, 2. Jan. 2040
- **(UNIX)** 0:00 Uhr, 1. Jan. 1904 – 0:00 Uhr, 17. Jan. 2038

Diese Bereiche können sich je nach Zeitzone und bei/ohne Sommerzeit um einige Tage verschieben.

Automatische Skalierung



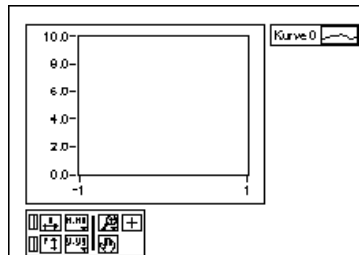
Mit **Autom. Skalierung (X oder Y)** wird die automatische Skalierung ein- oder ausgeschaltet.

Angepaßt

Ist **Angepaßt** eingeschaltet, werden die Endmarker auf ein Vielfaches des für den Maßstab verwendeten Inkrements gerundet. Wenn die Achsen genau auf den Bereich der Daten eingestellt werden sollen, schalten Sie den Menüpunkt **Angepaßt** im Popup-Menü des Graphen aus.

Scroll- und Zoomoptionen

Die Palette **Graph** ist bei jedem Graphen vorhanden, den Sie auf dem Frontpanel ablegen. Diese Palette verfügt über Steuerungen zum Scrollen (Rollen des Anzeigebereichs eines Graphen) und Vergrößern und Verkleinern (Zoomen) von Abschnitten des Graphen. Die nachstehende Abbildung zeigt einen Graphen und dessen Begleitpalette **Graph**.



Wenn Sie die Schaltfläche **Autom. Skalierung X** (siehe Abbildung links) drücken, skaliert der Graph automatisch die x-Achse. Wenn Sie die Schaltfläche **Autom. Skalierung Y** drücken (siehe Abbildung links), skaliert der Graph automatisch die y-Achse. Soll der Graph beide Achsen fortgesetzt skalieren, klicken Sie auf den Sperrschalter (siehe Abbildung links), um die automatische Skalierung eingeschaltet zu fixieren.





Mit Hilfe der Schaltflächen für **Achsenformatierung** (siehe Abbildung links) können Sie während der Ausführung jeweils die Kontrolle über das Format der x - und y -Achsenmarker behalten.

Sie können die verbleibenden drei Schaltflächen dazu verwenden, den Bedienungsmodus für den Graphen zu steuern.



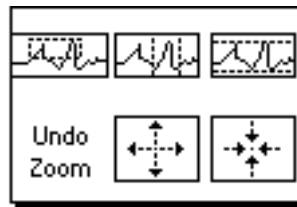
Normalerweise befinden Sie sich im standardmäßigen Arbeitsmodus, was durch das Pluszeichen oder Fadenkreuz angezeigt wird. Im Arbeitsmodus können Sie zum Umherbewegen von Cursors in den Graphen klicken.



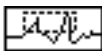
Wenn Sie das Scrollwerkzeug (siehe Abbildung links) drücken, wechseln Sie zu einem Modus, in dem Sie durch Klicken und Ziehen des Plotbereichs des Graphen einen Bildlauf durch die sichtbaren Daten durchführen können.



Wenn Sie das Zoom-Werkzeug drücken (siehe Abbildung links), können Sie Ausschnitte des Graphen vergrößern oder verkleinern. Wenn Sie auf das Zoom-Werkzeug klicken, erscheint ein Popup-Menü, aus dem Sie Zoom-Methoden wählen können. Dieses Menü ist in der folgenden Abbildung gezeigt.



Es folgt eine Beschreibung der einzelnen Menüpunkte.



Rechteckiger Zoombereich.



Rechteckiger Zoombereich, wobei der Zoom auf x -Daten beschränkt ist (die y -Achse bleibt unverändert).



Rechteckiger Zoombereich, wobei der Zoom auf y -Daten beschränkt ist (die x -Achse bleibt unverändert).



Macht die letzte Zoomoperation rückgängig. Setzt den Graphen auf die vorherige Einstellung zurück.



Vergrößert den Bereich um einen Punkt. Wenn Sie die Maustaste an einem bestimmten Punkt gedrückt halten, wird der Graph um diesen Punkt solange fortgesetzt vergrößert, bis Sie die Maustaste loslassen.



Verkleinert den Bereich um einen Punkt. Wenn Sie die Maustaste an einem bestimmten Punkt gedrückt halten, wird der Graph um diesen Punkt solange fortgesetzt verkleinert, bis Sie die Maustaste loslassen.



Hinweis

Bei den letzten beiden Modi, Vergrößern und Verkleinern um einen Punkt, wird die Zoomrichtung durch Drücken der <Umschalt>-Taste und Klicken umgekehrt.

Legendenoptionen

Der Graph verwendet für jeden neuen Plot einen Standardstil, es sei denn, Sie erstellen dafür einen benutzerspezifischen Plotstil. Sollen bestimmte Plots eines Multiplot-Graphen gewisse Merkmale aufweisen (soll etwa der dritte Plot blau sein), können Sie diese Merkmale mit der Legende einstellen, die mit dem Untermenü **Anzeigen** aus dem Popup-Menü des Graphen angezeigt oder verborgen werden kann. Sie können mit Hilfe der Legende außerdem für jeden Plot einen Namen angeben.



Wenn Sie **Legende** wählen, erscheint nur ein Plot. Sie können mehrere Plots erstellen, indem Sie eine Ecke der Legende mit dem Größenänderungswerkzeug nach unten verlängern. Nachdem Sie Plotmerkmale eingestellt haben, weist der Graph dem Plot diese Merkmale ungeachtet der Tatsache zu, ob die Legende angezeigt wird oder nicht. Wenn der Graph mehr als die in der Legende definierten Plots empfängt, zeichnet der Graph sie mit dem standardmäßigen Stil.

Wenn Sie den Graphenkörper verschieben, wird die Legende mitverschoben. Sie können die Position der Legende in Bezug auf den Graphen ändern, indem Sie die Legende einfach an eine andere Position verschieben. Vergrößern Sie die Legende nach links, um auf den Labels mehr Platz zu schaffen, oder nach rechts, um auf den Plotmustern mehr Platz zu schaffen.

Standardmäßig wird jeder Plot angefangen bei Null mit einer Zahl beschriftet. Sie können dieses Label wie andere Labels auch modifizieren. Rechts vom Plotlabel befindet sich das *Plotmuster*. Jedes Plotmuster besitzt ein eigenes Popup-Menü, mit dem Plot-, Linien-, Farb- und Punktstil des

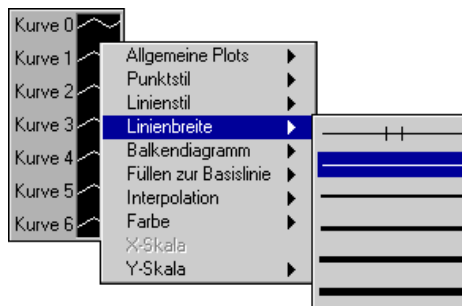
Plots geändert werden können. Das Array aus Punkten, das mit dem Graphen verbunden wird, wird mit den Merkmalen angezeigt, die Sie ihm in der Graphenlegende zuweisen.

Die nachstehende Abbildung zeigt das Popup-Menü des Plotmusters.



Der Menüpunkt **Allgemeine Plots** hilft Ihnen dabei, einen Plot für einen von sechs weitverbreiteten Plotstils zu konfigurieren, einschließlich eines Streudiagramms, Balkendiagramms und eines Diagramms der Art “Bis Null füllen”. Der Punkt-, Linien- und Füllstil der Menüpunkte in dieser Unterpalette ist vorkonfiguriert.

Die Menüpunkte **Punktstil**, **Linienstil** und **Linienbreite** zeigen Stile an, mit denen ein Plot differenziert werden kann. Die Unterpalette für Linienbreite enthält Breiten, die über die Standardeinstellung von 1 Pixel hinausgehen, sowie Haarlinienbreite. Letztere Option hat keine Auswirkung auf die Bildschirmanzeige, druckt jedoch eine sehr dünne Linie, sofern der Drucker und der Druckmodus Haarliniendruck unterstützen.

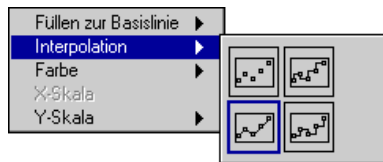


Hinweis *Unter Windows sind bei breiten Stiften nur Volllinien möglich.*

Der Menüpunkt **Balkendiagramme** verfügt über eine Auswahl aus vertikalen Balken, horizontalen Balken und Darstellung ohne Balken.

Der Menüpunkt **Füllen zur Basislinie** legt fest, bis wohin von der Basislinie aus gefüllt werden soll. **Null** füllt vom Plot aus bis zu einer Basislinie, die bei 0 generiert wird. **Unendlich** füllt vom Plot aus bis zur positiven Kante des Graphen. **Minus Unendlich** füllt vom Plot aus bis zur negativen Kante des Graphen. Mit dem unteren Abschnitt dieses Menüs können Sie einen bestimmten Plot des Graphen wählen, bis zu dem gefüllt werden soll.

Der Menüpunkt **Interpolation** ruft die in der folgenden Abbildung dargestellte Palette auf, in der Sie wählen, wie der Graph Linien zwischen geplotteten Punkten zeichnet. Bei der ersten Option wird keine Linie gezeichnet, wodurch sie sich für einen Streuplot eignet. Die Option unten links zeichnet eine gerade Linie zwischen geplotteten Punkten. Die zwei abgestuften Optionen, die Punkte mit einem rechtwinkligen Ellenbogen verbinden, sind nützlich, um histogrammartige Plots zu erstellen. Die Option rechts oben plottet zuerst in y -Richtung, und die Option rechts unten plottet zuerst in x -Richtung.



Der Menüpunkt **Farbe** zeigt die Palette zum Auswählen der Plotfarbe an. Sie können die Plots auch in der Legende mit dem Farbenwerkzeug farbig gestalten und die Plotfarben während der Ausführung des VIs ändern.

Der Menüpunkt **Y-Skala** zeigt eine Liste der y -Skalen im Graphen an. Diese Option wird bei gestapelten Diagrammen dazu verwendet, zu definieren, bei welcher Skala jeder Plot geplotet wird.

Kurvendiagramme

Das Kurvendiagramm ist ein besonderer Typ eines numerischen Anzeigeelements, das ein oder mehrere Plots anzeigt. Diagramme unterscheiden sich von Graphen darin, daß Diagramme vorherige Daten bis zu einem definierbaren Maximalwert beibehalten. Neue Daten werden an die alten Daten angefügt, so daß Sie den aktuellen Wert im Zusammenhang mit den vorherigen Daten sehen können.

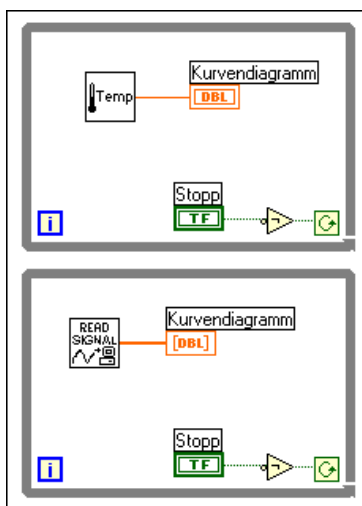
Ein Beispiel eines Kurvendiagramms finden Sie unter `examples\general\graphs\charts.llb`.

Datentypen für Kurvendiagramme

Sie können entweder einzelne Werte oder mehrere Werte auf einmal an Diagramme übergeben. Wie beim Kurvengraphen wird jeder Wert wie ein Teil einer gleichmäßig strukturierten Kurve behandelt, wobei jeder Punkt entlang der x-Achse vom vorangegangenen um einen Punkt versetzt wird.

Sie können entweder einen einzelnen Skalar oder ein Array aus mehreren Werten an das Diagramm übergeben. Bei einem einzelnen Plot behandelt das Diagramm diese Eingaben wie neue Daten.

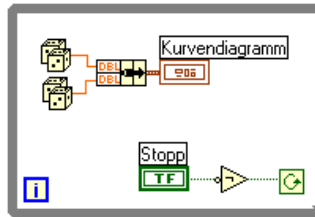
Die nachfolgenden Diagramme veranschaulichen die Verwendung des Diagramms in Verbindung mit derartigen Daten.



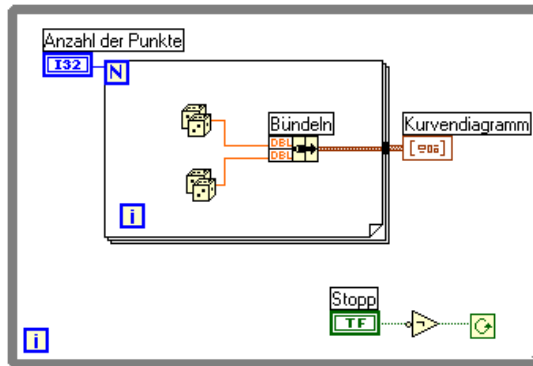
Wenn mehrere Punkte auf einmal an das Diagramm übergeben werden, wird es nicht so oft neu gezeichnet, als wenn es bei jedem Punkt gezeichnet wird.

Sie können Daten für mehrere Plots auf mehrere Arten an ein Kurvendiagramm übergeben. Die erste Methode besteht darin, die Daten zu einem Cluster aus skalaren Nummern zu bündeln, wobei jede Ziffer einen

einzelnen Punkt für jeden der Plots darstellt. Die nachfolgende Abbildung zeigt ein Beispiel hierfür.

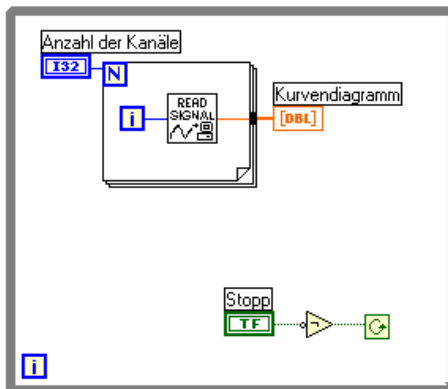


Wenn Sie mehrere Punkte für Plots bei einer einzigen Aktualisierung übergeben wollen, können Sie ein Array aus Clustern mit Ziffern mit dem Diagramm verbinden. Jede Ziffer stellt einen Punkt für jeden der Plots dar. Die folgende Abbildung zeigt ein Beispiel hierfür.



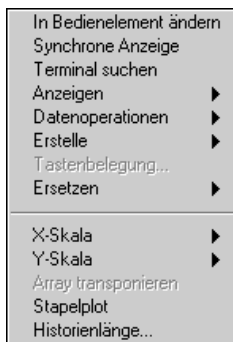
Wenn die Anzahl der Plots, die Sie darstellen wollen, erst bei der Ausführung bestimmt werden kann oder wenn Sie mehrere Punkte für Plots in einer einzigen Aktualisierung übergeben wollen, können Sie ein zweidimensionales Array aus Daten mit dem Diagramm verbinden. Wie beim Kurvengraphen werden Zeilen normal wie neue Daten für jeden Plot behandelt. Sie können den Menüpunkt **Array transponieren** aus dem

Popup-Menü des Kurvendiagramms dazu verwenden, Spalten als neue Daten für jeden Plot zu behandeln.



Optionen für Kurvendiagramme

Das Diagramm verfügt im wesentlichen über dieselben Funktionen wie der Graph, einschließlich der Legende und Palette, die in derselben Weise funktionieren. (Weitere Informationen finden Sie in den Abschnitten [Maßstaboptionen](#) und [Legendenoptionen](#) in diesem Kapitel.) Das Kurvendiagramm unterstützt keine Cursor. Die folgende Abbildung zeigt das Popup-Menü für Diagramme.



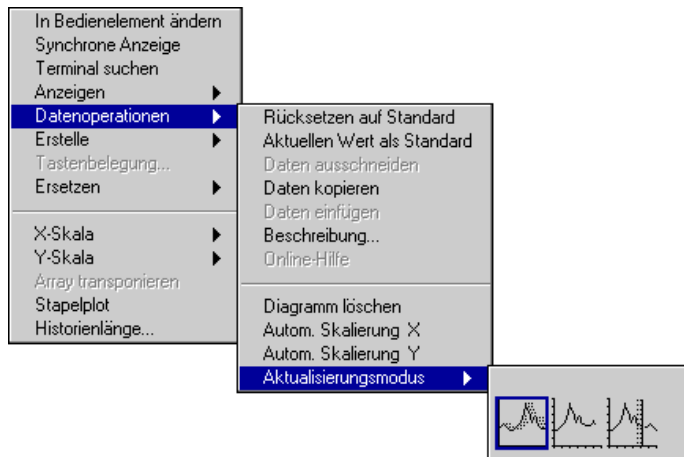
Mit dem Untermenü **Anzeigen** des Popup-Menüs für Diagramme können Sie optionale digitale Anzeigen und einen Rollbalken anzeigen oder verbergen. Der Menüpunkt **Digitale Anzeige** zeigt den zuletzt geplotteten Wert an. Der zuletzt vom Blockdiagramm an das Diagramm übergebene Wert ist der letzte neue Wert für das Diagramm. Es gibt nur eine digitale Anzeige pro Plot.

Sie können vorherige, im Puffer enthaltene Werte anzeigen, indem Sie die x -Achse mit dem Rollbalken bis zu einem Bereich mit zuvor geplotteten Werten rollen oder indem Sie den x -Achsenbereich so ändern, daß alte Daten angezeigt werden.

Es gibt eine Grenze für die Datenmenge, die im Puffer des Diagramms gespeichert werden kann, um zu vermeiden, daß die Speicherressourcen knapp werden. Die Standardgröße dieses Puffers beträgt 1.024 Punkte. Wenn das Diagramm diesen Grenzwert erreicht, werden die ältesten Punkte gelöscht, um Platz für neue Daten zu machen. Sie können die Länge dieses Puffers mit dem Menüpunkt **Historienlänge...** aus dem Popup-Menü des Diagramms ändern.

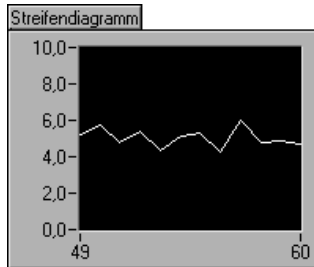
Aktualisierungsmodi von Diagrammen

Um die Verhaltensweise des Diagramms beim Eingang neuer Daten auf der Anzeige zu ändern, benutzen Sie den Menüpunkt **Aktualisierungsmodus** aus dem Untermenü **Datenoperationen** des Popup-Menüs des Diagramms (siehe nachstehende Abbildung).

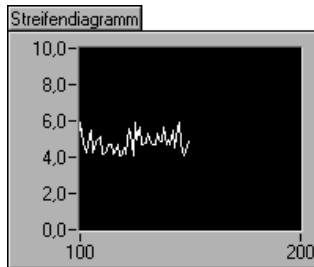


Die drei Menüpunkte — Streifen-, Oszilloskop- und Sweepdiagramm — werden in den folgenden Abbildungen dargestellt und jeweils im Anschluß daran erläutert. Der Standardmodus ist Streifendiagramm.

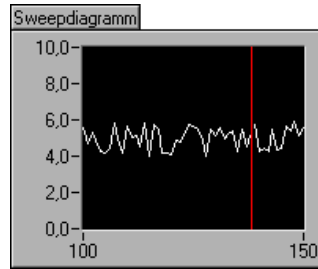
Das *Streifendiagramm* weist eine rollende Anzeige auf, ähnlich einem Papierstreifenschreiber. Jeder neu eingehende Wert wird am rechten Rand geplottet, und alte Werte werden nach links verschoben.



Der Modus *Oszilloskopdiagramm* weist, einem Oszilloskop ähnlich, eine zurückziehende Anzeige auf. Jeder neu eingehende Wert wird rechts vom vorhergehenden Wert geplottet. Wenn der Plot den rechten Rand des Plotbereichs erreicht, wird der Plot gelöscht und das Plotten fängt am linken Rand neu an. Das Oszilloskopdiagramm ist bedeutend schneller als das Streifendiagramm, da der zusätzliche Bearbeitungsaufwand wegfällt, der beim Rollen des Bildes anfällt.

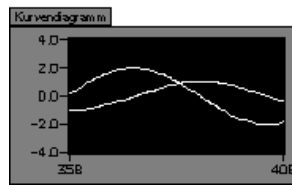


Der Modus *Sweepdiagramm* funktioniert im wesentlichen wie das Oszilloskopdiagramm, aber der Plot wird nicht gelöscht, wenn die Daten an den rechten Rand treffen. Statt dessen markiert eine vertikale Linie den Anfang neuer Daten und wird mit neu eingehenden Daten über den Bildschirm verschoben.

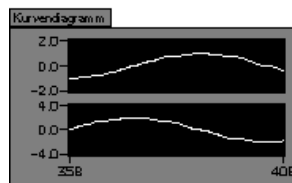


Gestapelte und überlagerte Plots

Standardmäßig zeigt das Diagramm mehrere Plots an, indem diese übereinander überlagert werden, etwa wie Graphen, die auf demselben Gitter gezeichnet werden. Die folgende Abbildung zeigt ein Beispiel für überlagerte Plots.



Als Alternative können Plots mit einer anderen y-Achse für jeden Plot übereinander gestapelt werden, indem Sie den Menüpunkt **Stapelplot** aus dem Popup-Menü des Diagramms wählen. Wenn Sie dies tun, kann die y-Achse für jedes Diagramm einen anderen Bereich haben. Die nachstehende Abbildung zeigt ein Beispiel eines Stapelplots.

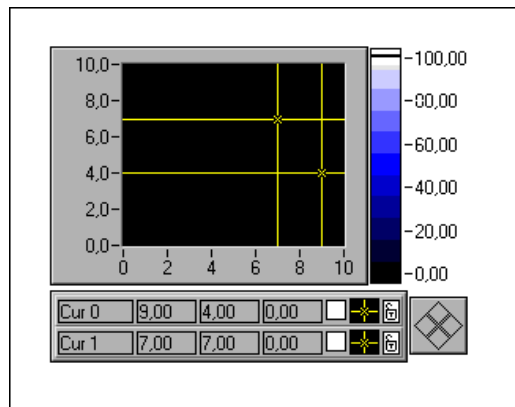
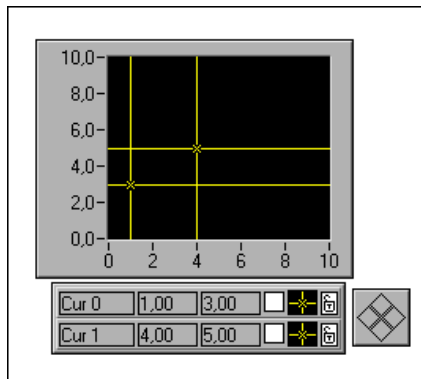


Wenn Sie Daten als Cluster in das Diagramm eingeben, stapelt das Diagramm automatisch die richtige Anzahl von Plotdarstellungen. Wenn Sie Daten als 2D-Array eingeben, müssen Sie mit der Legende, die Sie mit dem Untermenü **Anzeigen** des Popup-Menüs aufrufen, die richtige Anzahl von Plotdarstellungen erstellen. Wenn Sie die Legendenanzeige vergrößern, um die Plotanzahl zu erhöhen, wächst die Anzahl der Stapelplotanzeigen entsprechend mit.

Cursor des Graphen

Für jeden Graphen können Sie eine **Cursor**-Palette anzeigen, mit der Cursor auf den Graphen plaziert werden. Sie können den Cursor auf dem Graphen beschriften und einen Cursor als Marker verwenden. Wenn Sie einen Cursor als Marker verwenden, fixieren Sie den Cursor an einem Datenplot, damit der Cursor den Daten folgt.

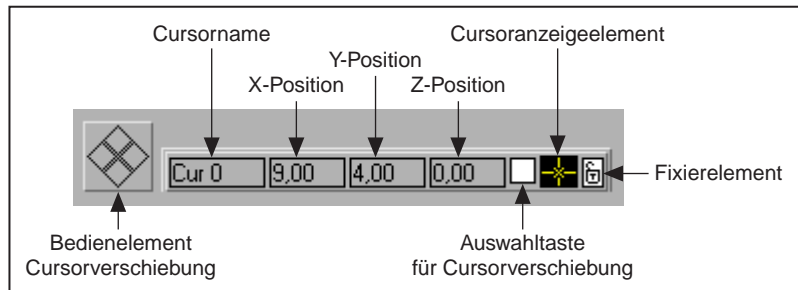
Die folgenden Abbildungen zeigen einen Kurvengraphen und einen Intensitätsgraphen jeweils mit angezeigter **Cursor**-Palette.



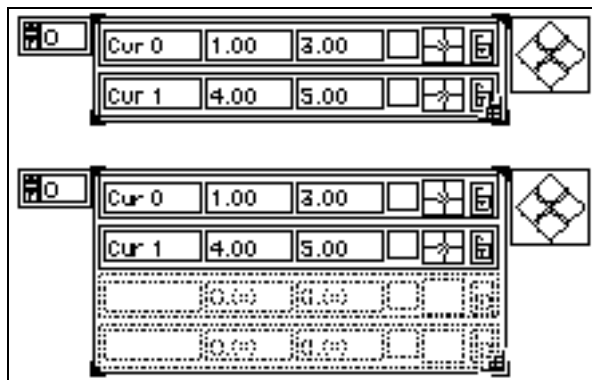
Jeder Cursor eines Graphen setzt sich aus den folgenden Bestandteilen zusammen.

- Einem Label
- x - und y -Koordinaten und, falls zutreffend, einer z -Koordinate
- Einer Schaltfläche, die den Plot für die Verschiebung mit dem Plotcursorblock markiert
- Einer Schaltfläche, die das Aussehen des Cursors steuert
- Einer Schaltfläche, die bestimmt, ob der Cursor an einem Plot fixiert oder frei beweglich ist

Die folgende Abbildung zeigt diese Bestandteile in einer Übersicht.

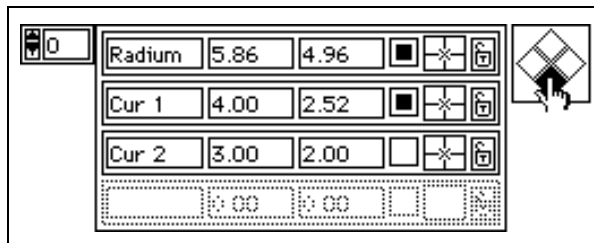



Die Palette **Cursor** verhält sich wie ein Array. Sie können sie dehnen, um Informationen über mehrere Cursor anzuzeigen, und Sie können die Indexsteuerung verwenden, um andere Cursorinformationen in der Palette anzuzeigen. Mit den Menüpunkten unter **Anzeigen** im Popup-Menü der Cursoranzeige können Sie die Indexsteuerung anzeigen.



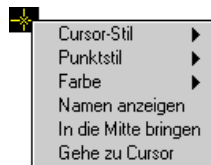
Um einen Cursor zu löschen, müssen Sie ihn mit den Menüpunkten **Auswahl beginnen** und **Auswahl beenden** aus dem Popup-Menü **Datenoperationen** wählen und dann mit dem Menüpunkt **Ausschneiden** aus demselben Menü ausschneiden. Weitere Informationen hierzu finden Sie im Abschnitt *Auswählen von Array-Zellen* in Kapitel 14, *Bedien- und Anzeigeelemente vom Typ Array und Cluster*.

Sie können einen Cursor in einem Graphen verschieben, indem Sie ihn mit dem Bedienwerkzeug ziehen oder indem Sie die Bewegungssteuerung des Cursors verwenden. Um einen Cursor zu ziehen, vergewissern Sie sich, daß weder das Scrollwerkzeug noch das Zoom-Werkzeug gewählt ist. Durch Klicken auf die Pfeile der Bewegungssteuerung des Cursors werden alle ausgewählten Cursor in die angegebene Richtung verschoben. Sie wählen Cursor entweder dadurch, daß Sie sie mit dem Bedienwerkzeug auf dem Graphen verschieben oder indem Sie auf die Auswahlfläche eines bestimmten Cursors klicken. Im folgenden Beispiel werden die oberen beiden Cursor vertikal nach unten verschoben.

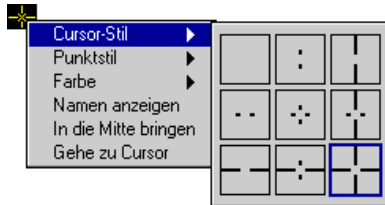



Steuerung der
Cursor-Anzeige

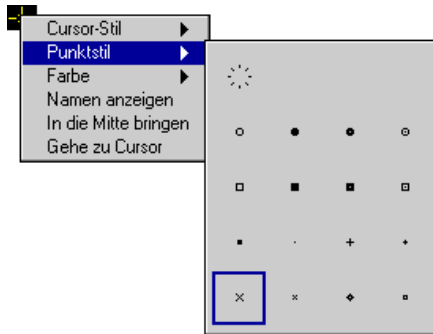
Durch Klicken mit dem Bedienwerkzeug auf die Steuerung der Cursor-Anzeige wird ein Popup-Menü angezeigt, mit dem das Aussehen des Cursors und die Anzeige des Cursornamens auf dem Plot gesteuert wird. Dieses Popup-Menü ist in der folgenden Abbildung dargestellt.



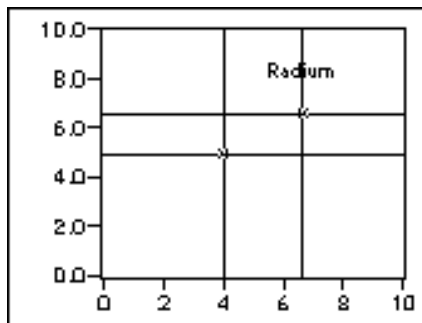
Aus diesem Menü wählen Sie den Stil des Fadenkreuzes. Die Fadenkreuze können aus einer vertikalen und/oder horizontalen Linie bestehen, die bis Unendlich geht, einer kleineren vertikalen und/oder horizontalen Linie mit einem Mittelpunkt um die Cursorposition, oder sie können keine Fadenkreuze besitzen (siehe nachfolgende Abbildung).



Sie können auch den Punktstil wählen, mit dem die Cursorposition und die Farbe des Cursors markiert wird (siehe nachstehende Abbildung).



Wählen Sie den Menüpunkt **Namen anzeigen** aus diesem Menü, um den Cursornamen auf dem Plot sichtbar zu machen (siehe nachstehende Abbildung).

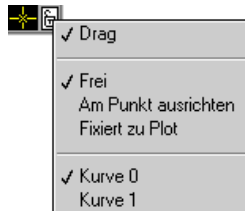


Durch Auswählen von **In die Mitte bringen** wird der Cursor wieder in den angezeigten Bereich des Graphen verschoben. Dies ist nützlich, wenn der Cursor aus dem sichtbaren Bereich verschoben wird. Durch Auswählen dieses Menüpunkts wird die (x,y) Koordinatenposition des Cursors geändert.

Durch Auswählen von **Gehe zu Cursor** wird der angezeigte Bereich des Graphen verschoben, so daß der Cursor sichtbar ist. Die Cursorposition bleibt konstant, aber die Achsen ändern sich, um den gewählten Cursor miteinzubeziehen. Die Größe des angezeigten Bereichs bleibt ebenfalls konstant. Diese Funktion ist nützlich, wenn der Cursor dazu verwendet wird, im Graphen einen Punkt von Interesse zu markieren, wie z.B. ein Minimum oder Maximum, und Sie wollen diesen Punkt ansehen.



Mit Hilfe der letzten Schaltfläche jedes Cursors können Sie einen Cursor auf einem bestimmten Plot fixieren. Durch Klicken auf die Fixierschaltfläche sehen Sie ein Popup-Menü, über das Sie den Cursor an einem bestimmten Plot fixieren können. Wenn Sie den Cursor an einem Plot fixieren, nimmt die Schaltfläche die Form eines geschlossenen Vorhängeschlosses an. Die folgende Abbildung zeigt das Popup-Menü der Fixierschaltfläche.



Der Menüpunkt **Drag** bestimmt, ob der Cursor mit der Maus verschoben werden kann. Ist **Drag** gewählt, können Sie den Cursor verschieben oder ziehen. Die Menüpunkte unterhalb der gestrichelten Linie im Menü bestimmen, wie der Cursor mit der Maus verschoben werden kann. Ist die Option **Drag** nicht gewählt, kann der Cursor auf dem Plot nicht umher verschoben werden.

Wählen Sie **Frei**, wenn Sie den Cursor an eine beliebige Stelle auf dem Graphen plazieren oder verschieben wollen. Wählen Sie **Am Punkt ausrichten**, wenn der Cursor auf einem beliebigen Plot stets am nächsten Punkt ausgerichtet werden soll. Wählen Sie **Fixiert zu Plot**, um den Cursor an einen bestimmten Plot anzuhängen. Wenn Sie **Fixiert zu Plot** zum erstenmal wählen, wird der Cursor automatisch an einem Punkt auf dem Plot fixiert. Nachdem der fixierte Cursor freigegeben und an eine beliebige neue Position verschoben worden ist, wird der Cursor durch Wählen von **Fixiert zu Plot** an die letzte Position des fixierten Cursors verschoben.

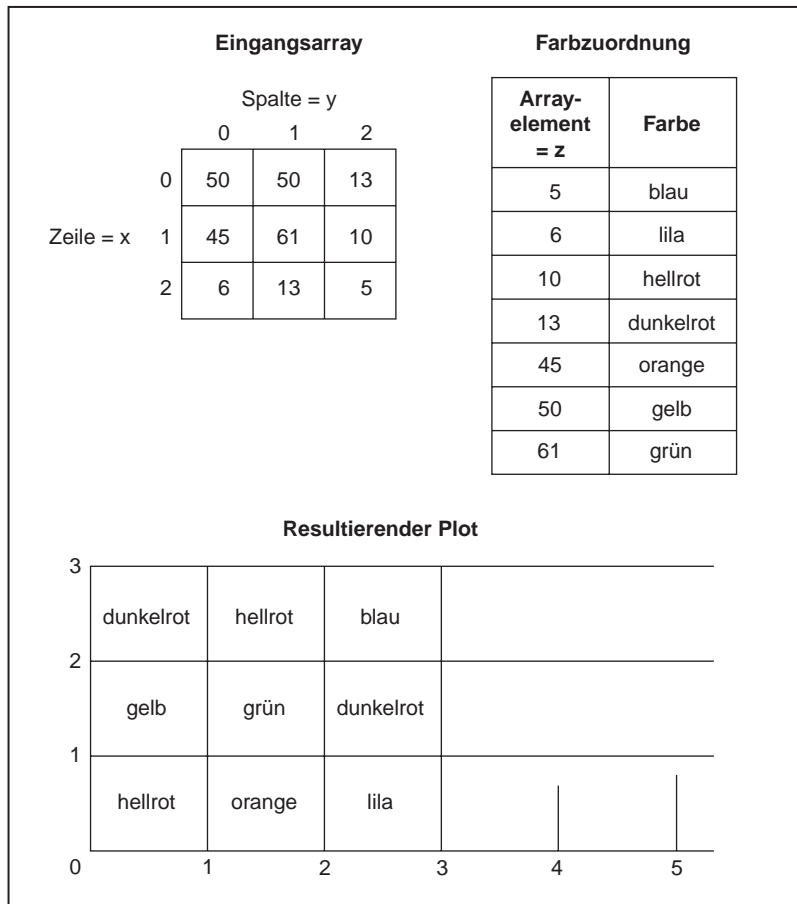
Unterhalb der zweiten gestrichelten Linie dieses Menüs befindet sich eine Liste der Plots, in denen der Cursor fixiert werden kann (zum Beispiel Plot 0, Plot 1, Plot 2 usw.).

Es gibt eine große Anzahl von Menüpunkten, mit denen Cursor oder Marker eines Graphen mit Hilfe von Attributknoten programmatisch erstellt, gesteuert und gelesen werden können. Weitere Informationen hierzu finden Sie in Kapitel 22, [Attributknoten](#).

Intensitätsdiagramme

Das Intensitätsdiagramm ist eine Methode der Darstellung von drei Dimensionen in einem zweidimensionalen Plot, indem Farbblocks auf eine kartesische Ebene plaziert werden. Das Intensitätsdiagramm nimmt ein zweidimensionales Array aus Zahlen an. Jede Zahl im Array repräsentiert eine bestimmte Farbe. Die Indizes eines Elements im zweidimensionalen

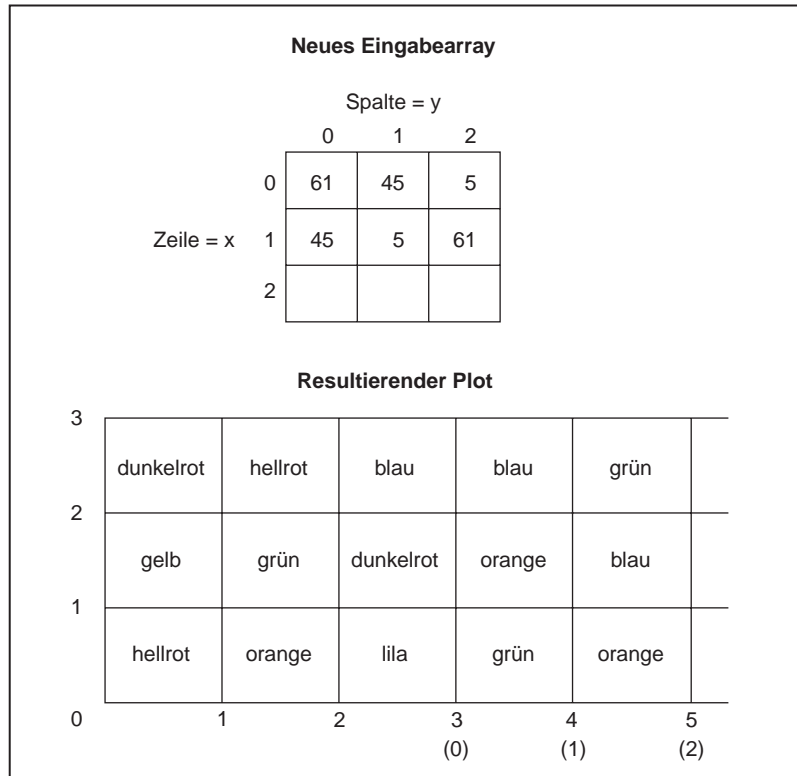
Array stellen die Plotposition für diese Farbe ein. Die folgende Abbildung veranschaulicht das Anwendungskonzept des Intensitätsdiagramms.



Sie können die Farben für das Intensitätsdiagramm mit Hilfe der Farbskala interaktiv definieren, oder Sie können sie mit dem Diagramm-Attributknoten programmatisch definieren. Der Abschnitt *Farbzuordnung* weiter unten in diesem Kapitel erläutert das Verfahren, mit dem einer Farbe eine Nummer zugeordnet werden kann.

Die Array-Indizes entsprechen dem linken unteren Vertex des Farbfeldes. Das Feld besitzt einen Einheitenbereich, der durch die Array-Indizes definiert ist. Das Intensitätsdiagramm kann bis zu 256 verschiedene Farben darstellen.

Sobald ein Datenblock geplottet worden ist, wird der Ursprung der kartesischen Ebene zur Rechten des vorherigen Datenblocks verschoben. Werden neue Daten an das Intensitätsdiagramm gesendet, erscheinen die neuen Daten rechts von den vorherigen Daten (siehe nachstehende Abbildung).



Ist die Diagrammanzeige voll, werden die ältesten Daten links aus dem Diagramm gerollt.

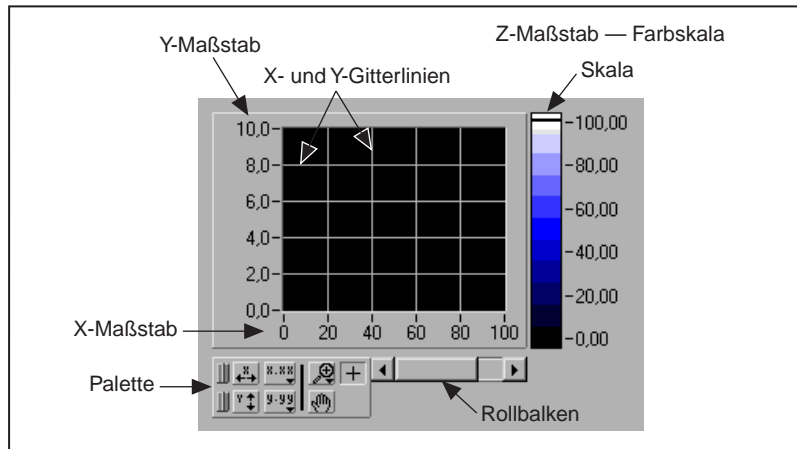
Beispiele für Intensitätsdiagramme finden Sie unter `examples\general\graphs\intgraph.llb`.

Optionen für Intensitätsdiagramme

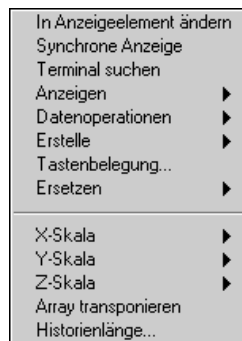
Das Intensitätsdiagramm teilt viele der optionalen Bestandteile der anderen Diagramme, von denen die meisten aus dem Untermenü **Anzeigen** des Popup-Menüs des Graphen angezeigt oder verborgen werden können. Zu diesen Menüpunkten gehört eine Palette, mit der Sie während der VI-Ausführung die Skala und das Format ändern können. Da das

Intensitätsdiagramm außerdem eine dritte Dimension (Farbe) aufweist, definiert eine Skala, ähnlich der Farbrampensteuerung, den Bereich der Zuordnungen von Werten zu Farben.

Die folgende Abbildung zeigt ein Diagramm mit all diesen optionalen Komponenten.



Intensitätsdiagramme verfügen über eine Reihe von Menüpunkten zur Anpassung der Datendarstellung. Die folgende Abbildung zeigt das Pop-up-Menü des Intensitätsdiagramms.



Die meisten dieser Menüpunkte sind mit denen für das Kurvendiagramm identisch. Mit dem Menü **Anzeigen** können Sie die Farbskala für die z-Skala anzeigen und verbergen. Die Menüs **X-Skala** und **Y-Skala** sind mit den entsprechenden Menüs für das Kurvendiagramm identisch.

Das Intensitätsdiagramm führt eine Historie mit Daten vorheriger Aktualisierungen. Sie können diesen Puffer durch Wählen von

Historienlänge... aus dem Popup-Menü des Diagramms konfigurieren. Die Standardgröße für ein Intensitätsdiagramm ist 128 Punkte. Beachten Sie, daß die Darstellung des Intensitätsdiagramms sehr speicherintensiv sein kann. Um beispielsweise ein Single-Diagramm mit einer Historie von 512 Punkten und 128 y-Werten darzustellen, sind $512 * 128 * 4$ Bytes (Single-Größe) bzw. 256 Kilobytes erforderlich. Wenn Sie große Mengen von Daten in einem Intensitätsdiagramm darstellen wollen, vergewissern Sie sich, daß ausreichend Speicher zur Verfügung steht.

Das Intensitätsdiagramm unterstützt die standardmäßigen Aktualisierungsmodi für Streifen-, Sweep- und Oszilloskop-Diagramme. Wie beim Kurvendiagramm wählen Sie den Aktualisierungsmodus aus dem Menü **Datenoperationen**.

Farbzuordnung

Sie können die Farbzuordnung interaktiv auf dieselbe Art und Weise einstellen wie die Farben für ein numerisches Bedienelement vom Typ "Farbrampe". Weitere Einzelheiten finden Sie im Abschnitt *Farbrampe* in Kapitel 9, *Numerische Bedien- und Anzeigeelemente*.

Bei Verwendung des Attributknotens gibt es zwei Möglichkeiten, die Farben programmatisch einzustellen. Sie können erstens die Zuordnungen Wert-zu-Farbe dem Attributknoten in derselben Weise angeben wie bei der Farbskala. Bei dieser Methode geben Sie das Attribut **Z Scale Info: Color Array** (Z-Skala Info: Array aus Farben) an. Dieses Attribut besteht aus einem Array aus Clustern, in dem jeder Cluster einen numerischen Grenzwert sowie die entsprechende anzuzeigende Farbe zu diesem Wert aufweist. Wenn Sie die Farbtabelle in dieser Weise spezifizieren, können Sie mit dem Attribut **Z Scale Info: High Color** (Z-Skala Info: Obere Farbe) eine obere Farbe außerhalb des Bereichs und mit dem Attribut **Z Scale Info: Low Color** (Z-Skala Info: Untere Farbe) eine untere Farbe außerhalb des Bereichs angeben. Die Gesamtanzahl der Farben ist auf 254 beschränkt, und mit der unteren und oberen Farbe außerhalb des Bereichs ergeben sich insgesamt 256 Farben. Wenn Sie mehr Farben angeben, wird die 254-Farben-Tabelle durch Interpolation der angegebenen Farben erzeugt.

Die zweite Methode, Farben programmatisch einzustellen, besteht darin, eine Farbtabelle mit dem Attribut **Color Table** (Farbtabelle) anzugeben. Bei dieser Methode können Sie ein Array aus bis zu 256 Farben angeben. An das Diagramm übergebene Daten werden Indizes auf der Basis der Farbskala zugeordnet. Wenn sich die Farbskala in einem Bereich zwischen Null und 100 bewegt, wird ein Wert Null dem Index Eins und ein Wert 100 dem Index 254 zugeordnet, wobei die dazwischenliegenden Werte

zwischen 1 und 254 interpoliert werden. Werte unter Null werden der unteren Farbe außerhalb des Bereichs (Index Null) und Werte über 100 werden der oberen Farbe außerhalb des Bereichs (Index 255) zugeordnet.



Hinweis

Die Farben, die das Intensitätsdiagramm (oder den -graph) darstellen soll, beschränken sich auf die exakten Farben und die exakte Anzahl der Farben, die Ihre Graphikkarte anzeigen kann. Eine weitere Beschränkung ist durch die für Ihre Anzeige reservierte Anzahl von Farben gegeben.

Intensitätsgraphen

Der Intensitätsgraph und das Intensitätsdiagramm funktionieren im wesentlichen gleich, mit der Ausnahme, daß ersterer die vorherigen Daten nicht beibehält. Bei jeder Übergabe neuer Daten werden die alten Daten mit dem Eingang der neuen Daten ersetzt.

Ein Beispiel für einen Intensitätsgraphen finden Sie in `examples\graphs\intgraph.llb`.

Datentypen für Intensitätsgraphen

Der Intensitätsgraph nimmt zweidimensionale Arrays aus Zahlen an, wobei mit dem Diagramm jede Zahl einer Farbe zugeordnet wird.

Zeilen der eingehenden Daten werden im Diagramm als neue Spalten dargestellt. Sollen die Zeilen als Reihen erscheinen, wählen Sie den Menüpunkt **Array transponieren** aus dem Popup-Menü des Diagramms.

Optionen von Intensitätsgraphen

Der Intensitätsgraph funktioniert mit der Ausnahme, daß er keine Diagramm-Aktualisierungsmodi besitzt, im wesentlichen wie das Intensitätsdiagramm. Da die vorherigen Daten bei jeder Aktualisierung ersetzt werden, steht weder ein Rollbalken zur Verfügung, noch sind Historienoptionen gegeben.

Der Intensitätsgraph kann wie andere Graphen Cursor haben. Jeder Cursor zeigt die x -, y - und z -Werte zu einem angegebenen Punkt im Graphen an. Weitere Informationen zur Manipulation der Cursor eines Graphen finden Sie im Abschnitt [Cursor des Graphen](#) in diesem Kapitel.

Die Farbuordnung wird wie beim Intensitätsdiagramm eingestellt.

ActiveX-Bedienelemente

Dieses Kapitel beschreibt die Funktionalität von ActiveX-Containern, die die Kommunikation zwischen G-basierter Software und anderen Anwendungen verbessert.

ActiveX-Erweiterungen des Frontpanels

Das Frontpanel verfügt über eine **ActiveX**-Unterpalette, die zwei ActiveX-Bedienelemente enthält, den ActiveX-Container und den ActiveX-Datentyp (siehe nachstehende Abbildung).



Bedien- und Anzeigeelement vom Typ “ActiveX-Datentyp”

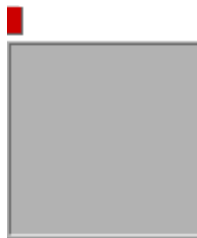
Das Bedien- und Anzeigeelement vom Typ “ActiveX-Datentyp” ermöglicht es Ihnen, ActiveX-Daten an die Software zu übergeben, wodurch die Funktionalität von ActiveX-Clients verbessert wird. Wenn Sie das Bedien- und Anzeigeelement vom Typ “ActiveX-Datentyp” in das Blockdiagramm plazieren, erscheint dieses wie folgt.



Benutzen Sie dieses Frontpanel-Objekt, wenn ActiveX-Daten in Anzeigedaten konvertiert werden.

ActiveX-Container

Der ActiveX-Container erlaubt neben integrierten Bedienelementen die Integration von ActiveX-Objekten in ein VI-Panel. Sie können diesen Container dazu verwenden, ActiveX-Bedienelemente und eingebettete Dokumente am Frontpanel anzuzeigen. Die folgende Abbildung zeigt den Container, wie er anfangs auf das Frontpanel plaziert wird.



Der ActiveX-Container erscheint im Blockdiagramm als Automation-Refnum-Terminal. Sie können dieses Terminal mit Automationsfunktionen verbinden und somit das in den Container eingebettete Objekt steuern. Besitzt das Objekt keine Automationschnittstelle, ist die Refnum des Terminals ungültig, und das Terminal ist bei den Automationsfunktionen nicht anwendbar.

Um ein Objekt in den Frontpanel-Container einzufügen, rufen Sie das Popup-Menü auf, und wählen Sie **ActiveX-Objekt einfügen**. Das Dialogfeld **ActiveX-Objekt auswählen** erscheint.

Es gibt zwei allgemeine Arten von möglichen Objekten, die in Containern enthalten sein können, und zwar ActiveX-Dokumente und ActiveX-Controls.

ActiveX-Dokumente — Diese Objekte können im Containerobjekt enthalten sein und durch Aufrufen des Popup-Menüs und Wählen der Option **Objekt bearbeiten** bearbeitet werden. Hierdurch wird zum Bearbeiten des Objekts ein neues Fenster geöffnet. Manche Dokumente unterstützen eine Automationschnittstelle und können im Diagramm in Automationsfunktionen verwendet werden.

ActiveX-Controls — Diese Objekte können innerhalb des Containerobjekts selbst aktiviert und ausgeführt werden. Sie besitzen eine Automationschnittstelle und können im Diagramm mit Automationsfunktionen gesteuert werden.

Objekte können auf dreierlei Art abgelegt werden. Die Aufzählung entspricht der Auswahl oben im Dialogfeld.

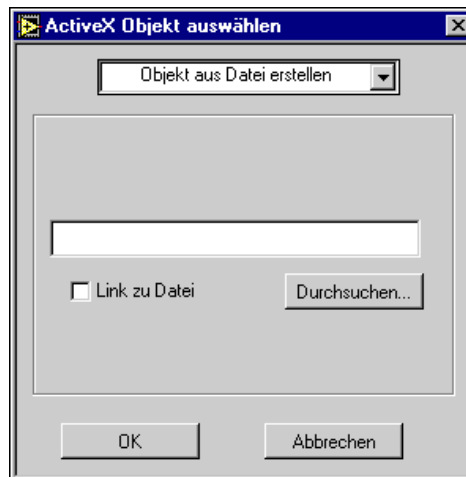
Dokument erstellen — Wählen Sie einen der im System registrierten Dokumenttypen.

Objekt aus Datei erstellen — Wählen Sie ein an einer beliebigen Speicherstelle im Dateisystem befindliches Dokument. Das Objekt kann entweder mit der Datei verknüpft oder statisch auf das Panel kopiert werden.

Bedienelement erzeugen — Wählen Sie ein ActiveX-Bedienelement, das im System registriert ist.



Zwei Arten von Objekten, Bedienelemente und Dokumente, können in den Container eingefügt werden. Sie können hierbei ein neues Bedienelement oder Dokument erstellen oder ein bestehendes Dokument einfügen. Um ein neues Bedienelement oder Dokument erstellen, wählen Sie **Dokument erstellen** (siehe Abbildung oben) und anschließend den gewünschten Objekttyp aus der Objektliste. Um ein bestehendes Dokument oder eine Datei einzufügen, wählen Sie **Objekt aus Datei erstellen**, wodurch sich das Dialogfeld wie folgt ändert.



Benutzen Sie die Taste **Durchsuchen...**, um Dokumente zum Einfügen zu suchen. Wenn Sie die Option **Link zu Datei** wählen, wird das Dokument stets aktualisiert, wenn das Frontpanel-Objekt aktualisiert wird. Wenn Sie das Kontrollkästchen **Link zu Datei** nicht wählen, wird eine statische Version des Dokuments eingefügt.

Um ein bestehendes Bedienelement einzufügen, wählen Sie **Bedienelement erzeugen**, wodurch sich das Dialogfeld wie folgt ändert.



Die verfügbaren Bedienelementtypen sind im System registriert.

Erstellen von ActiveX-Paletten

Unter **Projekt>ActiveX-Controls importieren...** können Sie auch einen Satz von ActiveX-Bedienelementen benutzerspezifische Bedienelemente konvertieren und diese den Palettenmenüs hinzufügen.

Wenn der Menüpunkt gewählt wird, wird eine Liste mit Bedienelementen im System aufgerufen. Sie können dann ein oder mehrere Bedienelemente wählen. Sie werden aufgefordert, ein Verzeichnis oder eine .lib-Datei zu wählen, in dem/der die benutzerspezifischen Bedienelemente gespeichert werden sollen. Das Standardablageziel ist `user.lib`, da alle Dateien und Verzeichnisse in diesem Verzeichnis automatisch in den Palettenmenüs erscheinen.

Blockdiagramm-Programmierung

Dieser Abschnitt enthält Informationen über die Komponenten, die zum Erstellen und Manipulieren eines Blockdiagramms in G erforderlich sind.

Teil III, *Blockdiagramm-Programmierung*, enthält folgende Kapitel.

- Kapitel 17, *Einführung in das Blockdiagramm*, beschreibt Terminals und Knoten — zwei der drei Elemente, aus denen Sie ein Blockdiagramm erstellen. Das dritte Element, die Verbindung, wird in Kapitel 18, *Verbinden des Blockdiagramms*, besprochen.
- Kapitel 18, *Verbinden des Blockdiagramms*, erläutert das Vorgehen beim Verbinden von Terminals im Blockdiagramm.
- Kapitel 19, *Strukturen*, beschreibt die Verwendung der For-Schleife, While-Schleife, Case-Struktur und Sequenzstruktur. Diese Strukturen befinden sich in der Palette **Funktionen»Strukturen**.
- Kapitel 20, *Formelknoten*, beschreibt die Verwendung von Formelknoten zur Ausführung von mathematischen Formeln im Blockdiagramm. Der Formelknoten steht in der Palette **Funktionen»Strukturen** zur Verfügung.
- Kapitel 21, *VI-Server*, beschreibt den Mechanismus zur programmatischen Steuerung von VIs und Anwendungen. Das Kapitel beschreibt außerdem das Verfahren der Remote-Steuerung von VIs oder Anwendungen.

- Kapitel 22, *Attributknoten*, beschreibt die Verwendung von Attributknoten, um Attribute von Frontpanel-Bedienelementen programmatisch zu setzen und zu lesen. Einige nützliche Eigenschaften umfassen Anzeigefarben, die Sichtbarkeit von Bedienelementen, Menüstrings für ein Ring-Bedienelement, Plotfarben für Graphen oder Diagramme und Cursor für Graphen.
- Kapitel 23, *Globale und lokale Variablen*, beschreibt, wie globale und lokale Variablen definiert und verwendet werden. Verwenden Sie globale Variablen, um von mehreren VIs aus leicht auf einen bestimmten Satz von Werten zuzugreifen. Lokale Variablen dienen innerhalb eines einzigen VI einem ähnlichen Zweck.

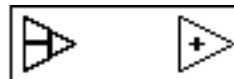
Einführung in das Blockdiagramm

Dieses Kapitel beschreibt Terminals und Knoten — zwei der drei Elemente, aus denen Sie ein Blockdiagramm erstellen. Das dritte Element, die Verbindung, wird in Kapitel 18, *Verbinden des Blockdiagramms*, besprochen.

Terminals und Knoten

Blockdiagramme werden aus Terminals, Knoten und Verbindungen erstellt.

Terminals sind Anschlüsse, über die Daten zwischen dem Blockdiagramm und dem Frontpanel sowie zwischen Knoten im Blockdiagramm ausgetauscht werden. Sie liegen auch den Icons von Funktionen und VIs zugrunde. Die folgende Abbildung zeigt ein Beispiel eines Terminal-Pattern auf der linken Seite und seinem entsprechenden Icon auf der rechten Seite. Um die Terminals für eine Funktion oder ein VI anzuzeigen, rufen Sie das Popup-Menü des Icons auf, und wählen Sie **Anzeigen» Terminals**.



Knoten sind Elemente der Programmausführung. Sie verhalten sich analog zu Anweisungen, Operatoren, Funktionen und Unterprogrammen in herkömmlichen Programmiersprachen.

Verbindungen sind die Datenpfade zwischen Eingangs- und Ausgangsterminals.

Terminals

G verfügt über viele Arten von Terminals. Ein Terminal ist im allgemeinen ein beliebiger Punkt, an dem eine Verbindung angebracht werden kann. G besitzt Terminals für Bedien- und Anzeigeelemente, Knotenterminals, *Konstanten* und Spezialterminals in Strukturen.

Terminals, die Daten zuführen, wie z.B. Bedienelement-Terminals von Frontpanels, Knoten-Ausgabeterminals und Konstanten, werden auch als *Quellterminals* bezeichnet. Knoten-Eingabeterminals und Anzeigeelement-Terminals von Frontpanels werden auch als *Ziel-* oder *Eingangsterminals* bezeichnet, da sie die Daten empfangen.

Terminals für Bedien- und Anzeigeelemente

Sie geben Werte in die Bedienelemente eines Frontpanels ein, und diese werden bei der Ausführung eines VIs von den Bedienelement-Terminals an das Blockdiagramm übergeben. Bei Abschluß der VI-Ausführung werden die Daten vom Blockdiagramm via Anzeigeelement-Terminals an die Anzeigeelemente des Frontpanels übergeben.

Tabelle 17-1, *Terminalsymbole für Bedien- und Anzeigeelemente in G*, zeigt die Symbole für einige der Terminals von Bedien- und Anzeigeelementen in G. Jedes Symbol weist ein Bild und eine Farbe auf, welche auf den Datentyp des Bedien- oder Anzeigeelements und im Falle von numerischen Objekten auch auf die Darstellung hinweisen. Terminals von Bedienelementen haben eine dickere Umrandung als die von Anzeigeelementen. Da ein Terminal zu seinem entsprechenden Bedien- oder Anzeigeelement gehört, kann es nicht ohne weiteres gelöscht werden; möchten Sie ein Bedien- oder Anzeigeelement löschen, tun Sie dies über das Frontpanel.

Ein Arrayterminal umfaßt einen der in eckigen Klammern angegebenen Datentypen und nimmt die Farbe des Datentyps eines Arrayelements an.

Tabelle 17-1. Terminalsymbole für Bedien- und Anzeigeelemente in G









































Bedien- element	Anzeige- element	Beschreibung	Farbe
		Extended Fließkommazahl	Orange
		Double Fließkommazahl	Orange
		Single Fließkommazahl	Orange

Tabelle 17-1. Terminalsymbole für Bedien- und Anzeigeelemente in G (Fortsetzung)

Bedien- element	Anzeige- element	Beschreibung	Farbe
		(Extended) komplexe Fließkommazahl	Orange
		(Double) komplexe Fließkommazahl	Orange
		(Single) komplexe Fließkommazahl	Orange
		Vorzeichenloser 32-Bit Integer-Wert	Blau
		Vorzeichenloser 16-Bit Integer-Wert	Blau
		Vorzeichenloser 8-Bit Integer-Wert	Blau
		Vorzeichenloser 32-Bit Integer-Wert (long Word)	Blau
		16-Bit-Integer-Wert (Word)	Blau
		8-Bit-Integer-Wert	Blau
		Cluster	Braun oder rosa
		Array	Verschieden
		Pfad	Aquamarin
		Refnum	Aquamarin
		Boolesch	Grün
		String	Rosa
		Enum	Blau
		OLE-Datentyp	Lila

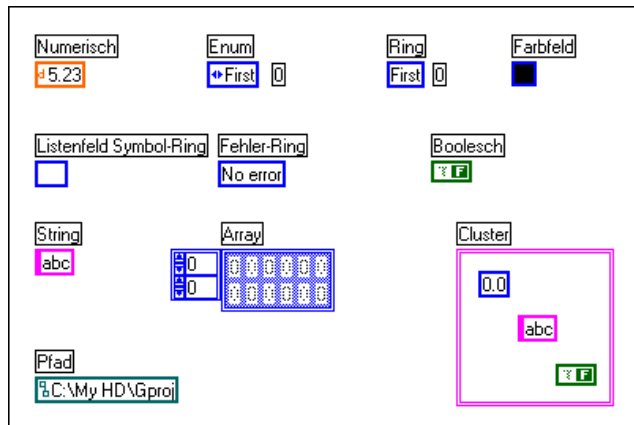
Konstanten

Konstanten sind Terminals in einem Blockdiagramm, die Datenwerte direkt an das Blockdiagramm übergeben. *Benutzerdefinierte Konstanten* werden vor der Programmausführung durch Bearbeiten der Konstante definiert, und ihre Werte können während der Ausführung nicht geändert werden. *Universelle Konstanten* haben feste Werte.

Benutzerdefinierte Konstanten

Die einfachste Art, eine benutzerdefinierte Konstante zu erstellen, besteht darin, das Popup-Menü eines Ein- oder Ausgangs aufzurufen und **Konstante erzeugen** zu wählen. Diese Konstanten stehen je nach dem Typ auch in verschiedenen Unterpalletten der Palette **Funktionen** zur Verfügung. Die meisten sind unten oder oben in der relevanten Palette zu finden, zum Beispiel sind die numerischen, Enum- und Ring-Konstanten in der untersten Reihe der Palette **Numerisch** zu finden. Drei Konstanten — Farbfeld-, Listensymbol- und Fehlerfeld-Konstanten — sind in der Palette **Numerisch»Zusätzliche Num. Konstanten** zu finden. Die Pfadkonstante befindet sich in der Unterpalette **Datei I/O»Dateikonstanten**.

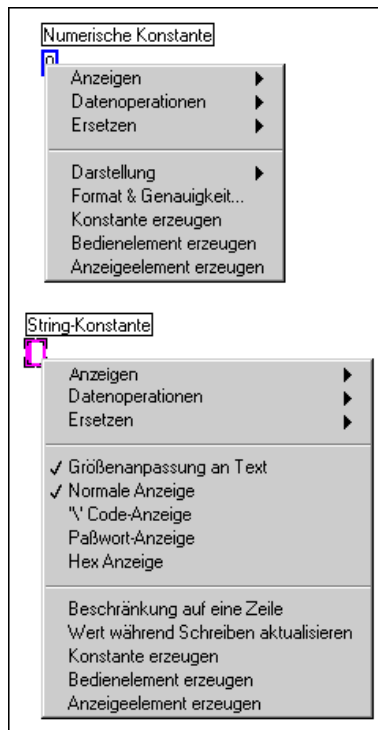
Die Konstanten sind in der nachstehenden Abbildung zu sehen.



Sie können die Konstante beschriften, indem Sie das Popup-Menü der Konstante aufrufen und **Anzeigen»Label** auswählen. Es erscheint ein hervorgehobenes Textfeld, wodurch angezeigt wird, daß Sie sofort Text eingeben können, ohne das Werkzeug zu wechseln.

Wie bei Labeln wird die Größe von benutzerdefinierten Konstanten während der Eingabe automatisch an den Textinhalt angepaßt. Wenn Sie die Größe oder Form einer Label- oder String-Konstante ändern, können Sie aus deren Popup-Menü den Menüpunkt **Größenanpassung an Text** wählen, so daß die Größe der Konstante oder des Labels automatisch an deren Inhalt angepaßt wird.

Mit dem Bedienwerkzeug stellen Sie den Wert einer benutzerdefinierten Konstante wie den eines digitalen Bedienelements, eines Booleschen Schiebeschalters oder eines String-Bedienelements am Frontpanel ein. Die *numerischen Konstanten* und *String-Konstanten* ähneln numerischen Bedienelementen von Frontpanels und besitzen Popup-Menüs, die den Popup-Menüs von Bedienelementen ähnlich sind (siehe nachfolgende Abbildung).



Sie können die Pfeiltasten dazu verwenden, den Wert einer neuen numerischen Konstante oder den gewählten Wert einer bestehenden Konstante um eins zu erhöhen oder verringern. Dies ist insbesondere beim Programmieren von Bedienelement-Parametern mit niedrigen Werten wie 1, 2 oder 3 nützlich.

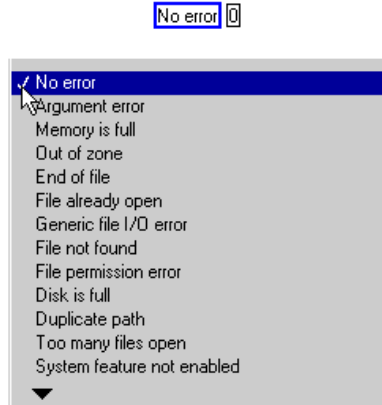
Die *Ring-Konstante* assoziiert wie ein Ring-Bedienelement auf dem Frontpanel Text mit einer Zahl. Der Wert der Ring-Konstante ist ein vorzeichenloser 16-Bit-Integer-Wert. Obgleich Sie die Darstellung einer Ring-Konstante in einen beliebigen numerischen Typ (ausgenommen komplexer Typen) ändern können, ist der Wert doch stets ein Integer-Wert im Bereich von Null bis $n - 1$.

Die *Aufzählungskonstante* ist der Ring-Konstante ähnlich, mit der Ausnahme, daß die mnemonischen Strings (mit einem Integer-Wert assoziierte Strings) als Teils des Typs betrachtet werden. Wenn die Aufzählung mit dem Auswahlterminal einer Case-Struktur verbunden ist, werden Cases entsprechend der mnemonischen Strings der Aufzählung und nicht der herkömmlichen numerischen Werte benannt. Die numerische Darstellung einer Aufzählung ist stets vorzeichenloses Byte, Word oder Long.

Sie können aus der *Farbfeld-Konstante* Farben wählen, um sie beim Farbfeld-Bedienelement, einem Bedienelement, dessen Werte in Korrelation zu bestimmten Farben stehen, zu verwenden. Stellen Sie das Farbfeld ein, indem Sie mit dem Färbewerkzeug oder dem Bedienwerkzeug darauf klicken und die gewünschte Farbe aus der Farbpalette wählen.

Die *Listensymbol-Konstante* wird bei der Zuweisung von Symbolen zu Objekten in einer Listenfeldsteuerung verwendet.

Die *Fehlerfeld-Konstante* ist ein vordefinierter Ring. Sie klicken mit dem Bedienwerkzeug auf die Konstante und wählen die gewünschte Fehlermeldung aus dem daraufhin angezeigten Dialogfeld (siehe nachfolgende Abbildung). Dieser Ring ist nützlich bei Funktionen, die Fehlercodes oder Fehlercluster zurückgeben, wie z.B. die Datei I/O-Funktionen, da das Blockdiagramm dadurch deskriptiver gestaltet wird. Wenn Sie beispielsweise versuchen, mit der Funktion "Datei öffnen" eine nicht existierende Datei zu öffnen, gibt die Funktion einen Fehlercode zurück. Sie können diesen Zustand testen, indem Sie die Fehlercodeausgabe eines Fehler-Rings, der auf den Wert von "Datei nicht gefunden" (7) eingestellt ist, überprüfen.



Sie können die *Pfadkonstante* zum Erstellen eines konstanten Pfadwerts im Blockdiagramm verwenden.

Die Pfad-, String- und Farbfeldkonstanten sind in der Größe veränderbar, die Booleschen, numerischen, Ring- und Aufzählungskonstanten jedoch nicht. Eine numerische Konstante paßt die Darstellung an die Quelle an, die Sie eingeben. Wenn Sie z.B. 1, 1 eingeben, wird die Konstante zu DBL. Dies wird durch die Einstellung **An Quelle angepaßt** im Popup-Menü **Darstellung** gesteuert.

Die Darstellung “An Quelle angepaßt” gibt an, ob eine numerische Konstante ihre Darstellung automatisch bestimmt, wenn Sie einen neuen Wert eingeben. Numerische Konstanten von Blockdiagrammen werden standardmäßig auf “An Quelle angepaßt” eingestellt, wenn Sie eine Konstante erstellen. Wenn Sie z.B. 2 in eine numerische Konstante DBL eingeben, ändert diese ihren Typ automatisch in I32. Wenn Sie die Darstellung einer Konstanten über das Popup-Menü ändern, wird **An Quelle angepaßt** implizit ausgeschaltet, damit der angegebene Typ der Konstanten beibehalten wird. Wenn Sie das alte Verhalten der automatischen Typenbestimmung anhand des eingegebenen Werts wiederherstellen wollen, müssen Sie aus dem Popup-Menü der Darstellung **An Quelle angepaßt** wählen.

Die Standarddarstellung der numerischen Konstanten ist eine Double-Fließkommazahl, wenn Sie eine Fließkommazahl eingeben, ein Long-Integer-Wert, wenn Sie einen Integer-Wert eingeben, oder eine komplexe Double-Zahl, wenn Sie eine komplexe Zahl eingeben. Wenn Sie beispielsweise '123' eingeben, ist die Darstellung ein Long-Integer-Wert, und wenn Sie '123' . eingeben, eine Double-Fließkommazahl. Sie können

die Darstellung mit dem Menüpunkt **Darstellung** aus dem Popup-Menü der Konstanten ändern.

Universelle Konstanten

Es gibt zwei Typen von universellen Konstanten: universelle numerische Konstanten und universelle String-Konstanten.

- Universelle numerische Konstanten — Satz präziser und allgemein gebräuchlicher mathematischer und physikalischer Werte, wie z.B. Pi und die Lichtgeschwindigkeit.
- Universelle String-Konstanten — Satz allgemein gebräuchlicher nicht darstellbarer Stringzeichen, wie z.B. Zeilenvorschub und Wagenrücklauf (Carriage Return).

Weitere Informationen über diese Konstanten finden Sie in der *Online-Referenz* von LabVIEW oder BridgeVIEW, die Sie durch Wählen von **Hilfe»Online-Referenz»Funktion und VI Referenz** erhalten.

Knoten

Knoten sind die Ausführungselemente eines Blockdiagramms. Die sechs Knotentypen sind *Funktionen*, *SubVIs*, *Strukturen*, *Code Interface Nodes* (CINs), *Formelknoten* und *Attributknoten*. Funktionen und SubVI-Knoten haben ähnliche Erscheinungsformen und Rollen in einem Blockdiagramm, aber sie weisen erhebliche Unterschiede auf. Funktionen werden im Abschnitt *Funktionen* dieses Kapitels besprochen. SubVIs werden in Kapitel 3, *Verwendung von SubVIs*, besprochen.

Strukturen, die das G-Datenflußmodell der Programmierung ergänzen, um die Ausführungsreihenfolge zu steuern, werden kurz im Abschnitt *Strukturen* dieses Kapitels besprochen. Ausführliche Informationen hierzu finden Sie in Kapitel 19, *Strukturen*.

CINs sind Schnittstellen zwischen dem Blockdiagramm und dem Code, den Sie in herkömmlichen Programmiersprachen, wie z.B. C oder Pascal, schreiben. Weitere Informationen hierzu entnehmen Sie dem Handbuch *LabVIEW Code Interface Reference Manual*, das im PDF-Format auf den Programmdisketten oder der CD vorhanden ist.

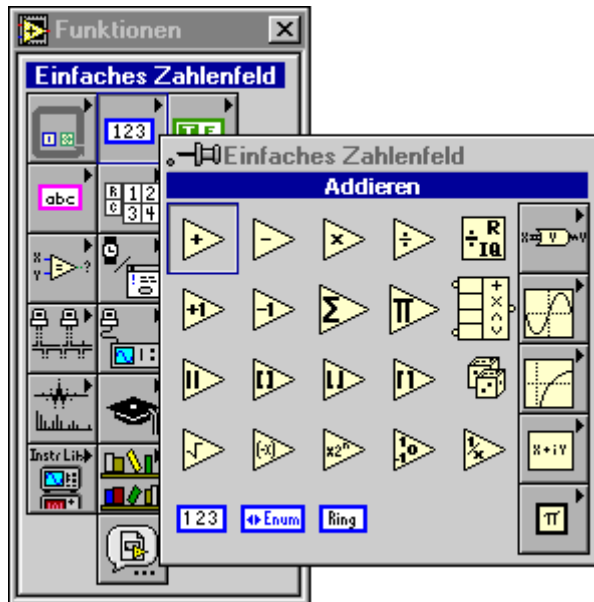
Formelknoten, die durch die Verwendung von Formeln im Blockdiagramm G-Funktionen ergänzen, werden in Kapitel 20, *Formelknoten*, besprochen.

Attributknoten, die Attribute programmatisch steuern, werden in Kapitel 22, *Attributknoten*, besprochen.

Funktionen

Funktionen sind elementare Knoten, die in G integriert sind. Sie führen elementare Operationen durch wie z.B. Addition von Zahlen, Datei I/O und Stringformatierung. Funktionen besitzen weder Frontpanels noch Blockdiagramme. In kompiliertem Zustand generieren sie internen Maschinencode. Ausführliche Beschreibungen einzelner Funktionen finden Sie unter **Online-Referenz»Funktion und VI Referenz**.

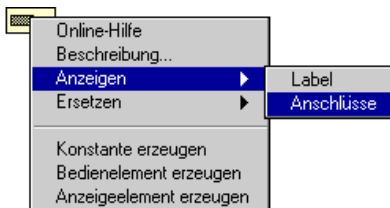
Sie wählen Funktionen aus der Palette **Funktionen**, wie in der nachstehenden Abbildung dargestellt.



Wenn Sie eine Funktion wählen, erscheint deren Icon im Blockdiagramm. Um ihr Label anzuzeigen, rufen Sie das Popup-Menü des Icons auf, und wählen Sie **Anzeigen»Label**. Sie können das Label ändern, indem Sie den Text mit dem Beschriftungswerkzeug hervorheben und Text neu eingeben. Sie können das Label der Funktion dazu verwenden, deren Verwendungszweck im Blockdiagramm zu vermerken.

Sie können das Hilfenfenster verwenden, um zu sehen, wie die Funktion verbunden werden kann, oder Sie können aus dem Popup-Menü des Icons

Anzeigen»Anschlüsse (siehe nachstehende Abbildung) wählen, um zu sehen, wo genau sich die Terminals befinden.

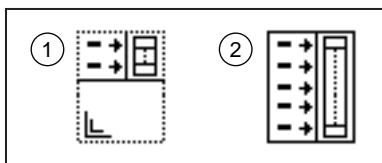


Wenn Terminals angezeigt werden, wählen Sie erneut **Anzeigen»Anschlüsse**, um statt dessen das Icon für die Funktion anzuzeigen.

Die Verbindung einer Funktion erfolgt über einen ihrer Terminals.

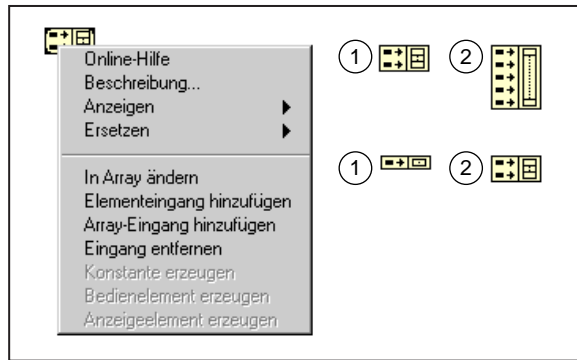
Manche Funktionen haben eine veränderliche Anzahl von Terminals. Wenn Sie z.B. ein Array aus drei Elementen erstellen, benötigt die Funktion "Array erstellen" drei Eingangsterminals, und wenn Sie eines aus 10 Elementen erstellen, benötigt die Funktion 10 Terminals.

Sie können die Anzahl der Terminals der *erweiterbaren* Funktionen durch die Größenanpassung des Icons mit dem Resizing-Cursor in derselben Weise wie andere Objekte ändern (siehe nachstehende Abbildung). Sie können Funktionen vergrößern oder verkleinern, Sie können eine Funktion jedoch nicht verkleinern, wenn dadurch verbundene Terminals entfernt würden.



Sie können die Anzahl der Terminals auch mit den Befehlen **Eingang hinzufügen** und **Eingang entfernen** aus dem Popup-Menü zu einem Terminal ändern (siehe nachfolgende Abbildung). Der Befehl **Eingang entfernen** entfernt das Terminal, dessen Popup-Menü aktiviert wurde, und trennt die gegebenenfalls bestehende Verbindung des Terminals. Der Befehl **Eingang hinzufügen** fügt unmittelbar nach dem aktiven Terminal

ein Terminal ein. Die vollständigen Namen dieser Befehle sind je nach der Funktion verschieden.



Strukturen

Beim Programmieren ist es manchmal erforderlich, bestimmte Codeabschnitte soundso oft oder bei Eintreten (=TRUE) einer bestimmten Bedingung zu wiederholen. In anderen Situationen ist es erforderlich, verschiedene Codeabschnitte bei Vorliegen unterschiedlicher Bedingungen oder in einer bestimmten Reihenfolge auszuführen. G verfügt über spezielle Knoten, sogenannte *Strukturen*. Sie helfen Ihnen dabei, die Codeausführung auf verschiedene Arten zu steuern, die andernfalls im Rahmen des G-Datenflusses nicht möglich sind.

Jede Struktur weist einen distinktiven, in der Größe veränderlichen Rahmen auf, in den Sie den Code einfassen, der unter Anwendung der Sonderregeln der Struktur ausgeführt werden. Aus diesem Grund wird das Diagramm innerhalb einer Strukturumrandung als Subdiagramm bezeichnet. Subdiagramme können ineinander verschachtelt werden.

G benutzt die folgenden Strukturen zur Ausführung von Code.

- For-Schleife — Wiederholt die Ausführung eine vorgegebene Anzahl von Malen.
- While-Schleife — Wiederholt die Ausführung des Subdiagramms, solange eine Bedingung zutrifft (=TRUE).
- Case-Struktur — Enthält mehrere Subdiagramme, von denen je nach dem Wert, der an das Selektor-Terminal übergeben wird, nur eins ausgeführt wird.
- Sequenzstruktur — Führt Code in der numerischen Reihenfolge der Subdiagramme aus.

Da Strukturen Knoten sind, besitzen Sie Terminals, über die sie mit anderen Knoten verbunden sind. Die Terminals, die Daten in und aus Strukturen speisen, werden als *Tunnel* bezeichnet. Tunnel sind in der Position veränderbare Anknüpfungspunkte für Verbindungen von außerhalb und innerhalb der Strukturen. Weitere Informationen hierzu finden Sie in Kapitel 19, *Strukturen*.

Ersetzen und Einfügen von Objekten des Blockdiagramms

Angenommen, Sie verwenden in einem Blockdiagramm die Funktion “Inkrementieren”, und die Funktion “Dekrementieren” ist eigentlich die richtige Wahl. Sie können den Knoten der Funktion “Inkrementieren” löschen, anschließend den Knoten der Funktion “Dekrementieren” aus der Palette **Funktionen** wählen und damit eine neue Verbindung erzeugen. Sie können auch den Menüpunkt **Ersetzen** aus dem Popup-Menü des Inkrement-Knotens verwenden. Durch Wählen von **Ersetzen** greifen Sie auf die Palette **Funktionen** zu, aus der Sie die Funktion “Dekrementieren” wählen können. Der Vorteil dieser Methode besteht darin, daß der neue Knoten den alten ersetzt, ohne daß dies Auswirkungen auf die Verbindung hätte. Eine Funktion kann durch eine beliebige andere Funktion ersetzt werden, es ist allerdings möglich, daß dadurch unverbundene Verbindungen entstehen, falls die Anzahl der Terminals oder Datentypen in jedem Funktionsknoten verschieden ist.

Mit **Ersetzen** kann auch eine Konstante durch eine andere Konstante oder eine Struktur durch eine ähnliche Struktur, z.B. eine While-Schleife durch eine For-Schleife, ersetzt werden.



Hinweis

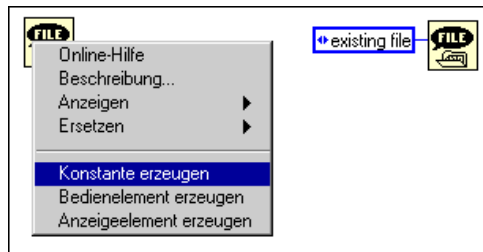
Wenn Sie bei Verwendung des Befehls Ersetzen in Verbindung mit einem SubVI ein VI auswählen, dessen Name dem bereits im Arbeitsspeicher befindlichen VI entspricht, bezieht sich der ersetzte Knoten auf das bereits im Arbeitsspeicher befindliche und nicht das gewählte VI.

Popup-Menüs von Verbindungen verfügen über den Menüpunkt **Einfügen**. Durch Wählen von **Einfügen** greifen Sie auf die Palette **Funktionen** zu, aus der Sie eine beliebige Funktion oder ein beliebiges VI der Palette wählen können. Der Editor spleißt dann den Knoten, den Sie wählen, in die Verbindung, deren Popup-Menü Sie aufgerufen haben. Sie müssen jedoch auf jeden Fall die Verbindung überprüfen, wenn der Knoten mehrere Ein- oder Ausgangsterminals besitzt, da die Verbindungen eventuell nicht zum erwarteten Terminal gehen.

Automatisches Hinzufügen von Konstanten und Bedien- und Anzeigeelementen

Statt eine Konstante oder ein Bedien- oder Anzeigeelement durch Auswahl aus einem Menü und manuelles Verbinden mit einem Terminal zu erstellen, können Sie das Popup-Menü des Terminals aufrufen und entweder **Konstante erzeugen**, **Bedienelement erzeugen** oder **Anzeigeelement erzeugen** wählen, um automatisch ein Objekt mit dem passenden Datentyp zu erstellen. Vorausgesetzt, daß dieses Vorgehen sinnvoll ist, wird hierbei die neu erstellte Konstante bzw. das Bedien- oder Anzeigeelement automatisch für Sie verbunden. Gehört das Terminal zu einer Typendefinition, erstellen Sie eine Konstante oder ein Bedien- oder Anzeigeelement für eine Typendefinition. Wird die Typendefinition später geändert, wird das Bedien- oder Anzeigeelement oder die Konstante dementsprechend aktualisiert.

Wenn Sie beispielsweise eine Konstante für den Eingang **Auswahl-Modus** einer Funktion “Dateidialog” benötigen, können Sie das Popup-Menü des Eingangs aufrufen und **Konstante erzeugen** wählen. Der Editor erstellt für Sie einen Aufzählungstyp, der alle für den Eingang **Auswahl-Modus** zulässigen Werte umfaßt. Vorausgesetzt, daß der Moduseingang noch nicht anderweitig verbunden ist, wird die neue Konstante automatisch verbunden.



Andere nützliche Stellen, an denen Sie die Befehle **Konstante erzeugen**, **Bedienelement erzeugen** oder **Anzeigeelement erzeugen** anwenden können, sind die Ausgänge von Funktionen oder VIs, Verbindungen, Konstanten und Terminals für Bedien- und Anzeigeelemente eines Frontpanels.

Verbinden des Blockdiagramms

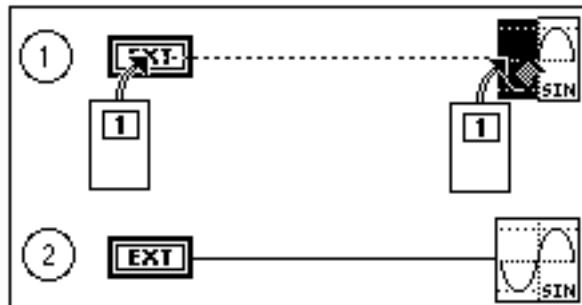
Dieses Kapitel erläutert das Vorgehen beim Verbinden von Terminals im Blockdiagramm.

Verbindungstechniken

Zum Verbinden von Terminals verwenden Sie das Verbindungswerkzeug. Der Cursorpunkt oder Hot Spot des Werkzeugs ist die Spitze des freien Verbindungssegments (siehe nachstehende Abbildung).



Das Symbol links repräsentiert die Maus. Der Pfeil am Ende dieses Maussymbols zeigt in den Abbildungen in diesem Kapitel auf den Bereich, an dem zu klicken ist, und die auf der Maustaste angegebene Zahl zeigt an, wie oft geklickt werden muß.

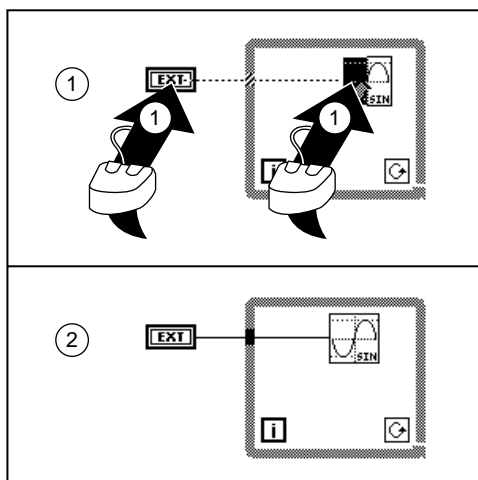


Um zwischen zwei Terminals eine Verbindung zu erzeugen, klicken Sie mit dem Verbindungswerkzeug auf das erste Terminal, verschieben Sie das Werkzeug zum zweiten Terminal, und klicken Sie dann auf das zweite Terminal (siehe vorstehende Abbildung). Halten Sie die Maustaste nicht gedrückt, während Sie die Maus vom ersten zum zweiten Terminal verschieben. Es spielt keine Rolle, auf welches Terminal Sie zuerst klicken. Der Terminalbereich blinkt, wenn der Hot Spot des Verbindungswerkzeugs richtig auf dem Terminal positioniert ist. Durch Klicken auf das Terminal

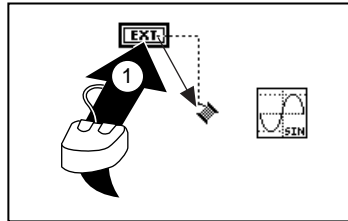
wird die Verbindung erzeugt. Wenn Sie die erste Verbindung erstellen, wird, während Sie den Cursor über das Diagramm verschieben, eine Verbindungslinie zwischen dem Terminal und dem Werkzeug gezeichnet, als würden Sie eine Spule abwickeln. Sie brauchen die Maustaste hierzu nicht gedrückt zu halten.

Um eine Verbindung von einer bestehenden Verbindung aus zu erzeugen, führen Sie die soeben beschriebene Operation durch, wobei Sie die Operation an der bestehenden Verbindung beginnen oder abschließen. Wenn das Verbindungswerkzeug richtig positioniert ist, blinkt die Verbindung, um die neue Verbindung an der bestehenden zu befestigen.

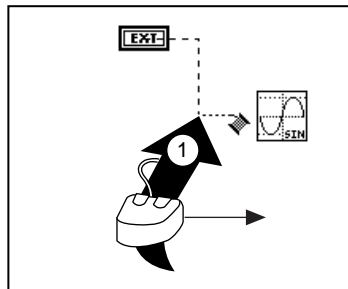
Sie können mit Hilfe der grundlegenden Verbindungsoperation direkt eine Verbindung von einem Terminal außerhalb der Struktur zu einem Terminal innerhalb der Struktur erzeugen. An der Stelle, an der die Verbindung die Strukturgrenze kreuzt, wird automatisch ein *Tunnel* erstellt (siehe nachstehende Abbildung).



Verbindungen werden, je nach der Richtung, in die Sie das Verbindungswerkzeug zuerst verschieben, vertikal oder horizontal von Terminals abgewickelt. Verbindungen werden vertikal verschoben, wenn Sie das Werkzeug aufwärts oder abwärts bewegen, und horizontal, wenn Sie das Werkzeug nach links oder rechts bewegen. Die Verbindung wird, ungeachtet der genauen Position des Hot Spots, wenn Sie darauf die Maustaste klicken, auf den Terminals zentriert (siehe nachfolgende Abbildung).



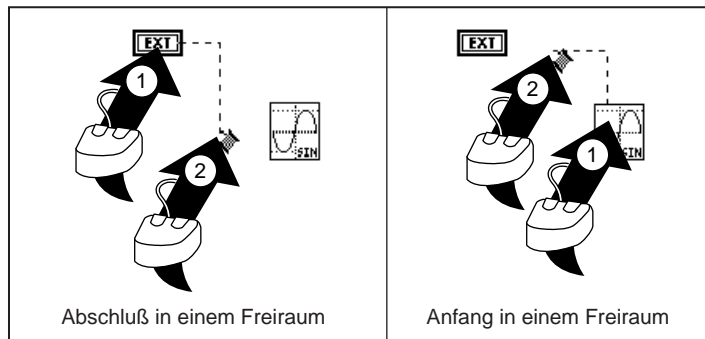
Sie können die Verbindungen ohne zu klicken in einem Winkel von 90 Grad biegen. Sie klicken an einem offenen Bereich, um die Verbindung zu heften und die Richtung zu wechseln (siehe nachstehende Abbildung).



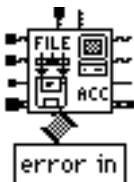
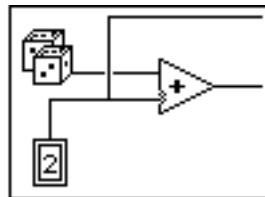
Hinweis

Sie können auch durch Drücken der Leertaste zwischen horizontaler und vertikaler Richtung wechseln. Durch Drücken der <Strg>-Taste (Windows), <Optionstaste> (Macintosh), <mittleren Taste> (UNIX) und Klicken können Sie den letzten Heftpunkt löschen. Ist der letzte Heftpunkt das Terminal oder die Verbindung, auf die Sie zuerst geklickt haben, wird die Verbindung durch Löschen des Heftpunkts komplett entfernt.

Durch Doppelklicken mit dem Verbindungswerkzeug können Sie in einem Freiraum eine Verbindung beginnen oder abschließen (siehe nachfolgende Abbildung).



Bei überkreuzenden Verbindungen wird in der zuerst gezeichneten Linie eine kleine Unterbrechung angezeigt (siehe nachstehende Abbildung), als würde sie unterhalb der zweiten Verbindung verlaufen.

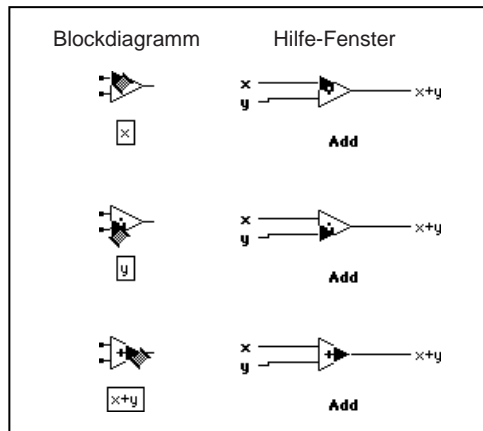


Wenn Sie eine komplizierte Funktion oder ein kompliziertes SubVI verbinden, ist es wichtig, auf die Verbindungsstümpfe und den Tip-Strip zu achten, die angezeigt werden, wenn sich das Verbindungswerkzeug dem VI-Icon nähert.

Verbindungsstümpfe, die abgeschnittenen Verbindungen, die auf der linken Seite um das VI-Icon angezeigt werden, zeigen durch ihren Stil, ihre Stärke und Farbe den Datentyp jedes Terminals an. (Einzelheiten hierzu finden Sie auf der *G-Kurzübersicht*.) Punkte am Ende der Stümpfe weisen auf Eingänge hin. Ausgänge haben keine Punkte. Die Richtung, in die die Stümpfe gezeichnet werden, zeigen die zur Erzeugung übersichtlicher Diagramme empfohlene Verbindungsrichtung an. Wenn Sie ein Terminal verbinden, wird der Verbindungsstumpf für dieses Terminal nicht mehr angezeigt.

Der Tip-Strip, das links abgebildete Feld mit Text, zeigt den Namen des Terminals an, zu dem die Verbindung durch Klicken der Maustaste hergestellt wird.

Sie könnten auch die Funktion des **Hilfe**-Fensters verwenden, die jedes Anschlußfeld-Terminal hervorhebt. Mit dieser Funktion können Sie genau sehen, wo die Verbindung anknüpfen muß. Die drei Verbindungen der Funktion “Hinzufügen” werden in der nachfolgenden Abbildung anhand eines einfachen Beispiels veranschaulicht.

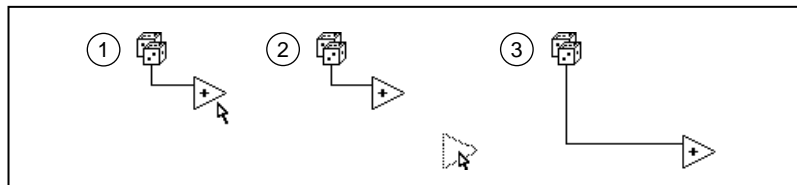


Hinweis

Die oben veranschaulichte Funktion des Hilfe Fensters funktioniert nicht bei Funktionen, die vergrößert werden können. Ein Beispiel hierfür ist die Funktion “Array bündeln”.

Dehnen von Verbindungen

Sie können verbundene Objekte einzeln oder in Gruppen verschieben, indem Sie die gewählten Objekte mit dem Positionierwerkzeug an die neue Position verschieben. Die Verbindungen zu den gewählten Objekten werden automatisch gedehnt. Die Fähigkeit, Verbindungen zu dehnen, wird in der folgenden Abbildung veranschaulicht.



Wenn Sie die gewählten Objekte duplizieren oder von einem Diagramm zu einem anderen verschieben — zum Beispiel vom Blockdiagramm zu einem Struktur-Subdiagramm — bleiben die Verbindungen zurück, sofern Sie sie nicht mit auswählen.

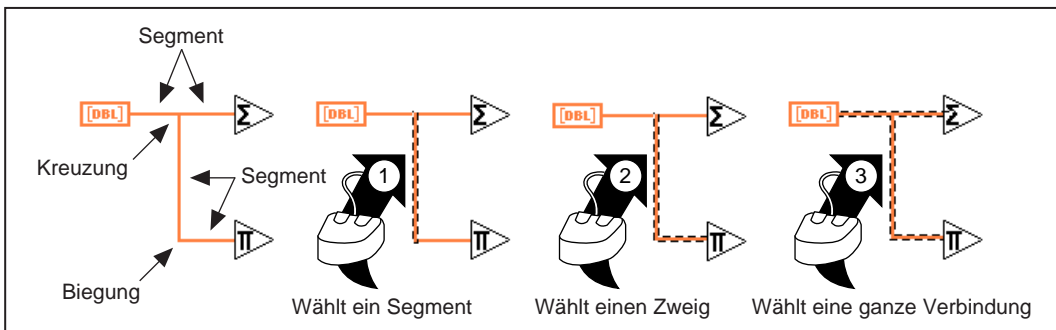


Hinweis

Die Dehnung von Verbindungen erzeugt gelegentlich Verbindungsstümpfe oder offene Verbindungen, auf die wir in den Unterabschnitten Verbindungsstümpfe und Offene Verbindungen des Abschnitts Beheben von Verbindungsproblemen in diesem Kapitel näher eingehen. Sie müssen die Stümpfe oder offenen Verbindungen entfernen, damit das VI ausgeführt werden kann. Die einfachste Methode stellt hier der Befehl Bearbeiten»Ungültige Verbindungen entfernen dar. Hiermit werden auch unnötige Schleifen in Verbindungen entfernt.

Auswählen, Verschieben und Löschen von Verbindungen

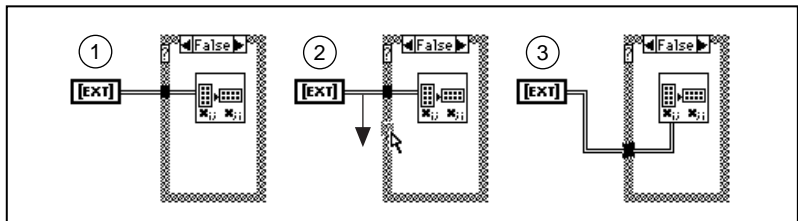
Ein *Segment* einer Verbindung ist ein einzelner horizontaler oder vertikaler Abschnitt einer Verbindung. Der Punkt, an dem drei oder vier Verbindungssegmente zusammentreffen, ist eine *Kreuzung*. Eine *Biegung* in einer Verbindung ist der Punkt, an der zwei Segmente aneinander anknüpfen. Ein *Verbindungszweig* umfaßt alle Verbindungssegmente von Kreuzung, Terminal zu Kreuzung oder Terminal zu Terminal, wenn dazwischen keine Kreuzungen liegen. Ein Mausklick mit dem Positionierwerkzeug auf eine Verbindung wählt ein Segment aus. Durch Doppelklicken wird ein Zweig ausgewählt. Durch Dreifachklicken wird eine gesamte Verbindung ausgewählt. Drücken Sie die <Entf>-Taste oder <Rücktaste>, um den gewählten Teil der Verbindung zu entfernen. Dieser Prozeß wird in der nachfolgenden Abbildung veranschaulicht.



Die Neupositionierung eines verbundenen Objekts kann ein überschüssiges Segment oder ein eigenartig positioniertes Verbindungssegment zur Folge haben. Um ein Verbindungssegment neu zu positionieren, ziehen Sie es mit dem Positionierwerkzeug an die neue Position. Sie können ein oder mehrere Segmente durch Wählen und Ziehen neu positionieren.

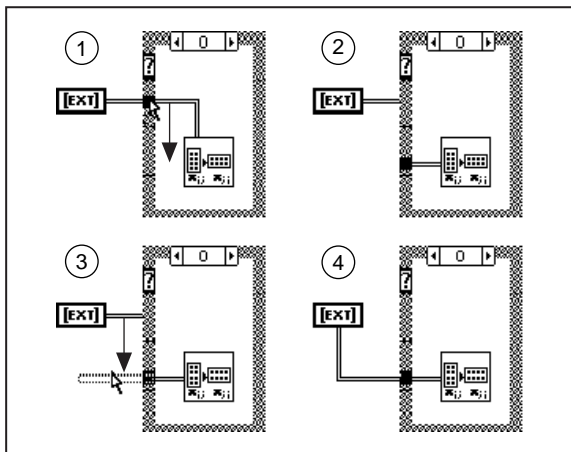
Verwenden Sie die <Umschalt>-Taste, um das Ziehen der Verbindung auf die horizontale oder vertikale Richtung zu beschränken. Die anfängliche Verschiebungsrichtung bestimmt, ob die Verbindung auf horizontale oder vertikale Translation beschränkt ist. Sie können ausgewählte Segmente auch pixelweise verschieben, indem Sie auf der Tastatur die Pfeiltasten drücken. Halten Sie die <Umschalt>-Taste gedrückt, während Sie die Pfeiltasten drücken, um ausgewählte Segmente um mehrere Pixel auf einmal zu verschieben. Angrenzende, nicht ausgewählte Segmente werden dabei gedehnt, um die Änderung zu ermöglichen. Sie können mehrere, auch unverbundene, Verbindungssegmente gleichzeitig wählen und ziehen.

Wenn Sie einen Tunnel verschieben, bleibt normalerweise eine Verbindung zwischen dem Tunnel und dem verbundenen Knoten erhalten (siehe nachstehende Abbildung).



Das Verschieben eines Tunnels erzeugt manchmal ein überschüssiges Verbindungssegment, das jedoch unterhalb des Strukturrahmens liegt. Sie können dieses Segment nicht auswählen und ziehen, da es verborgen ist, aber es verschwindet, wenn Sie das damit verbundene Segment ziehen (siehe nachstehende Abbildung). Wenn Sie sich nicht sicher darüber sind,

welche Verbindung an einen Tunnel angeschlossen ist, klicken Sie dreimal auf die Verbindung.



Um die Teile einer Verbindung innerhalb und außerhalb einer Schleifenstruktur auf einmal auszuwählen, wählen Sie den Teil der Verbindung auf der einen Seite der Struktur, und halten Sie die <Umschalt>-Taste gedrückt, während Sie den Teil der Verbindung auf der anderen Seite der Struktur auswählen. Sie können einer Gruppe zuvor ausgewählter Objekte ein Objekt hinzufügen, indem Sie die <Umschalt>-Taste gedrückt halten, während Sie das neue Objekt auswählen. Sie können ferner ein Auswahlrechteck um die beiden Teile der Verbindung ziehen. Die Struktur wird nur ausgewählt, wenn Sie sie komplett mit einem Auswahlrechteck umgeben. Bei anderen Knoten reicht die Berührung mit dem Rechteck aus, um ausgewählt zu werden.

Offscreen-Verbindungen

Möglicherweise müssen Sie ein Objekt auf dem Bildschirm mit einem Objekt verbinden, das nicht auf dem Bildschirm angezeigt wird, z.B. wenn die Objekte so weit auseinander liegen, daß sie nicht auf einmal auf den Bildschirm passen. Um ein Onscreen-Objekt mit einem Offscreen-Objekt zu verbinden, klicken Sie mit dem Verbindungswerkzeug zuerst auf ein Terminal des Onscreen-Objekts. Sie können dann das Diagramm automatisch rollen, indem Sie das Verbindungswerkzeug etwas über die Kante des Blockdiagramms hinausziehen. Ist das zweite Objekt auf dem Bildschirm zu sehen, klicken Sie auf ein Terminal dieses Objekts, um die Verbindung fertigzustellen.

Beheben von Verbindungsproblemen

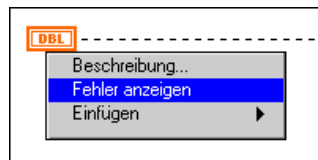
Ungültige Verbindungen

- - -



Wenn Sie bei der Platzierung von Verbindungen einen Fehler machen, wird unter Umständen eine *defekte Verbindung* erzeugt, die durch eine gestrichelte Linie angezeigt wird. Manchmal liegt eine fehlerhafte Verbindung vor, die nicht sichtbar ist, da das defekte Verbindungssegment sehr klein oder von einem Objekt verborgen ist. Wenn die Taste **Ausführen** einen gebrochenen Pfeil anzeigt (siehe Abbildung links), Sie jedoch im Blockdiagramm keine Probleme feststellen können, wählen Sie **Bearbeiten»Ungültige Verbindungen entfernen** für den Fall, daß verborgene defekte Verbindungssegmente vorhanden sind. Kehrt die Taste “Ausführen” wieder zum ungebrochenen Zustand zurück (siehe Abbildung links), ist das Problem behoben. Ist dies nicht der Fall, klicken Sie auf die Taste **Unterbrochene Ausführung**, um eine Liste von Fehlern anzuzeigen.

Wenn Sie nicht wissen, weshalb eine bestimmte Verbindung defekt ist, rufen Sie das Popup-Menü der defekten Verbindung auf, und wählen Sie daraus **Fehler anzeigen** (siehe Abbildung unten). Dadurch wird das Dialogfeld **Fehlerliste** angezeigt, welches die Fehler auflistet. Klicken Sie auf einen der Fehler. Dadurch wird die betreffende fehlerbehaftete Verbindung ausgewählt, so daß Sie sie entweder löschen oder reparieren können.



Weitere Informationen finden Sie in den Abschnitten *Debugging von nicht ausführbaren VIs* und *Debugging von ausführbaren VIs* in Kapitel 4, *Ausführung und Debugging von VIs und SubVIs*.

Die folgende Liste enthält einige der möglichen Verbindungsfehler.

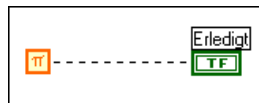
- Verbindung: Typenkonflikt
- Verbindung: Dimensionskonflikt
- Verbindung: Elementenkonflikt
- Verbindung: Refnum-Konflikt
- Verbindung: Klassenkonflikt

- Verbindung:Enumeration-Konflikt
- Verbindung:Einheitenkonflikt
- Verbindung:hat mehrere Quellen
- Verbindung:hat keine Quelle
- Verbindung:hat offene Verbindungen
- Verbindung:ist Teil eines Kreises

Informationen über Verbindungen von Strukturen finden Sie im Abschnitt [Probleme bei Verbindungen von Strukturen](#) in Kapitel 19, [Strukturen](#).

Typenkonflikte von Verbindungen

Unterschiedliche Typen treten auf, wenn Sie zwei Objekte verschiedener Datentypen miteinander verbinden, wie z.B. ein numerisches und ein Boolesches Objekt (siehe nachstehende Abbildung).



Fehler durch Dimensionskonflikt und Elementenkonflikt treten in ähnlichen Situationen auf — wenn Sie zwei Arrays miteinander verbinden, deren Elemente zwar vom selben Typ sind, nicht aber ihre Dimensionen, und wenn Sie zwei Cluster miteinander verbinden, deren Elemente jeweils unterschiedliche Typen aufweisen.

Der Refnum-Konflikt tritt auf, wenn Sie zwei verschiedene Arten von Refnum-Werten oder zwei Datenprotokolldatei-Refnums miteinander verbinden, die unterschiedliche Datensatztypen aufweisen.

Der Fehler durch Klassenkonflikt tritt auf, wenn Sie zwei gleichartige Refnums miteinander verbinden, die unterschiedliche Klassen aufweisen.

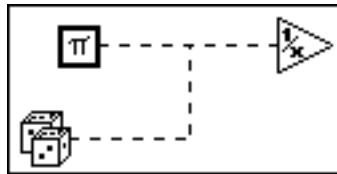
Der Enumeration-Konflikt tritt auf, wenn Sie zwei inkompatible Aufzählungen Enum miteinander verbinden. Zwei Aufzählungen sind kompatibel, wenn sie genau denselben Satz Strings aufweisen oder wenn eine einen Satz Strings aufweist, der mit den ersten Strings des gegenüberstehenden Satz Strings identisch ist.

Der Einheitenkonflikt tritt auf, wenn Sie zwei Objekte miteinander verbinden, deren Einheiten einander nicht entsprechen.

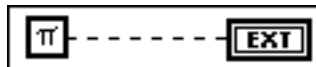
Alle anderen Arten von unterschiedlichen Typen resultieren in einem Fehler infolge Typenkonflikt. Diese Probleme treten üblicherweise auf, wenn Sie versehentlich eine Verbindung zum falschen Terminal einer Funktion oder eines SubVIs herstellen. Wählen und entfernen Sie die Verbindung, und führen Sie sie dem richtigen Terminal zu. Sie können das Hilfefenster zu Hilfe nehmen, um derartige Fehler zu vermeiden. In anderen Situationen müssen Sie eventuell den Typ eines der Terminals ändern.

Mehrere Quellen von Verbindungen

Sie können eine einzige Datenquelle mit mehreren Zielen verbinden, Sie können jedoch nicht mehrere Datenquellen mit einem Ziel verbinden. In dem nachfolgenden Beispiel müssen Sie eine Quelle trennen.



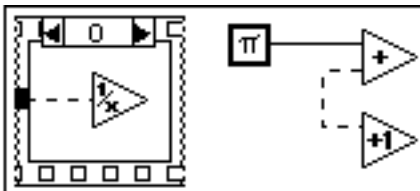
Während der Konstruktion eines Frontpanels haben Sie eventuell ein Bedienelement abgelegt, obwohl Sie eigentlich ein Anzeigeelement plazieren wollten. Wenn Sie versuchen, einen Ausgabebewertungsbereich mit dem Terminal eines Bedienelements am Frontpanel zu verbinden, erhalten Sie einen Fehler bezüglich mehrerer Quellen. Rufen Sie das Popup-Menü des Terminals auf, und wählen Sie den Befehl **In Anzeigeelement ändern**, um diesen Fehler zu korrigieren.



Keine Verbindungsquelle

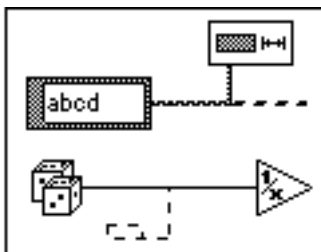
Die nachfolgende Abbildung zeigt zwei Beispiele für Verbindungen ohne Quelle. Im ersten Fall führt ein Tunnel der Funktion "Kehrwert" Daten zu, es gibt aber keine Quelle, die dem Tunnel Daten zuführt. Im anderen Fall sind zwei Funktionseingänge miteinander verbunden, aber es gibt keine Datenquelle für sie. Die Lösung dieses Problems besteht darin, eine Datenquelle mit dem Tunnel und jeweils den Funktionen "Addieren" bzw. "Inkrementieren" zu verbinden. Sie erhalten diesen Fehler auch, wenn Sie zwei Terminals für Anzeigeelemente auf dem Frontpanel miteinander verbinden, obwohl eins davon ein Bedienelement sein müßte. Rufen Sie

in diesem Fall das Popup-Menü eines der Terminals auf, und wählen Sie den Befehl **In Bedienelement ändern**.

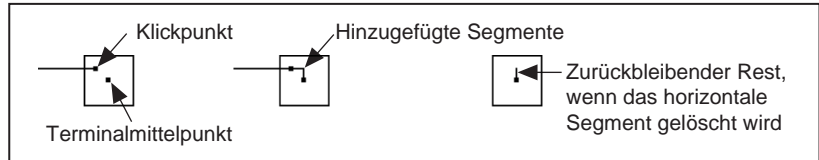


Offene Verbindungen

Offene Verbindungen, die in der nachstehenden Abbildung zu sehen sind, sind Zweige, die nicht mit einem Terminal verbunden sind. Diese können aus der Dehnung von Verbindungen, dem Retracing, während des Verbindungsprozesses oder dem Löschen von verbundenen Objekten resultieren. Durch Wählen von **Bearbeiten»Ungültige Verbindungen entfernen** werden offene Verbindungen gelöscht.



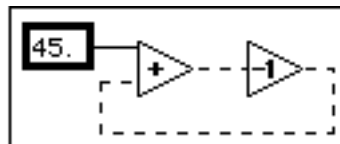
Wenn Sie eine Verbindung zu einem Terminal erstellen, klicken Sie selten auf das Terminal, wenn sich der Cursor genau auf dem Mittelpunkt befindet. Wenn Sie etwas vom Mittelpunkt abweichen, wird automatisch ein Verbindungssegment vom Klickpunkt zum Mittelpunkt des Terminals erzeugt. Wenn Sie dann die zum Terminal verlaufende Verbindung entfernen, kann eine offene Verbindung zurückbleiben, was zu einem unterbrochenem Ausführungspfeil führt. Dieses Problem wird in der nachstehenden Abbildung dargestellt.



Um diese Situation zu vermeiden, klicken Sie dreimal auf eine Verbindung, um sicherzustellen, daß alle Teile gewählt sind, bevor Sie sie löschen. Sie können zum Löschen offener Verbindungen auch **Bearbeiten»Ungültige Verbindungen löschen** wählen.

Verbindungen mit Endlosschleifen

Verbindungen dürfen keine Endlosschleifen bilden. Das heißt, daß Verbindungen keine geschlossenen Schleifen von Icons oder Strukturen bilden dürfen (siehe nachstehende Abbildung). Das Ausführungssystem kann Endlosschleifen nicht ausführen, da jeder Knoten darauf wartet, daß der andere ihm Daten zuführt, bevor er ausgeführt wird.



Der Abschnitt *Schieberegister* in Kapitel 19, *Strukturen*, beschreibt die richtige Art und Weise, Daten in einer sich wiederholenden Berechnung zurückzuführen.

Zu vermeidende Verbindungssituationen

Die nachfolgenden Abschnitte beschreiben Situationen, die zwar keine ungültigen Verbindungen erzeugen, aber das Blockdiagramm schwer lesbar machen und den Anschein erwecken, als würden Vorgänge ausgeführt, die eigentlich nicht ausgeführt werden.

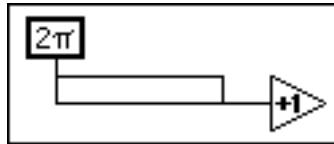


Hinweis

Wenn Sie sich nicht sicher sind, welches Objekt mit einer Verbindung verknüpft ist, können Sie zum Auswählen des Zweiges auf die Verbindung doppelklicken oder zum Auswählen der gesamten Verbindung dreimal klicken.

Schleifen von Verbindungen

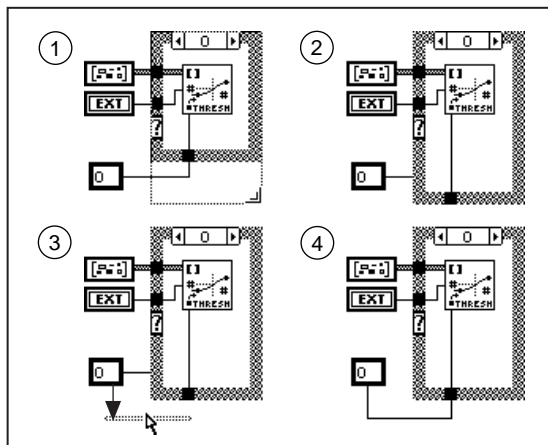
Eine Schleife einer Verbindung ist zwar kein Fehler, stellt aber ein schlechtes Design dar, da sie das Diagramm unnötigerweise überlädt. Doppelklicken Sie auf einen der Zweige, um ihn auszuwählen, und löschen Sie ihn anschließend. Die nachstehende Abbildung zeigt eine Verbindungsschleife.



Bearbeiten»Ungültige Verbindungen entfernen entfernt einen Zweig dieser Schleifen, um die Verbindungen zu bereinigen.

Verborgene Verbindungssegmente

Versuchen Sie, Verbindungen unter einem Rand einer Struktur oder zwischen überlappenden Objekten zu vermeiden, da manche Segmente der resultierenden Verbindungen dadurch verborgen werden könnten. Die nachstehende Abbildung zeigt ein Beispiel für verborgene Verbindungssegmente.

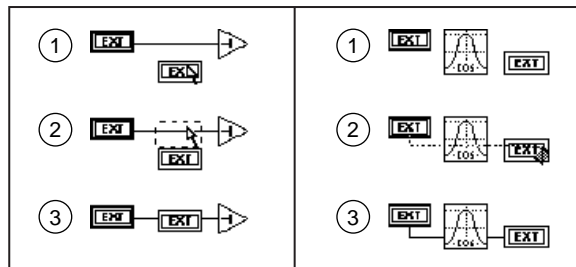


Verborgene Verbindungssegmente können versehentlich erzeugt werden, z.B. wenn Sie einen Tunnel verschieben oder eine Struktur vergrößern (siehe Nummer 1 und 2 des obigen Beispiels). Nummer 3 und 4 des Beispiels veranschaulichen eine weniger verwirrende Ausführung dieses Diagramms. Sie können das mit der Konstanten verbundene

Verbindungssegment ziehen, damit die verborgenen Verbindungssegmente wieder angezeigt werden. Drücken Sie die <Umschalt>-Taste, um das Ziehen der Verbindung einzuschränken und die Wahrscheinlichkeit, daß offene Verbindungen erzeugt werden, zu verringern.

Erzeugen von Verbindungen unterhalb von Objekten

Verbindungen werden nur an den Objekten festgemacht, auf die Sie klicken. Durch Ziehen eines Terminals oder Icons auf eine Verbindung wird der Anschein erweckt, daß hier eine Verbindung zu diesen Objekten besteht, was nicht der Fall ist, wie Sie auf der linken Seite der nachstehenden Abbildung sehen können (1, 2, 3 links).

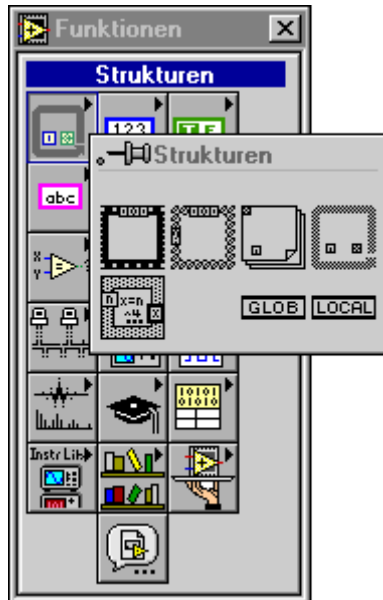


Wird eine Verbindung durch ein Icon oder Terminal gezogen, wird ebenfalls der Anschein erweckt, als bestehe eine Verbindung mit diesen Objekten, während die Verbindung tatsächlich hinter dem Icon verläuft, wie Sie in der Abbildung sehen (1, 2, 3 rechts). Vermeiden Sie derartige Situationen, da sie visuell verwirrend sind.

Weitere Informationen zu diesem Problem finden Sie im Abschnitt [Warnungen](#) in Kapitel 4, *Ausführung und Debugging von VIs und SubVIs*.

Strukturen

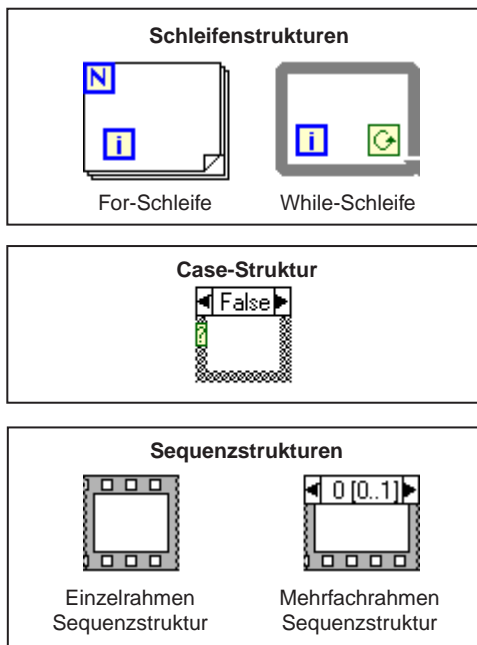
Dieses Kapitel beschreibt die Verwendung der For-Schleife, While-Schleife, Case-Struktur und Sequenzstruktur. Diese Strukturen befinden sich in der Palette **Funktionen»Strukturen** (siehe nachstehende Abbildung).



Beispiele für die Anwendung dieser Strukturen finden Sie in `examples\general\structs.llb`.

Strukturen sind Knoten, die den Ausführungsfluß in einem Blockdiagramm ergänzen, wie dies in einer herkömmlichen Programmiersprache durch Steuerungsstrukturen geschieht. Das Icon für jede G-Struktur ist ein in der

Größe veränderbares Feld mit einem kennzeichnenden Rahmen (siehe nachstehende Abbildung).



Strukturen verhalten sich insofern wie andere Knoten, als sie automatisch ausgeführt werden, sobald ihre Eingangsdaten verfügbar sind, und sie führen ihren Ausgangsverbindungen erst bei Abschluß der Ausführung Daten zu. Jede Struktur führt jedoch ihr Subdiagramm entsprechend den in den folgenden Abschnitten beschriebenen Regeln aus.

Ein Subdiagramm ist die Sammlung von Knoten, Verbindungen und Terminals, die innerhalb der Strukturgrenze residieren. Die For-Schleife und While-Schleife besitzen je ein Subdiagramm. Die Case-Struktur und Sequenzstruktur können jedoch mehrere Subdiagramme umfassen, die wie Karten in einer Weise gestapelt sind, so daß immer nur eine(s) sichtbar ist. Subdiagramme werden wie das Blockdiagramm der höchsten Ebene konstruiert; Subdiagramme können in Blockdiagrammen übliche Terminals, Knoten (einschließlich anderer Strukturen) und Verbindungen enthalten.

Sie erstellen Terminals zur automatischen Übergabe von Daten an eine und von einer Struktur, wobei Verbindungen die Strukturgrenzen überschreiten. Diese Grenzterminals werden als *Tunnel* bezeichnet. Bei Tunneln ist stets eine Kante im Innern der Struktur exponiert und eine Kante auf der

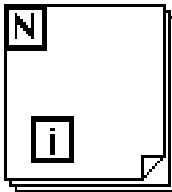
Außenseite. Ein Tunnel residiert stets am Rand der Struktur, sie können ihn aber am Rand durch Ziehen mit dem Positionierwerkzeug beliebig verschieben.

Strukturen besitzen auch andere Terminals, die für jeden Strukturtyp spezifisch sind. Diese Terminals werden mit den passenden Strukturen in den folgenden Abschnitten beschrieben.

Strukturen vom Typ For-Schleife und While-Schleife

Mit der For-Schleife und der While-Schleife werden sich wiederholende Operationen gesteuert, und zwar entweder solange, bis eine bestimmte Anzahl von Iterationen abgearbeitet worden ist (For-Schleife) oder solange eine vorgegebene Bedingung zutrifft (TRUE) (While-Schleife).

For-Schleife



Iterationsterminal



Zählerterminal

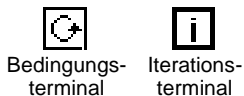
Eine For-Schleife wiederholt ihr Subdiagramm solange, bis der *Zähler* abgearbeitet ist, wobei der Zähler gleich dem im *Zählerterminal* enthaltenen Wert ist. Sie können den Zähler explizit einstellen, indem Sie einen Wert von außerhalb der Schleife mit der linken oder oberen Seite des Zählerterminals verbinden, oder Sie können den Zähler per *Auto-Indizierung* implizit einstellen (weitere Informationen hierzu siehe Abschnitt *Auto-Indizierung* in diesem Kapitel). Die anderen Kanten des Zählerterminals sind im Innern der Schleife exponiert, damit Sie intern auf den Zähler zugreifen können.

Das *Iterationsterminal* beinhaltet die aktuelle Anzahl der abgeschlossenen Iterationen; während der ersten Iteration 0, während der zweiten 1 usw. bis zu $N - 1$. Sowohl das *Zähler-* als auch das *Iterationsterminal* sind vorzeichenbehaftete Long-Integer-Werte im Bereich von 0 bis $2^3 - 1$. Wenn Sie eine Fließkommazahl mit dem Zählerterminal verbinden, rundet G diese gegebenenfalls ab und wandelt das Format dem gültigen Bereich entsprechend um. Wenn Sie 0 mit dem Zählerterminal verbinden, wird die Schleife nicht ausgeführt.

Die For-Schleife entspricht dem folgenden Pseudocode.

```
for i = 0 to N - 1
    Execute subdiagram
```

While-Schleife



Eine While-Schleife führt das Subdiagramm solange aus, bis ein mit dem *Bedingungsterminal* verbundener Boolescher Wert FALSE ist. G überprüft den Wert des Bedingungsterminals am Ende jeder Iteration, und wenn der Wert TRUE ist, erfolgt eine weitere Iteration, so daß die Schleife stets mindestens einmal ausgeführt wird. Der Standardwert des Bedingungsterminals ist FALSE, so daß im unverbundenen Zustand nur eine Iteration der Schleife erfolgt.

Das Iterationsterminal beinhaltet ebenso wie bei der For-Schleife die Anzahl der ausgeführten Iterationen.

Die While-Schleife entspricht dem folgenden Pseudocode.

```
Do
    Execute subdiagram (which sets condition)
While condition is TRUE
```

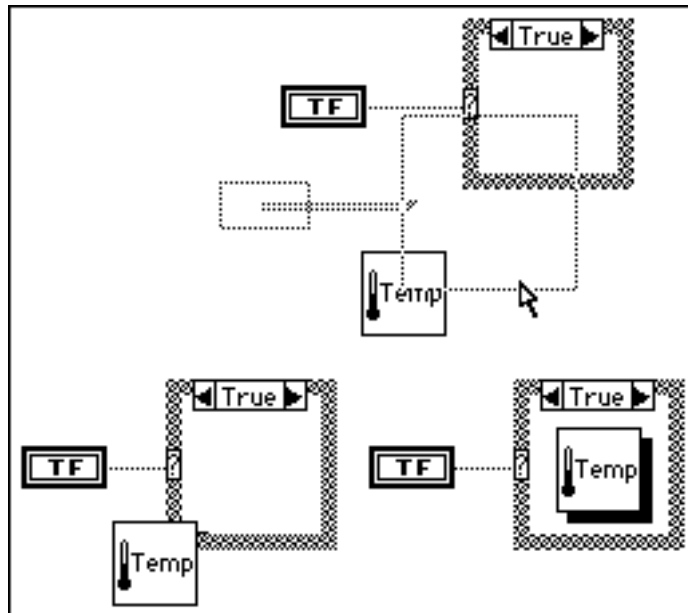
Sie können beiden Schleifenstrukturen Terminals, die als *Schieberegister* bezeichnet werden, hinzufügen. Mit Hilfe von Schieberegistern werden Daten von der aktuellen Iteration an die nächste Iteration übergeben. Weitere Informationen hierzu finden Sie im Abschnitt [Schieberegister](#) in diesem Kapitel.

Plazieren von Objekten in Strukturen

Sie können ein Objekt durch Ziehen in eine Struktur plazieren, oder indem Sie die Struktur um das Objekt herum konstruieren.

Sie können ein Objekt nicht in eine Struktur einfügen, indem Sie die Struktur über das Objekt ziehen. Wenn Sie eine Struktur verschieben, die ein anderes Objekt überlappt, zeigt der Editor das Objekt an, das er als das vordere betrachtet. Wenn Sie die Struktur komplett auf ein anderes Objekt ablegen, und der Editor macht dieses zum vorderen Objekt, wird ein dicker Schatten angezeigt, um Ihnen eine Warnung zu geben, daß sich das Objekt unterhalb der Struktur und nicht innerhalb der Struktur befindet. Ansonsten wird die Struktur vollständig vom Objekt verdeckt. Weitere Informationen darüber, welche Objekte von der Anwendung im Vordergrund dargestellt werden, finden Sie im Abschnitt [Positionieren von Objekten](#) in Kapitel 2,

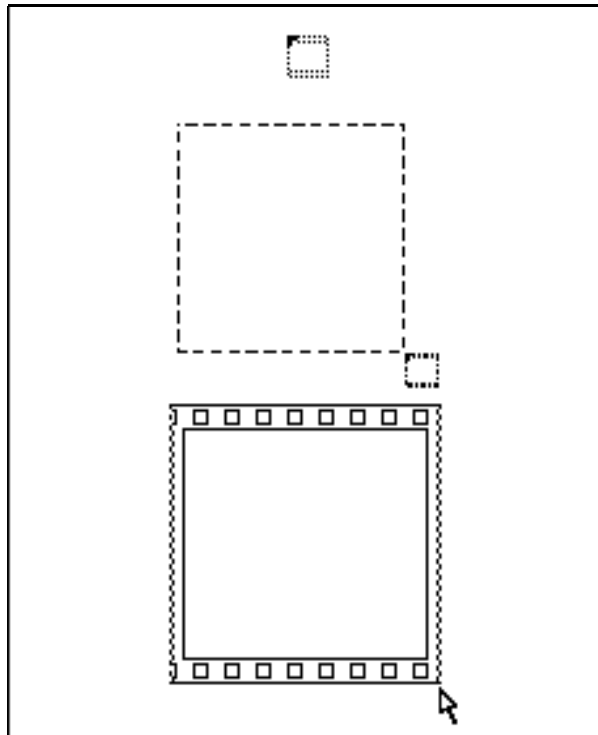
Bearbeiten von VIs. Beide Situationen sind in der nachfolgenden Abbildung dargestellt.



Plazierung und Größenanpassung von Strukturen im Blockdiagramm

Wenn Sie eine Struktur aus der Palette **Funktionen»Strukturen** wählen und daran gehen, sie in das Diagramm einzufügen, sehen Sie anstelle des Cursors ein kleines Icon der Struktur. Das Icon ist zur Plazierung im Blockdiagramm bereit. Wenn Sie die Maustaste klicken und ohne Größenveränderung freigeben, befinden Sie sich noch solange im Größenanpassungsmodus, bis Sie aufs neue klicken. Während Sie die Maustaste gedrückt halten, können Sie auf das Diagramm klicken und die Struktur, deren Größe Sie anpassen wollen, ziehen. Die folgende Abbildung zeigt ein Beispiel hierfür. Sie können die Größe der Struktur

auch anpassen, *nachdem* Sie sie im Diagramm ablegen, indem Sie mit dem Größenanpassungscursor auf eine beliebige Ecke klicken und ziehen.

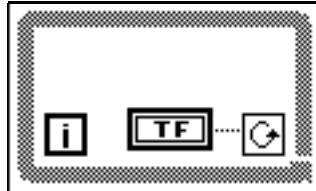


Plazieren von Terminals in Schleifen

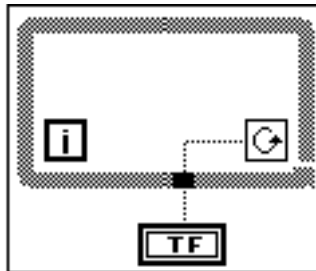
Eingänge übergeben Daten zu Beginn der Schleifenausführung an eine Schleife. Ausgänge übergeben Daten erst dann aus einer Schleife, nachdem die Schleife alle Iterationen abgeschlossen hat.

Sie müssen ein Terminal *in* eine Schleife plazieren, wenn die Schleife den Wert des Terminals bei jeder Iteration überprüfen soll. Wenn Sie z.B. das Terminal eines Booleschen Bedienelements am Frontpanel in eine While-Schleife einfügen und das Terminal mit dem Bedingungsterminal der Schleife verbinden, überprüft die Schleife den Wert des Terminals bei *jeder* Iteration, um festzustellen, ob eine weitere Iteration erforderlich ist.

Sie können diese While-Schleife anhalten (siehe nachstehende Abbildung), indem Sie den Wert des Bedienelements am Frontpanel in FALSE abändern.



Wenn Sie das Terminal des Booleschen Bedienelements außerhalb der While-Schleife platzieren (siehe Abbildung unten), erzeugen Sie eine Endlosschleife, wenn das Boolesche Bedienelement beim Starten der Schleife TRUE ist.



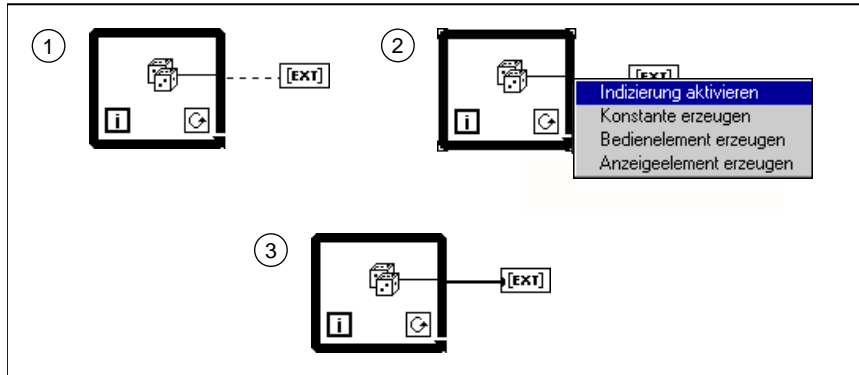
Wenn das Bedienelement beim Start TRUE war, wird die Ausführung dieser Schleife durch Ändern des Bedienelementwerts in FALSE nicht angehalten, da der Wert erst weitergeleitet wird, wenn die Schleife angehalten und das VI erneut ausgeführt wird. Wenn Sie versehentlich eine Endlosschleife erzeugen, können Sie sie durch Abbruch des VIs anhalten. Klicken Sie zum Anhalten der Ausführung auf die Abbruchtaste.

Auto-Indizierung

Die Strukturen der For-Schleife und While-Schleife können an ihren Rändern Arrays auto-indizieren und sammeln. Diese Fähigkeiten werden global als *Auto-Indizierung* bezeichnet. Wenn Sie ein Array von einer externen Quelle mit einem Eingangstunnel am Rand der Schleife verbinden und die Auto-Indizierung am Eingangstunnel aktivieren, treten Komponenten dieses Arrays, angefangen bei der ersten Komponente, nacheinander in die Schleife ein. Die Schleife indiziert Skalare aus eindimensionalen Arrays, eindimensionale Arrays aus zweidimensionalen Arrays usw. Die umgekehrte Aktion erfolgt an Ausgangstunneln – Skalare

sammeln sich sequentiell in eindimensionalen Arrays, eindimensionale Arrays sammeln sich in zweidimensionalen Arrays usw.

Die folgende Abbildung zeigt das Aussehen von Tunneln auf den Strukturgrenzen von Schleifen mit und ohne Auto-Indizierung. Die Verbindung wird mit dem Dimensionswechsel am Rand der Schleife dicker.



For-Schleifen werden oft zur sequentiellen Verarbeitung von For-Schleifen verwendet. Aus diesem Grund ist bei Arraytunneln die Auto-Indizierung standardmäßig aktiviert, wenn Sie eine Verbindung zwischen einem Array und einer For-Schleife (Ein- oder Ausgang) erstellen. While-Schleifen werden im allgemeinen nicht zu diesem Zweck verwendet, weshalb bei ihnen die Auto-Indizierung nicht standardmäßig aktiviert ist. Um die Auto-Indizierung eines Tunneln am Rand der Schleife zu aktivieren oder deaktivieren, müssen Sie das Popup-Menü des Tunneln am Rand der Schleife aufrufen und **Indizierung aktivieren** oder **Indizierung deaktivieren** wählen.

Auto-Indizierung zur Einstellung des For-Schleifenzählers

Wenn Sie die Auto-Indizierung bei einem Array mit Eingang in eine For-Schleife beginnen, wird hierbei der Zähler automatisch auf die Größe des Arrays eingestellt, wodurch die Notwendigkeit beseitigt wird, explizit eine Verbindung zum Zählerterminal zu erstellen. Wenn Sie die Auto-Indizierung für mehrere Tunnel starten oder wenn Sie auch den Zähler explizit einstellen, nimmt der Zähler die geringste Auswahl an. Wenn beispielsweise zwei auto-indizierte Arrays in die Schleife mit 10 bzw. 20 Komponenten eingegeben werden und Sie einen Wert von 15 mit dem Zählerterminal verbinden, wird die Schleife zehnmal

ausgeführt, und die Schleife indiziert nur die ersten 10 Komponenten des zweiten Arrays.

Ausgangsarrays der Auto-Indizierung empfangen bei jedem Schleifendurchlauf ein neues Ausgabeelement. Die Größe der auto-indizierten Ausgangsarrays entspricht stets der Anzahl der Iterationen (im obigen Beispiel 10).

Ist die Auto-Indizierung an einem Ausgangstunnel deaktiviert, wird nur der Elementwert der letzten Iteration der Schleife ausgegeben.

Auto-Indizierung bei While-Schleifen

Wenn Sie die Auto-Indizierung für ein Array, das in eine While-Schleife eingegeben wird, aktivieren, indiziert die While-Schleife das Array ebenso wie eine For-Schleife. Die Anzahl der von einer While-Schleife ausgeführten Iterationen ist jedoch nicht durch die Größe des Arrays beschränkt, da die While-Schleife so oft durchlaufen wird, wie ein bestimmter Zustand zutrifft (=TRUE). Wenn eine While-Schleife bei der Indizierung das Ende des Eingangsarrays erreicht, wird der Standardwert für den Elementtyp des Arrays in die Schleife eingegeben. Auto-indizierte Ausgangsarrays empfangen bei jedem Schleifendurchlauf ein Ausgabeelement. Ihre Größe wächst solange an, wie die While-Schleife ausgeführt wird.



Hinweis

Bei der Auto-Indizierung bei einer While-Schleife, die zu oft durchlaufen wird, können die Speicherressourcen erschöpft werden. LabVIEW-Benutzer können weitere Informationen über das Erstellen von Arrays mit While-Schleifen in Kapitel 5, Arrays, Cluster und Graphen, des LabVIEW Benutzerhandbuchs entnehmen, um dieses Problem zu vermeiden.

Nullmalige Ausführung einer For-Schleife

Wenn Sie den Zähler auf Null setzen, wird das Subdiagramm einer For-Schleife überhaupt nicht ausgeführt. Der Wert aller Skalarvariablen, die die For-Schleife verlassen, richtet sich nach den folgenden Regeln.

- Ein mit Auto-Indizierung an einem Ausgangstunnel erstelltes Ausgangsarray ist leer.
- Der Ausgang eines initialisierten Schieberegisters ist der Anfangswert. Weitere Informationen hierzu finden Sie im Abschnitt [Schieberegister](#) in diesem Kapitel.

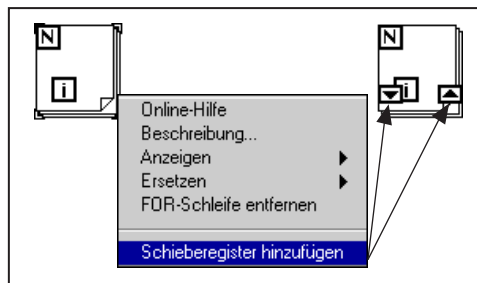
- Ein Array, das einen nicht indizierenden Ausgangstunnel durchlaufen hat, ist ebenfalls leer.
- Alle Skalare, die nicht indizierende Ausgangstunnel durchlaufen haben, sind nicht definiert, und ihre Werte sind nicht zuverlässig.

Es gibt zwei Möglichkeiten, wie der Schleifenzähler standardmäßig oder nicht standardmäßig auf Null gesetzt wird. Sie können ein leeres Eingangsarray auto-indizieren, oder Sie können eine Null oder eine negative Zahl explizit mit dem Zählerterminal verbinden.

Wenn Sie Eingangsarrays auto-indizieren oder den Schleifenzähler mit einer Variablen einstellen, analysieren Sie das Diagramm, um festzustellen, ob eine nullmalige Zählung stattfindet, und stellen Sie sicher, daß das Diagramm ein leeres Array handhaben kann.

Schieberegister

Schieberegister, die für For-Schleifen und While-Schleifen zur Verfügung stehen, sind lokale Variablen, die Werte vom Abschluß einer Iteration zum Anfang der nächsten zurückführen oder übertragen. Durch Wählen von **Schieberegister hinzufügen** aus dem Popup-Menü des Schleifenrandes erstellen Sie an der vertikalen Grenze an einer beliebigen Stelle ein Register (siehe nachstehende Abbildung). Dieser Menüpunkt ist an der oberen oder unteren Kante der Struktur nicht verfügbar. Sie können ein Schieberegister durch Ziehen entlang der vertikalen Grenze neu positionieren.

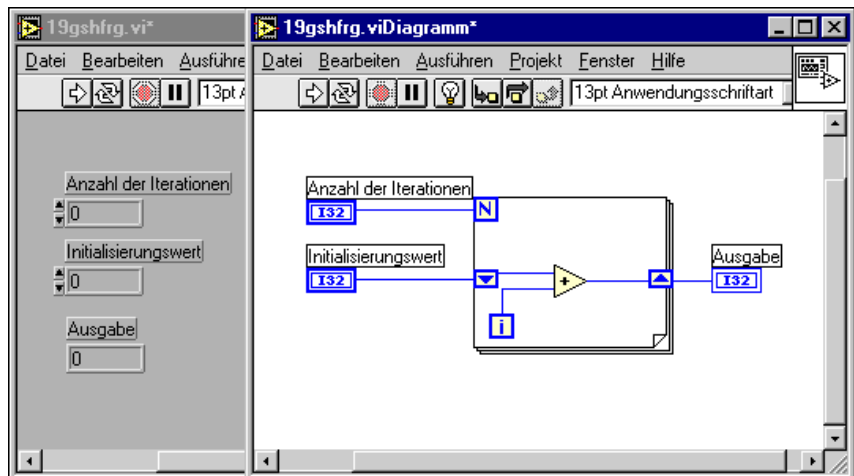


Ein Schieberegister besitzt auf den vertikalen Seiten der Schleifengrenze ein sich gegenüberliegendes Terminalpaar. Das *rechte Terminal*, das Rechteck mit dem Pfeil nach oben, speichert die Daten bei Abschluß einer Iteration. G verschiebt die Daten am Ende der Iteration, und sie erscheinen rechtzeitig zur nächsten Iteration im *linken Terminal*, dem Rechteck mit dem nach unten weisenden Pfeil. Sie können Schieberegister für einen beliebigen Datentyp verwenden, jedoch müssen die Daten, die mit den

Terminals jedes Registers verbunden werden, denselben Datentyp aufweisen. Sie können mehrere Schieberegister in einer bestimmten Struktur erstellen.

Um ein Schieberegister zu initialisieren, verbinden Sie einen außerhalb der Schleife befindlichen Wert mit dem linken Terminal. Wenn Sie das Register nicht initialisieren, benutzt die Schleife den Wert, der bei der letzten Ausführung der Schleife in das Register geschrieben worden ist, oder falls die Schleife zuvor noch nicht ausgeführt worden ist, den Standardwert für diesen Datentyp. Benutzen Sie initialisierte Schieberegister, um eine konsistente Verhaltensweise zu gewährleisten. LabVIEW-Benutzer können weitere Informationen zur Verwendung nicht initialisierter Schieberegister in Kapitel 3, *Schleifen und Diagramme*, im *LabVIEW Benutzerhandbuch* finden. BridgeVIEW-Benutzer können Kapitel 10, *Loops and Charts*, im *BridgeVIEW User Manual* konsultieren.

Wenn die Ausführung der Schleife abgeschlossen ist, verbleibt der zuletzt im Schieberegister gespeicherte Wert im rechten Terminal. Wenn das rechte Terminal mit einem Objekt außerhalb der Schleife verbunden wird, wird dieser letzte Wert bei Abschluß der Schleife ausgegeben. Diese Eigenschaft ist in der nachstehenden Abbildung zu sehen.

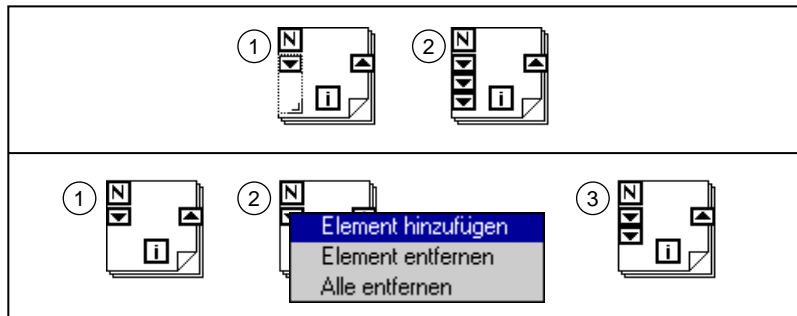


Die linke Seite eines Schieberegisters kann mehrere Terminals beinhalten, damit es mehrere vorherige Werte speichern kann.

Um einem bestimmten Schieberegister Terminals hinzuzufügen oder daraus zu entfernen, verwenden Sie den Größenanpassungscursor, um die Größe der linken Terminals anzupassen. Als Alternative können Sie auch



den Befehl **Element hinzufügen** aus dem Popup-Menü des Schieberegisters dazu verwenden, dem Schieberegister zusätzliche linke Terminals hinzuzufügen. Hinzugefügte Terminals erscheinen direkt unter demjenigen, dessen Popup-Menü Sie aufrufen. Verwenden Sie den Befehl **Element entfernen**, um das Terminal, dessen Popup-Menü Sie aufrufen, zu entfernen. Die nachfolgende Abbildung zeigt beide Methoden. Die einzige Möglichkeit, überflüssige verbundene Terminals zu entfernen, besteht darin, den Befehl **Element entfernen** aus dem Popup-Menü des Schieberegisters zu wählen. Durch Auswahl von **Alle entfernen** werden die Schieberegister gelöscht.



Das oberste linke Terminal enthält den Wert aus der vorherigen Iteration, $i - 1$. Das Terminal unmittelbar darunter enthält den Wert aus der Iteration $i - 2$ usw. mit jedem nachfolgenden Terminal.

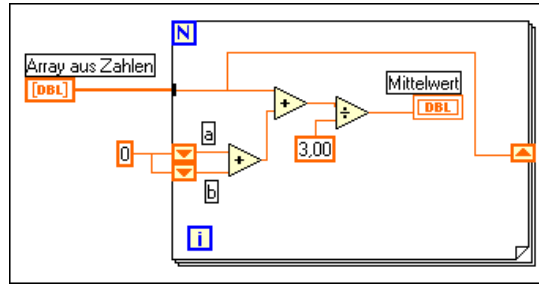
Wenn Sie ein linkes Terminal eines Schieberegisters initialisieren, müssen Sie alle initialisieren.

Der folgende Pseudocode zeigt eine dem G-Blockdiagramm entsprechende Routine, die den Durchschnitt von drei Werten bestimmt.

```

a = b = 0
for i = 0 to N - 1
    avg = (x[i] + a + b)/3
    b = a
    a = x[i]

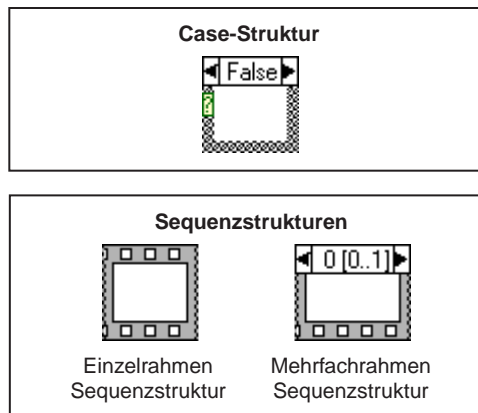
```

Ein Beispiel für das Vorgehen bei der Einstellung eines Schieberegisters finden Sie in `examples\general\structs.llb\Random Average.vi`.

Case-Strukturen und Sequenzstrukturen

Sowohl Case-Strukturen als auch Sequenzstrukturen können mehrere Subdiagramme besitzen, die wie ein Kartenspiel konfiguriert sind, bei dem immer nur eine Karte sichtbar ist. An oberster Stelle jedes Randes einer Struktur befindet sich das *Anzeigefenster des Subdiagramms*, das in der Mitte einen *Diagramm-Identifikator* enthält und auf beiden Seiten eine Dekrement- und eine Inkrementtaste. Der Diagramm-Identifikator einer Case-Struktur zeigt die Werte an, die die Ausführung des gegenwärtig angezeigten Subdiagramms auslösen. Bei einer Sequenzstruktur gibt der Diagramm-Identifikator an, welches Subdiagramm gerade angezeigt wird. Ist der Diagramm-Identifikator ein numerischer Wert, folgt ihm eine Bereichsangabe für den Diagramm-Identifikator, die den Mindest- und Höchstwert anzeigt, der im Subdiagramm der Struktur gültig ist.



Durch Klicken auf die Dekrement- (links) oder Inkrement-Taste (rechts) wird jeweils das vorherige oder nächste Subdiagramm angezeigt. Durch Dekrementieren vom ersten Subdiagramm aus wird das erste und durch Inkrementieren vom letzten Subdiagramm aus wird das letzte Subdiagramm angezeigt. Andere Verwendungen des Anzeigefensters werden in den Abschnitten *Case-Strukturen und Sequenzstrukturen* in diesem Kapitel erläutert.

Ein Beispiel einer Case-Struktur finden Sie in `examples\general\structs.llb\SquareRoot.vi`.

Case-Strukturen

Case-Strukturen wählen ein einziges Subdiagramm, oder einen Case-Wert, zur Ausführung auf der Basis eines Eingangswerts, der als Auswahl bezeichnet wird. Bei Case-Strukturen können Sie folgende Operationen ausführen:

- Listen und Bereiche von Auswahlwerten angeben, die einem Case entsprechen.
- Ein Objekt vom Typ Integer-Wert, Boolesch, String oder Aufzählung als Auswahl verwenden.
- Einen Standard-Case (Aktion) angeben.
- Cases anhand des ersten Auswahlwerts sortieren.

Wenn Sie eine Case-Struktur in ein Blockdiagramm plazieren, geben Sie die Auswahlwerte direkt in das Auswahl-Label der Case-Struktur ein. Sie können die Auswahlwerte auch mit dem Beschriftungswerkzeug bearbeiten. Sie können einen einzelnen Wert oder Listen und Bereiche von Werten eingeben, die den Case auswählen. Um eine Liste anzugeben, trennen Sie die Werte durch Kommata, wie z.B. `-1, 0, 5, 10`. Ein Bereich wird mit `10..20` angegeben, womit alle Zahlen von einschließlich 10 bis 20 gemeint sind. Sie können auch unendliche Bereiche wie z.B. `..0` (alle Zahlen, die kleiner oder gleich 0 sind) oder `100..` (alle Zahlen, die größer oder gleich 100 sind) spezifizieren. Listen und Bereiche können miteinander kombiniert werden, wie z.B. bei `..5, 6..10, 12, 13, 14`. Wenn Sie eine Auswahl eingeben, die überlappende Bereiche enthält, zeigt die Case-Struktur die Auswahl in kompakterer Form neu an. Das vorherige Beispiel wird hierbei als `..10, 12..14` angezeigt.

Sie können in einer Case-Auswahl auch String- und Enum-Werte verwenden. Diese stehen immer in Anführungszeichen, wie z.B. `"rot"`, `"grün"`, `"blau"`. Sie brauchen bei der Eingabe der Werte die

Anführungszeichen jedoch nur dann einzugeben, wenn der String oder die Aufzählung ein Komma oder das Symbol “..” enthält.



Hinweis

Wenn Sie einen Auswahlwert eingeben, der nicht dem Typ des mit dem Auswahlicon verbundenen Objekts entspricht, wird der Wert rot angezeigt, und das VI kann nicht ausgeführt werden. Wegen des möglichen Rundungsfehlers, der der Fließkomma-Arithmetik eigen ist, erlaubt es LabVIEW außerdem nicht, Fließkommazahlen in Labeln einer Case-Auswahl zu verwenden. Wenn Sie einen Fließkommamatyp mit dem Case verbinden, wird der Typ wie in vorherigen Versionen von LabVIEW auf den nächsten Integer-Wert gerundet. Wenn Sie versuchen, einen Fließkommawert wie z.B. 1,2 in die Case-Auswahl einzugeben, wird diese rot angezeigt.

Der unten dargestellte Case hat einen numerischen Datentyp, weshalb die String-Auswahl rot angezeigt wird.

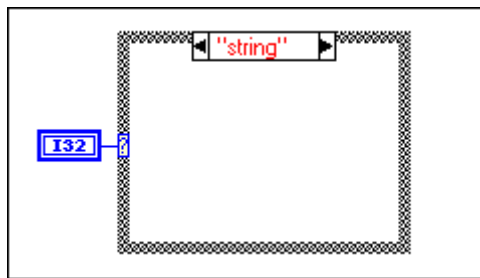


Abbildung 19-1. Unterschiedliche Typen in einer Case-Struktur

Um weitere Cases hinzuzufügen, befolgen Sie das im Abschnitt [Hinzufügen von Subdiagrammen](#) in diesem Kapitel erläuterte Verfahren.



Hinweis

Case-Anweisungen in anderen Programmiersprachen führen im allgemeinen keine Cases aus, falls ein Case-Wert außerhalb des Bereichs liegt. In G müssen Sie einen Standard-Case festlegen, der Werte außerhalb des Bereichs handhabt, oder jeden möglichen Eingangswert explizit auflisten.

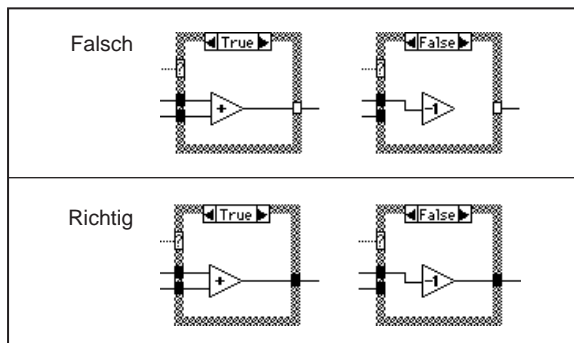
Sie können die Auswahl an einer beliebigen Stelle am linken Rand entlang positionieren, die Auswahl muß jedoch verbunden werden. Die Auswahl paßt sich automatisch an den Datentyp an. Wenn Sie den Wert der Auswahl von einem numerischen in einen Booleschen Datentyp ändern, werden die Cases 0 und 1 in FALSE und TRUE geändert. Sind andere Cases vorhanden (2 bis n), werden diese von G nicht verworfen, falls die Änderung der Datentypen zufällig erfolgt ist. Sie müssen diese überflüssigen Cases jedoch löschen, bevor die Struktur ausgeführt werden kann.

Dasselbe Prinzip gilt, wenn Sie eine Aufzählung mit der Auswahl verbinden und es mehr Cases als Aufzählungselemente gibt. Der Diagramm-Identifikator für solche Cases wird als roter numerischer Identifikator dargestellt, um darauf hinzuweisen, daß diese Cases gelöscht werden müssen, bevor die Struktur ausgeführt werden kann.

Wenn Sie Case-Auswahlwerte in einen anderen Typ konvertieren, ziehen Sie folgendes in Betracht. Wenn Sie numerische Werte wie z.B. 23 in einen String konvertieren, beträgt der Stringwert 23. Wenn Sie einen String in einen numerischen Wert konvertieren, werden nur diejenigen Auswahlstrings konvertiert, die eine Zahl darstellen. Die anderen Werte bleiben Strings. Wenn Sie eine Zahl in ein Boolesches Element konvertieren, dann wird 0 in `False` und 1 in `True` konvertiert, während die anderen Werte zu Strings werden.

Die Daten an allen Eingangsterminals (Tunnel und Auswahlterminal) stehen allen Cases zur Verfügung. Cases brauchen weder Eingangsdaten zu verwenden, noch Ausgangsdaten zu liefern, *wenn aber ein Case Ausgangsdaten liefert, so müssen dies alle tun*. Wenn Sie nicht von jedem Case aus Daten mit einem Ausgangstunnel verbinden, wird der Tunnel weiß (siehe oberes Beispiel der nachstehenden Abbildung), und die Taste **Ausführen** zeigt einen unterbrochenen Pfeil an.

Wenn alle Cases Daten an den Tunnel liefern, wird dieser schwarz (wie im unteren Beispiel der nachstehenden Abbildung), und der Pfeil der Ausführungstaste erscheint nicht unterbrochen.



Die Menüpunkte des Popup-Menüs der Case-Struktur beinhalten Elemente für nicht-Boolesche Cases (siehe nachfolgende Abbildung).

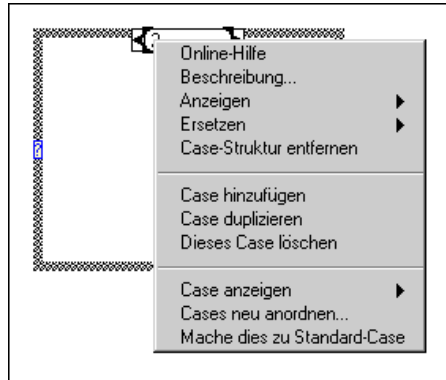


Abbildung 19-2. Menüpunkte des Popup-Menüs für eine Case-Struktur

Die Funktion **Case hinzufügen** fügt nach dem sichtbaren Case einen neuen Case ein. Vor dem sichtbaren Case können keine Cases eingefügt werden.

Sie können die Case-Reihenfolge ändern, indem Sie **Cases neu anordnen...** wählen. Hierbei wird das folgende Dialogfeld angezeigt.



Abbildung 19-3. Dialogfeld "Cases neu anordnen"

Mit der Taste **Sortieren** werden die Case-Auswahlwerte anhand des ersten Auswahlwertes sortiert. Um die Position einer Auswahl zu ändern, klicken Sie auf den zu verschiebenden Auswahlwert, und ziehen Sie ihn an die gewünschte Position. **Selektor-String vervollständigen** zeigt die Case-Auswahl für den Fall, daß sie längenmäßig nicht in das Feld **Case-Liste** paßt, in ihrer Gesamtheit an. Durch Klicken auf die Taste **Hilfe** steht zu jedem aufgeführten Bedienelement kontextsensitive Hilfe zur Verfügung.

**Hinweis**

Diese Funktion ändert nur die Reihenfolge, in der die Cases in der Case-Struktur erscheinen. Sie wirkt sich nicht auf das Run-Time-Verhalten der Case-Struktur aus.

Der Menüpunkt **Mache dies zu Standard-Case** aus dem Popup-Menü gibt einen bestimmten Case an, der auszuführen ist, wenn der Auswahlwert in der Case-Struktur nicht in der Liste geführt wird. Das Wort **Standard** ist der oberste Eintrag in der Liste der Auswahlwerte der Case-Struktur. Wenn eine Case-Struktur nicht für jeden möglichen Auswahlwert eine Auswahl enthält, müssen Sie einen Standard-Case angeben.

Das Hinzufügen, Verschieben und Löschen von Case-Subdiagrammen wird im Anschluß an den nachfolgenden Abschnitt, *Sequenzstrukturen*, diskutiert, da diese Operationen bei beiden Strukturen ähnlich ablaufen.

Sequenzstrukturen

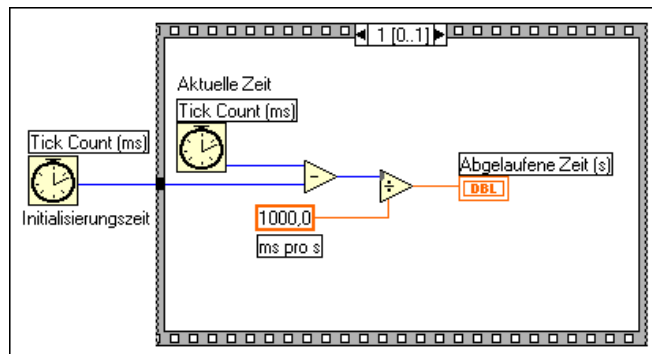
Die Sequenzstruktur, die wie ein Einzelbild eines Films aussieht, besteht aus einem oder mehreren Subdiagrammen oder *Rahmen*, die nacheinander ausgeführt werden. Ein Beispiel eines VIs, das eine Sequenzstruktur verwendet, finden Sie in `examples\general\structs.llb\TimingTemplate.vi`.

Das Bestimmen der Ausführungsreihenfolge eines Programms durch die Anordnung seiner Elemente in Reihe wird als *Steuerungsfluß* bezeichnet. BASIC, C und die meisten anderen Programmiersprachen besitzen einen inhärenten Steuerungsfluß, da Anweisungen in der Reihenfolge ausgeführt werden, in der sie im Programm erscheinen. Die Sequenzstruktur ist eine Methode, einen Steuerungsfluß zu realisieren, wenn die Datenabhängigkeiten nicht ausreichen. Eine Sequenzstruktur führt zuerst Rahmen 0 aus, gefolgt von Rahmen 1 und Rahmen 2, bis der letzte Rahmen ausgeführt ist. Erst wenn der letzte Rahmen vollständig ausgeführt ist, verlassen die Daten die Struktur.

Wie im übrigen Blockdiagramm bestimmen Datenabhängigkeiten in jedem Rahmen die Ausführungsreihenfolge von Knoten.

Die Sequenzstruktur wird zur Steuerung der Ausführungsreihenfolge von Knoten verwendet, zwischen denen keine Datenabhängigkeiten bestehen. Ein Knoten, der seine Daten direkt oder indirekt von einem anderen Knoten empfängt, besitzt eine Datenabhängigkeit von dem anderen Knoten und wird stets nach Abschluß dieses Knotens ausgeführt. Wenn Datenabhängigkeit besteht oder wenn die Ausführungsreihenfolge nicht maßgeblich ist, ist es nicht notwendig, die Sequenzstruktur zu verwenden.

Diese Situation tritt oft ein, wenn Sie feststellen wollen, wie lange die Ausführung einer Funktion dauert. Die Funktion "Tick Count" gibt die aktuelle Zeit in Millisekunden zurück. Die Zeit ist für die Durchführung der Funktion nicht erforderlich, es besteht also keine Datenabhängigkeit zwischen der Zeitmessung und der Funktion. Wie in der nachstehenden Abbildung zu sehen ist, wird die Funktion "Tick Count" links von der Sequenzstruktur vor der Sequenzstruktur ausgeführt. Dann wird die Funktion in Rahmen 0 ausgeführt, und die Berechnung der abgelaufenen Zeit erfolgt in Rahmen 1. Die Sequenzstruktur setzt die geeignete Ausführungsreihenfolge durch. (Rahmen 0 ist nicht abgebildet).

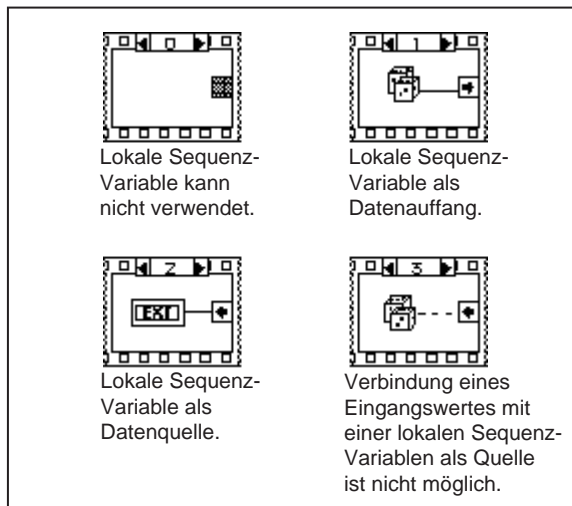


Ausgangstunnel von Sequenzstrukturen können im Gegensatz zu Case-Strukturen lediglich *eine* Datenquelle haben. Der Ausgang kann von einem beliebigen Rahmen ausgehen; beachten Sie jedoch, daß Daten erst nach vollständig erfolgter Ausführung der Struktur, nicht bereits nach Ausführung der einzelnen Rahmen, die Struktur verlassen. Daten an Eingangstunneln stehen wie bei Case-Strukturen allen Rahmen zur Verfügung.

Um Daten von einem Rahmen an einen beliebigen nachfolgenden Rahmen zu übergeben, verwenden Sie ein Terminal, das als *lokale Sequenz-Variable* bezeichnet wird. Um eine lokale Sequenz-Variable zu erhalten, wählen Sie **Lokale Sequenz-Variable hinzufügen** aus dem Popup-Menü des Strukturrandes. Dieser Menüpunkt ist nicht verfügbar, wenn Sie das

Popup-Menü einer lokalen Sequenz-Variablen oder des Anzeigefensters des Subdiagramms aufrufen. Sie können das Terminal an eine beliebige unbesetzte Stelle auf dem Rand ziehen. Benutzen Sie zum Entfernen eines Terminals den Befehl **Entfernen** aus dem Popup-Menü der lokalen Sequenz-Variablen.

Ein nach außen weisender Pfeil erscheint in dem Terminal der lokalen Sequenz-Variablen, das zum Rahmen mit der Datenquelle gehört. Das Terminal in nachfolgenden Rahmen enthält einen nach innen weisenden Pfeil, der anzeigt, daß das Terminal eine Quelle dieses Rahmens ist. In Rahmen vor dem Quellrahmen können Sie die lokale Sequenz-Variable nicht verwenden, und sie erscheint als nicht bedienbares Rechteck. Die folgenden Abbildungen zeigen das Terminal der lokalen Sequenz-Variablen.

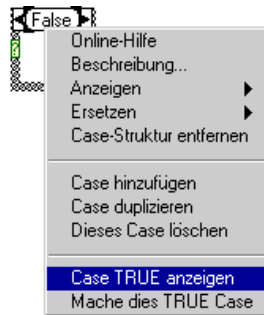


Bearbeiten von Case-Strukturen und Sequenzstrukturen

Da die Bearbeitung und Manipulation der Case-Strukturen und Sequenzstrukturen ähnliche Verfahren umfaßt, werden in den folgenden Beispielen nur die Case-Struktur und ihre Popup-Menüs gezeigt. Ersetzen Sie bei Sequenzstrukturen einfach das Wort *Case* durch das Wort *Rahmen*. Eine neue Case-Struktur umfaßt zwei Cases, kann aber dahingehend bearbeitet werden, daß sie nur einen Case aufweist. Eine neue Sequenzstruktur hat einen Rahmen.

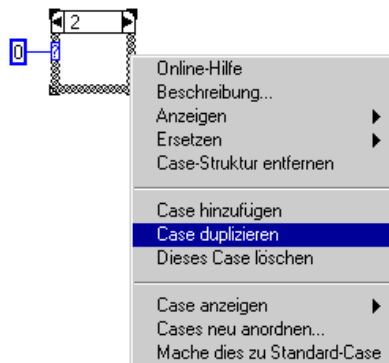
Verschiebungen zwischen Subdiagrammen

Die schnellste Methode, das vorhergehende oder nachfolgende Subdiagramm anzuzeigen, besteht darin, im Anzeigefenster auf die Dekrement- oder Inkrement-Taste zu klicken. Wenn Sie mehrere Subdiagramme überspringen wollen, klicken Sie auf den Identifikator des Subdiagramms, und wählen Sie das Zielsubdiagramm aus dem Popup-Menü (siehe nachfolgende Abbildung). Sie können hierzu auch den Befehl **Case TRUE anzeigen** aus dem Popup-Menü des Randes verwenden.



Hinzufügen von Subdiagrammen

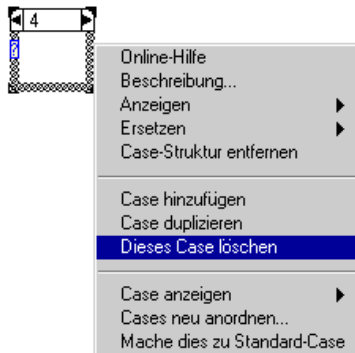
Wenn Sie **Case duplizieren** entweder aus dem Popup-Menü des Strukturrandes oder des Diagramm-Identifikators wählen (siehe Abbildung unten), wird daran anschließend eine Kopie des sichtbaren Subdiagramms eingefügt.



Wenn Sie in einer Sequenzstruktur Subdiagramme hinzufügen oder entfernen, paßt sich der Diagramm-Identifikator automatisch dementsprechend an. Bei Case-Strukturen ist die Reihenfolge der Subdiagramme für die Programmausführung irrelevant, kann aber mit dem Menüpunkt **Cases neu anordnen...** aus dem Popup-Menü geändert werden.

Löschen von Subdiagrammen

Um das sichtbare Subdiagramm zu löschen, wählen Sie aus dem in der folgenden Abbildung gezeigten Popup-Menü des Strukturrandes den Menüpunkt **Dieses Case löschen**. Dieser Befehl ist nicht verfügbar, wenn nur ein Subdiagramm vorhanden ist.

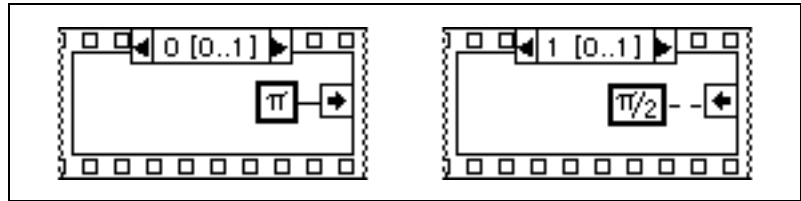


Probleme bei Verbindungen von Strukturen

Die folgenden Abschnitte behandeln fehlerhafte Verbindungen von Strukturen.

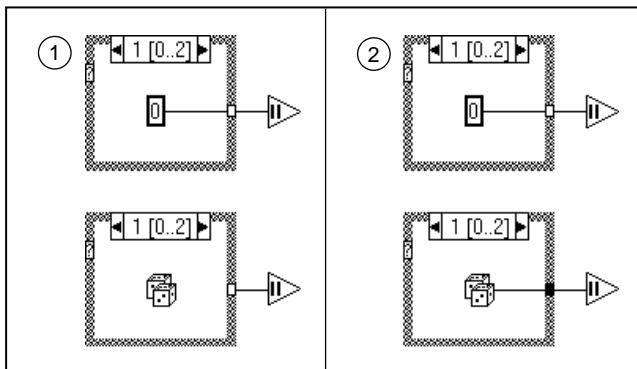
Zuweisen mehrerer Werte zu einer lokalen Sequenz-Variable

Sie können der lokalen Variable einer Sequenzstruktur nur in einem Rahmen einen Wert zuweisen, obgleich Sie den Wert in allen nachfolgenden Rahmen verwenden können. Die Abbildung unten links zeigt den Wert Pi, der der lokalen Sequenz-Variablen in Rahmen 0 zugewiesen ist. Wenn Sie nun versuchen, dieser lokalen Variablen in Rahmen 1 einen anderen Wert zuzuweisen, erzeugen Sie eine ungültige Verbindung. Dieser Fehler ist eine abgewandelte Form des Fehlers aufgrund mehrerer Quellen.



Keine Verbindung zu einem Tunnel in allen Cases einer Case-Struktur

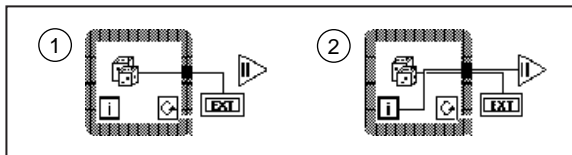
Eine Verbindung von einer Case-Struktur zu einem Objekt außerhalb der Struktur erzeugt einen ungültigen Tunnel, falls nicht für alle Cases eine Quelle mit dem Objekt verbunden wird (siehe Fall 1 des Beispiels unten). Dies ist eine abgewandelte Form des Fehlers aufgrund einer fehlenden Quelle, da mindestens ein Case bei der Ausführung keinen Datenwert erzeugt. Durch eine Verbindung aller Cases zum Tunnel, wie dies in Fall 2 des Beispiels unten gezeigt ist, wird das Problem berichtigt. Dies ist kein Verstoß durch mehrere Quellen, da nur ein Case ausgeführt wird, der nur einen Ausgangswert pro Ausführungsdurchgang der Case-Struktur erzeugt.



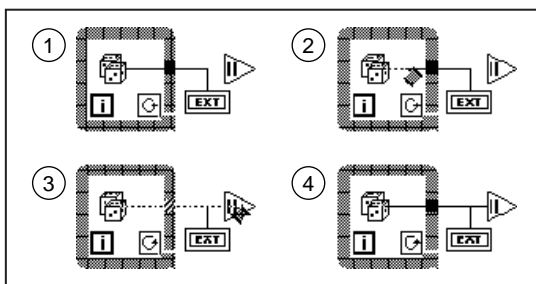
Überlappende Tunnel

Da G beim Erstellen von Verbindungen Tunnel erstellt, überlappen diese manchmal. Überlappende Tunnel beeinträchtigen die Ausführung des Diagramms nicht, aber sie können die Bearbeitung erschweren. Vermeiden Sie es, überlappende Tunnel zu erstellen. Treten Sie dennoch auf, ziehen

Sie einen Tunnel beiseite, um den anderen zu exponieren. Betrachten Sie das folgende Beispiel.



Es läßt sich schwer sagen, welcher Tunnel sich im Vordergrund befindet. Wenn Sie versuchen, eine Verbindung zu einem Tunnel zu erstellen, während Sie überlappen, können Fehler auftreten, obwohl Sie die ungültigen Verbindungen immer entfernen und von vorne anfangen können.

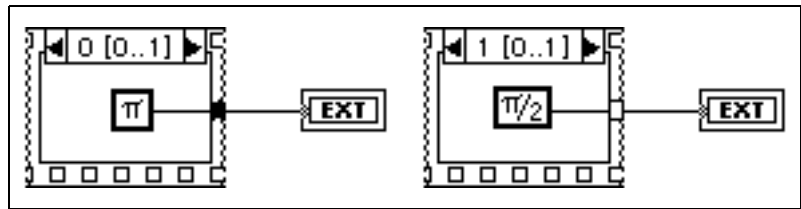


Wenn Sie ein Objekt innerhalb einer Struktur mit einem Objekt außerhalb verbinden müssen, wobei eine solche Verbindung bereits existiert, erzeugen Sie *keine neue Verbindung* durch die Struktur (siehe obige Abbildung). Beginnen Sie die zweite Verbindung statt dessen beim Tunnel. In diesem Beispiel verursachen zwei überlappende Tunnel kein Problem. Handelte es sich hier aber um eine Case-Struktur, könnte es in jedem Case den Anschein haben, als würden zwei überlappende ungültige Tunnel verbunden. Sie können immer alle Verbindungen eines Tunnels entfernen, um diesen verschwinden zu lassen, und die Verbindungen dann erneut richtig plazieren.

Wenn die VI-Tunnel vollständig überlappen, erscheint eine Warnung im Fenster "Fehlerliste", wenn Sie **Warnungen anzeigen** wählen. Weitere Informationen über derartige Probleme finden Sie im Abschnitt [Warnungen](#) in Kapitel 4, *Ausführung und Debugging von VIs und SubVIs*.

Verbindungen von mehreren Rahmen einer Sequenzstruktur

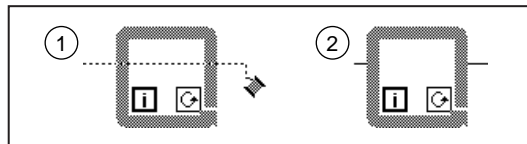
Diese nächste Abbildung zeigt eine weitere Abwandlung des Fehlers aufgrund von mehreren Quellen. Zwei Rahmen einer Sequenzstruktur versuchen, demselben Tunnel Werte zuzuweisen. Der Rahmen wird weiß, um diesen Fehler zu signalisieren.



Verbindungen unterhalb einer Struktur statt durch eine Struktur

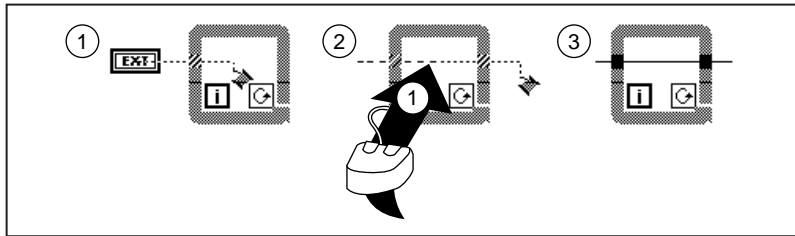
Um eine Verbindung durch eine Struktur zu plazieren, müssen Sie entweder im Innern oder auf den Rand der Struktur klicken (siehe Abbildung unten).

Wenn Sie nicht im Innern oder auf den Rand der Struktur klicken, verläuft die Verbindung, wie hier dargestellt, unter der Struktur hindurch.



Wenn das Verbindungswerkzeug den linken Rand der Struktur überquert, erscheint ein hervorgehobener Tunnel, wodurch angezeigt wird, daß der Editor bei Klicken der Maustaste an dieser Stelle einen Tunnel erstellt. Wenn Sie das Werkzeug, ohne die Maustaste zu klicken, über die Struktur hinweg weiterziehen, bis es den rechten Rand der Struktur berührt, erscheint am rechten Rand ein zweiter hervorgehobener Tunnel. Wenn Sie das Verbindungswerkzeug, ohne zu klicken, über den rechten Rand der Struktur hinausziehen, werden beide Tunnel wieder ausgeblendet, und die Verbindung wird unter der Struktur hinweg verlegt, anstatt hindurch.

Untersuchen Sie die folgende Abbildung.



Wenn Sie die Verbindung innerhalb der Struktur anheften, verläuft sie allerdings auch dann durch die Struktur, wenn Sie das Verbindungswerkzeug über den rechten Rand der Struktur hinausziehen.

Entfernen von Strukturen, ohne Elemente in einer Struktur zu löschen

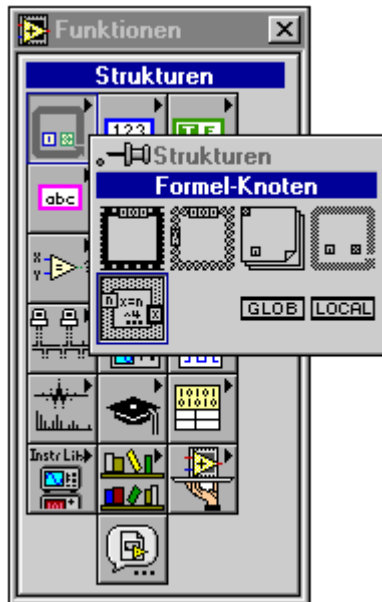
Sie können eine Struktur entfernen (For-Schleife, While-Schleife, Case-Struktur oder Sequenzstruktur), ohne dabei den Inhalt der Struktur zu verlieren. Wenn Sie das Popup-Menü eines dieser Objekte aufrufen, sehen Sie einen Menüpunkt zum Löschen der Struktur. Im Falle von For-Schleifen und While-Schleifen, wird der Inhalt der Schleife in das zugrundeliegende Diagramm kopiert. Außerdem werden alle Tunnel automatisch miteinander verbunden.

Im Falle einer Case-Struktur oder Sequenzstruktur bleibt beim Entfernen der Struktur lediglich der aktuelle Rahmen oder Case erhalten. Alle anderen Rahmen oder Cases werden gelöscht. Sie erhalten eine Warnung darüber, daß verborgene Rahmen oder Cases verlorengehen und erhalten die Gelegenheit, die Operation abubrechen.

Formelknoten



Dieses Kapitel beschreibt die Verwendung des Formelknotens zur Ausführung mathematischer Formeln im Blockdiagramm. Der Formelknoten steht in der Palette **Funktionen»Strukturen** zur Verfügung.



Verwendung von Formelknoten

Der Formelknoten ist ein den vier Strukturen (For-Schleife, While-Schleife, Case-Struktur und Sequenzstruktur) ähnliches, in der Größe veränderbares Feld. Statt eines Subdiagramms enthält der Formelknoten

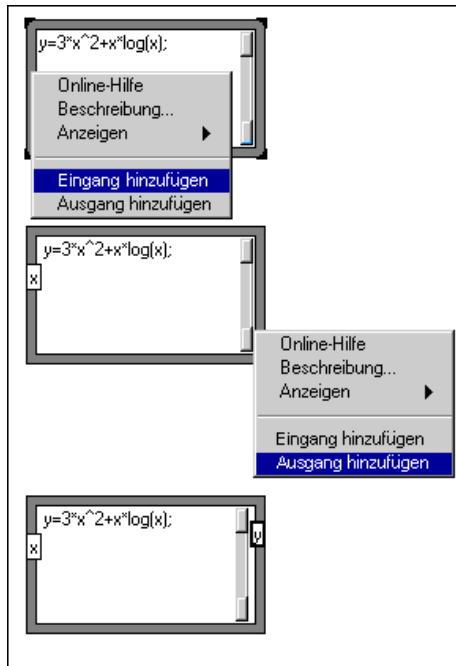
jedoch eine oder mehrere Formelanweisungen, die durch Semikolon getrennt werden (siehe nachstehendes Beispiel).

```
y=3*x^2+x*log(x);
```

Formelanweisungen verwenden eine Syntax, die der für arithmetische Ausdrücke in textbasierten Programmiersprachen ähnlich ist. Sie können Kommentare hinzufügen, indem Sie diese mit Schrägstrichen-Sternchen (`/*Kommentar*/`) umgeben.

Ein Beispiel für ein VI, das einen Formelknoten verwendet, finden Sie in `examples\general\structs.llb\Equations.vi`.

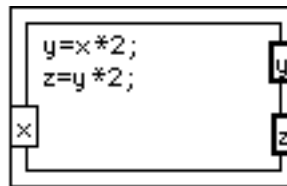
Das Popup-Menü am Rand des Formelknotens enthält Menüpunkte, mit denen Eingangs- und Ausgangsvariablen hinzugefügt werden können (siehe nachstehende Abbildung).



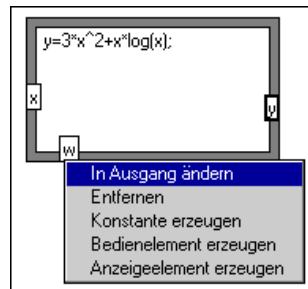
Ausgangsvariable sind durch einen dickeren Rand gekennzeichnet.

Es gibt keine Grenze für die Anzahl der Variablen oder Formeln in einem Formelknoten. Weder zwei Eingänge noch zwei Ausgänge dürfen denselben Namen haben. Ein Ausgang kann jedoch denselben Namen haben wie ein Eingang.

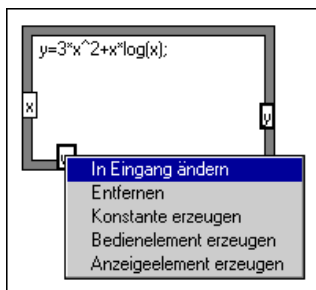
Jede Variable, die in einem Formelknoten für eine Berechnung verwendet wird, muß als Eingang oder Ausgang deklariert werden. Zwischenvariable, d.h. Variable, die Ergebnissen von Operationen zugewiesen sind, die nach Eingängen in den Knoten, aber vor dem endgültigen Ausgang aus dem Knoten berechnet werden, müssen als Ausgänge deklariert werden. Es ist jedoch nicht notwendig, Zwischenvariablen mit Knoten außerhalb des Formelknotens zu verbinden. In der folgenden Abbildung müssen sowohl die y -Variable als auch die z -Variable selbst dann als Ausgänge deklariert werden, wenn der Wert von y nicht mit externen Knoten verbunden ist.



Durch Wählen von **In Ausgang ändern** aus dem Popup-Menü (siehe nachstehende Abbildung) können Sie einen Eingang in einen Ausgang umwandeln.



Durch Wählen von **In Eingang ändern** aus dem Pop-up-Menü (siehe nachstehende Abbildung) können Sie einen Ausgang in einen Eingang umwandeln.



Alle Variablen sind Fließkomma-Skalare, deren Präzision von der Konfiguration Ihres Computers abhängig ist. Variablen haben keine Einheiten. Alle Eingangsvariablen, die in den Formeln erscheinen, müssen verbunden werden. Alle verbundenen Ausgangsvariablen müssen mindestens in einer Anweisung zugewiesen werden; dies bedeutet, daß sie auf der linken Seite des Gleichheitszeichens stehen müssen. Eine Ausgangsvariable kann zwar in einem Ausdruck auf der rechten Seite des Gleichheitszeichens vorkommen, G prüft aber nicht nach, ob sie in einer vorherigen Anweisung zugewiesen worden ist. Erfolgt eine Zuweisung als Unterausdruck, ist der Wert des Unterausdrucks der zugewiesene Wert; zum Beispiel:

$$x = \sin(y = \pi/3)$$

weist y den Wert $\frac{\pi}{3}$ und dann x den Wert $\sin\left(\frac{\pi}{3}\right)$ zu.

Tritt ein Syntaxfehler auf, können Sie auf die Taste **Unterbrochene Ausführung** klicken, um die Fehlerliste aufzurufen. In der Liste zeigt der Formelknoten einen Abschnitt der Formel mit einem #-Zeichen an, womit der Punkt markiert wird, an dem das Formelfeld den Fehler festgestellt hat.

Funktionen und Operatoren des Formelknotens

Alle Funktionsnamen müssen in Kleinbuchstaben geschrieben werden. Tabelle 20-1 enthält eine Übersicht über die Funktionen des Formelknotens.

Tabelle 20-1. Funktionen des Formelknotens

Funktion	Entsprechender G-Funktionsname	Beschreibung
abs(x)	Absoluter Wert	Gibt den absoluten Wert von x zurück.
acos(x)	Inverser Kosinus	Berechnet den inversen Kosinus von x im Bogenmaß.
acosh(x)	Inverser Kosinus hyperbolikus	Berechnet den inversen Kosinus hyperbolikus von x im Bogenmaß.
asin(x)	Inverser Sinus	Berechnet den inversen Sinus von x im Bogenmaß.
asinh(x)	Inverser Sinus hyperbolikus	Berechnet den inversen Sinus hyperbolikus von x im Bogenmaß.
atan(x)	Inverser Tangens	Berechnet den inversen Tangens von x im Bogenmaß.
atanh(x)	Inverser Tangens hyperbolikus	Berechnet den inversen Tangens hyperbolikus von x im Bogenmaß.
ceil(x)	Auf nächstgrößere Zahl runden	Rundet x auf den nächstgrößeren Integer-Wert (kleinster Integer-Wert ist größer oder gleich x).
cos(x)	Kosinus	Berechnet den Kosinus von x im Bogenmaß.
cosh(x)	Kosinus hyperbolikus	Berechnet den Kosinus hyperbolikus von x im Bogenmaß.

Tabelle 20-1. Funktionen des Formelknotens (Fortsetzung)

Funktion	Entsprechender G-Funktionsname	Beschreibung
$\cot(x)$	Kotangens	Berechnet den Kotangens von x im Bogenmaß ($1/\tan(x)$).
$\csc(x)$	Kosekans	Berechnet den Kosekans von x im Bogenmaß ($1/\sin(x)$).
$\exp(x)$	Exponentialfunktion	Berechnet den Wert von e hoch x .
$\expm1(x)$	Exponentialfunktion (Arg) - 1	Berechnet den Wert von e hoch x minus eins ($e^x - 1$).
$\text{floor}(x)$	Auf nächstkleinere Zahl runden	Rundet x auf den nächstkleineren Integer-Wert (größter Integer-Wert ist kleiner als oder gleich x).
$\text{getexp}(x)$	Mantisse & Exponent	Gibt den Exponent von x zurück.
$\text{getman}(x)$	Mantisse & Exponent	Gibt die Mantisse von x zurück.
$\text{int}(x)$	Auf nächste ganze Zahl runden	Rundet das Argument auf die nächste ganze Zahl.
$\text{intrz}(x)$	Gegen 0 runden	Rundet x auf die nächste ganze Zahl zwischen x und Null.
$\ln(x)$	Natürlicher Logarithmus	Berechnet den natürlichen Logarithmus von x (zur Basis e).
$\lnp1(x)$	Natürlicher Logarithmus (Arg + 1)	Berechnet den natürlichen Logarithmus von $(x + 1)$.
$\log(x)$	Logarithmus der Basis 10	Berechnet den Logarithmus von x (zur Basis 10).
$\log2(x)$	Logarithmus der Basis 2	Berechnet den Logarithmus von x (zur Basis 2).
$\max(x,y)$	Max & Min	Vergleicht x und y und gibt den größeren Wert zurück.
$\min(x,y)$	Max & Min	Vergleicht x und y und gibt den kleineren Wert zurück.

Tabelle 20-1. Funktionen des Formelknotens (Fortsetzung)

Funktion	Entsprechender G-Funktionsname	Beschreibung
<code>mod(x,y)</code>	Quotient & Rest	Berechnet den Rest von x/y , wenn der Quotient nach unten gerundet wird.
<code>rand()</code>	Zufallszahl (0 bis 1)	Erzeugt eine Fließkommazahl zwischen (ausschließlich) 0 und 1.
<code>rem(x,y)</code>	Rest	Berechnet den Rest von x/y , wenn der Quotient auf die nächste ganze Zahl gerundet wird.
<code>sec(x)</code>	Sekans	Berechnet den Sekans von x im Bogenmaß ($1/\cos(x)$).
<code>sign(x)</code>	Vorzeichen	Gibt 1 zurück, falls x größer als 0 ist, 0 falls x gleich 0 ist und -1 , falls x kleiner als 0 ist.
<code>sin(x)</code>	Sinus	Berechnet den Sinus von x im Bogenmaß.
<code>sinc(x)</code>	Si	Berechnet den Sinus von x geteilt durch x im Bogenmaß ($\sin(x)/x$).
<code>sinh(x)</code>	Sinus hyperbolicus	Berechnet den Sinus hyperbolicus von x im Bogenmaß.
<code>sqrt(x)</code>	Quadratwurzel	Berechnet die Quadratwurzel von x .
<code>tan(x)</code>	Tangens	Berechnet den Tangens von x im Bogenmaß.
<code>tanh(x)</code>	Tangens hyperbolicus	Berechnet den Tangens hyperbolicus von x im Bogenmaß.
<code>x^y</code>	x^y	Berechnet den Wert von x hoch y .

Syntax von Formelknoten

Die Syntax von Formelknoten wird nachstehend in BNF-Notation (Bakus–Naur Form) zusammengefaßt. Eckige Klammern beinhalten optionale Elemente.

```

<assignlst> := <outputvar> = <aexpr> ; [ <assignlst> ]
<aexpr> := <expr> | <outputvar> = <aexpr>
<expr> := <expr> <binaryoperator> <expr>
        | <unaryoperator> <expr>
        | <expr> ? <expr> : <expr>
        | ( <expr> )
        | <inputvar>
        | <outputvar>
        | <const>
        | <function> ( <arglist> )
<binaryoperator> := + | - | * | / | ^ | != | ==
        | > | < | >= | <= | && | ||
<unaryoperator> := + | - | !
<arglist> := <aexpr> [ , <arglist> ]
<const> := pi | <number>

```

In der folgenden Liste sind die Operatoren nach Vorrang (von niedrigster zu höchster Priorität) aufgeführt.

=	Zuweisung
? :	Bedingung
	Logisches ODER
&&	Logisches UND
!= ==	Ungleichheit, Gleichheit
< > <= >=	andere Relationen: kleiner als, größer als, kleiner als oder gleich, größer als oder gleich
+ -	Addition, Subtraktion
* /	Multiplikation, Division
+ - !	unär: positiv, negativ, logische Negation
^	Potenzierung

Potenzierung und der Zuweisungsoperator sind rechtsassoziativ (Gruppierung von rechts nach links). Alle anderen binären Operatoren sind linksassoziativ. Der numerische Wert von TRUE ist 1 und von FALSE 0 (für Ausgänge). Der logische Wert von 0 ist FALSE, und jede Zahl ungleich 0 ist TRUE. Der logische Wert des Bedingungsausdrucks

`<lexpr> ? <texpr>: <fexpr>`

ist `<texpr>`, falls der logische Wert von `<lexpr>` TRUE ist, andernfalls `<fexpr>`.

Fehler in Formelknoten

Tabelle 20-2 listet vom Formelknoten feststellbare Fehler auf.

Tabelle 20-2. Fehler in Formelknoten

Fehlermeldung	Bedeutung der Fehlermeldung
Syntax-Fehler	Falsch gebrauchter Operator usw.
Falscher Token	Unbekanntes Zeichen.
Es ist eine Ausgangsvariable erforderlich	Die Zuweisung zu einer Eingangsvariable ist nicht zulässig.
Fehlende Ausgangsvariable	Es wurde versucht, eine Zuweisung zu einer nicht existierenden Ausgangsvariablen zu definieren.
Fehlende Variable	Bezug zu einer nicht existierenden Eingangs- oder Ausgangsvariablen.
Zu wenig Argumente	Nicht genügend Argumente für eine Funktion.
Zu viele Argumente	Zu viele Argumente für eine Funktion.
Nicht abgeschlossene Argumentenliste	Formel endet ohne eine schließende Klammer der Argumentenliste.
Fehlende Klammer (links)	Dem Funktionsnamen folgt keine Argumentenliste.
Fehlende Klammer (rechts)	Formel endet, ohne daß alle Klammern geschlossen sind.
Fehlender Doppelpunkt	Falscher Gebrauch des Bedingungsoperators bei drei Argumenten.
Fehlendes Semikolon	Formelanweisung endet nicht mit einem Semikolon.
Fehlendes Gleich-Zeichen	Formelanweisung ist keine gültige Zuweisung.

VI-Server

Dieses Kapitel beschreibt den Mechanismus zur programmatischen Steuerung von VIs und Anwendungen. Das Kapitel beschreibt außerdem das Verfahren der Remote-Steuerung von VIs oder Anwendungen.

Der Zugriff auf viele der G-Funktionen wird programmatisch mit dem VI-Server ermöglicht. Sie können von einem beliebigen ActiveX-Client auf der Windows 95/NT-Plattform auf diese Funktionen zugreifen. Mit Hilfe einiger Funktionen der Sprache G können Sie auf allen Plattformen auch auf alle VI-Serverkapazitäten zugreifen. Diese Funktionen machen es auch möglich, beinahe alle VI-Serveroperationen in einer lokalen Version von BridgeVIEW oder LabVIEW sowie eine beliebige Remote-Version in einem TCP/IP-Netzwerk durchzuführen. Weitere Informationen über ActiveX-Server- und Clientfähigkeiten finden Sie im *LabVIEW Benutzerhandbuch*, im *BridgeVIEW User Manual* sowie in der **Online-Referenz** Ihrer Software.

Verwendung des VI-Servers

Nachstehend einige der Aufgaben, die Sie mit dem VI-Server bewältigen können.

- Sie können VIs dynamisch in den Arbeitsspeicher laden, anstatt sie statisch mit der Anwendung zu verknüpfen, und dann wie einen normalen SubVI-Aufruf mit den VI-Serverfunktionen aufrufen. Dies kann dann nützlich sein, wenn Sie es mit einer großen Anwendung zu tun haben und Arbeitsspeicher oder Startzeit einsparen möchten. Sie können die Speicherauslastung Ihrer Anwendung und den Zeitaufwand zum Laden der Anwendung in den Arbeitsspeicher dadurch reduzieren, daß Sie selten gebrauchte Komponenten der Anwendung, wie einen Satz von Konfigurationsdialogfeldern, nur bei Bedarf laden und ausführen. Wenn der Benutzer die Operationen beendet, können Sie die VIs freigeben und den Arbeitsspeicher wieder freisetzen.
- Aspekte der Benutzerschnittstelle eines VIs lassen sich programmatisch steuern. So werden Sie eventuell die Position eines VI-Fensters dynamisch bestimmen oder durch ein Panel scrollen wollen, so daß ein bestimmter Teil des Panels sichtbar ist, oder das

Panelfenster schließen oder öffnen. All diese Eigenschaften eines VI-Frontpanels lassen sich mit dem VI-Server steuern.

- Sie können leicht eine Serveranwendung erstellen, die Funktionen exportiert, die von BridgeVIEW oder LabVIEW im Internet aufgerufen werden können. Angenommen, Sie verfügen über eine Anwendung zur Datenerfassung, die Daten von Remote-Standorten erfaßt und protokolliert, und Sie möchten diese Daten gelegentlich von Ihrem Computer aus abrufen. Mit Hilfe einer einfachen Voreinstellung können Sie manche VIs über das Internet aufrufbar machen, so daß die Übertragung der neuesten Daten so einfach ist wie ein SubVI-Aufruf. Der VI-Server kümmert sich um alle Netzwerkeinheiten und ermöglicht die Ausführung unabhängig von der Plattform, auf der der Client oder Server ausgeführt wird.
- Sie können Eigenschaften eines VIs programmatisch ändern und diese Änderungen auf dem Datenträger speichern. Angenommen, Sie wollen bei der Entwicklung Ihrer Anwendung VIs so konfigurieren, daß Debugging verfügbar ist, Run-Time Popup-Menüs verfügbar sind, Rollbalken sichtbar und Fenster in der Größe veränderbar sind. Wenn Sie die Anwendung jedoch verteilen, möchten Sie diese Funktionen aber womöglich ausschalten und gleichzeitig sicherstellen, daß bestimmte andere Eigenschaften richtig eingestellt sind, wie z.B. ob ein VI ablauffinvariant ist, auf welches Ausführungssystem das VI eingestellt ist sowie der Pfad der Hilfedatei eines VIs. All diese Eigenschaften können mit dem VI-Server programmatisch eingestellt und abgefragt werden, was es Ihnen ermöglicht, Anwendungen zu schreiben, die VIs bearbeiten, anstatt dies für jedes einzelne VI mit dem Dialogfeld "VI-Einstellungen" tun zu müssen.
- Sie können eine Plugin-Architektur für Ihre Anwendung erstellen, damit Sie der Anwendung auch nach der Auslieferung an Kunden Funktionen hinzufügen können. Angenommen, Sie besitzen einen Satz von Daten, die VIs filtern und alle dieselben Parameter verwenden. Wenn Sie die Anwendung so gestalten, daß diese Filter aus einem Plugin-Verzeichnis dynamisch geladen werden, können Sie Ihre Anwendung mit einem Teilsatz dieser Filter ausliefern und Benutzern zusätzliche Filterungsoptionen einfach dadurch zur Verfügung stellen, daß Sie die neuen Filter-VIs in dem Plugin-Verzeichnis ablegen.

Fähigkeiten des VI-Servers

Zugriff auf die Funktionalität des VI-Servers erfolgt über Referenzen zu zwei Hauptklassen von Objekten: das Anwendungsobjekt und das VI-Objekt. Wenn Sie eine Referenz zu einem dieser Objekte erstellen, können Sie sie an eine Funktion übergeben, die das Objekt verarbeitet. Ist dieser Vorgang abgeschlossen, schließen Sie die Referenz. Diese Programmierkonvention ist der Datei I/O und Netzwerkreferenzen ähnlich.

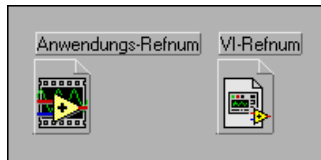
Ein wichtiger Aspekt sowohl der Anwendungs- als auch der VI-Referenzen ist ihre Netzwerktransparenz. Dies bedeutet, daß Sie Referenzen zu Objekten auf Remote-Computern auf dieselbe Art und Weise öffnen können wie Referenzen zu den Objekten auf Ihrem eigenen Computer. Nachdem Sie eine Referenz zu einem Remote-Objekt geöffnet haben, können Sie es mit wenigen Einschränkungen in genau derselben Weise behandeln wie ein lokales Objekt. Bei Operationen an einem Remote-Objekt kümmert sich der VI-Server darum, die Informationen zur Operation über das Netzwerk zu senden und die Ergebnisse zurückzusenden. Ihr Programm sieht praktisch identisch aus, ganz gleich ob die Operation über das Netzwerk oder lokal abläuft.

Mit einer Referenz zu einer G-Anwendung können Sie Informationen über die G-Umgebung abrufen, wie z.B. die Plattform, auf der die Anwendung ausgeführt wird, die Version von BridgeVIEW oder LabVIEW oder die Liste aller gegenwärtig in den Arbeitsspeicher geladenen VIs. Sie können auch Informationen wie z.B. den aktuellen Benutzernamen oder die Liste der in andere Anwendungen exportierten VIs einstellen.

Verfügen Sie über eine Referenz zu einem VI, laden Sie sie in den Arbeitsspeicher. Sobald die Referenz geöffnet ist, bleibt das VI solange im Arbeitsspeicher, bis Sie die Referenz schließen. Es besteht die Möglichkeit, daß gleichzeitig mehrere Referenzen zu einem VI geöffnet sind. In diesem Fall verbleibt das VI solange im Arbeitsspeicher, bis alle Referenzen zum VI geschlossen sind. Mit einer Referenz zu einem VI können Sie alle Eigenschaften eines VIs, die in den **VI-Einstellungen** verfügbar sind, sowie dynamische Eigenschaften, wie z.B. die Fensterposition des Frontpanels, abrufen und einstellen. Sie können das VI auch programmatisch drucken, an einem anderen Speicherort speichern und seine Strings zu Übersetzungszwecken exportieren und importieren.

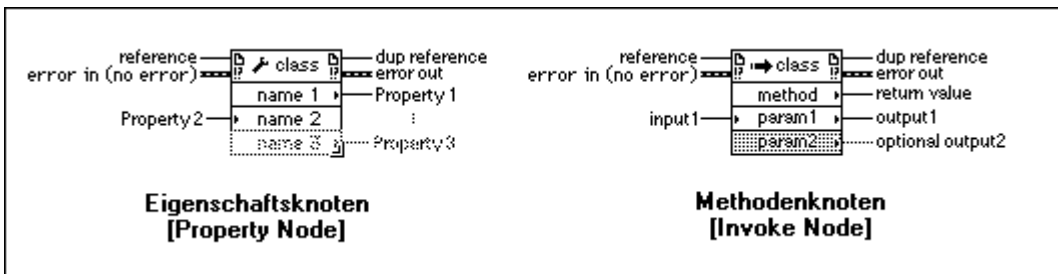
Anwendungs- und VI-Referenzen

In den meisten Anwendungen erstellen Sie eine Referenz zu einem VI mit Hilfe der Funktion “Öffnen der Anwendungsreferenz” bzw. der Funktion “Öffnen der VI Referenz”. Wenn Sie die Referenz auf dem Frontpanel anzeigen wollen, wählen Sie **Pfad & RefNum** aus der Palette **Elemente**. Wählen Sie die Anwendung oder VI-RefNum. Der Datentyp dieser Refnum wird standardmäßig als Anwendungs-Refnum angezeigt. Wenn Sie dies in eine VI-Refnum ändern wollen, rufen Sie das Popup-Menü der Refnum auf, und wählen Sie **VI-Serverklasse auswählen» Virtuelles Instrument**.



Verwendung des Eigenschaftsknotens und Methodenknotens bei Anwendungs- und VI-Referenzen

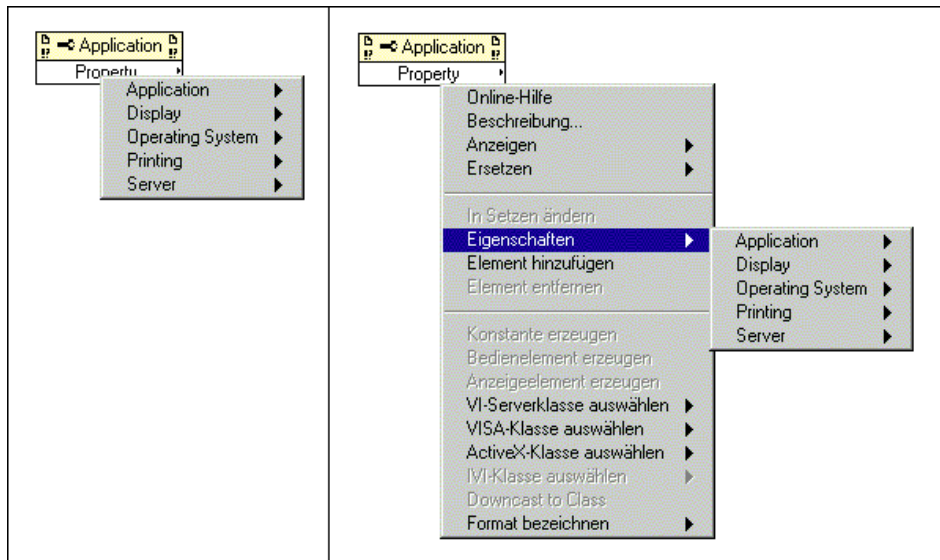
Viele der für Anwendungs- und VI-Referenzen verfügbaren Operationen sind nur über den “Eigenschaftsknoten” und “Methodenknoten” verfügbar. Diese Knoten sind insofern ähnlich, als sie beide oben im Knoten zwei Eingänge und zwei Ausgänge haben und darunter eine veränderliche Liste von Ein- und Ausgängen.



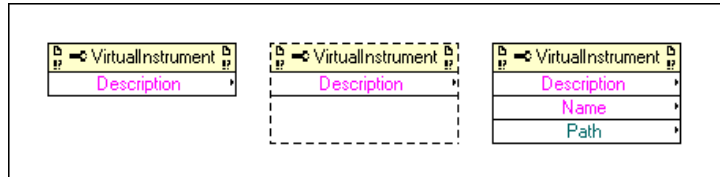
Referenz ist eine Anwendungs- oder VI-Referenz und **Referenz-Duplikat** ist eine Kopie der Referenz, die dazu verwendet wird, den Wert an andere Funktionen zu übergeben. **Fehlereingang** ist der standardmäßige Fehlercluster, der Informationen darüber, ob ein Fehler aufgetreten ist, sowie den Code und eine Beschreibung des Fehlers enthält. Existiert ein Fehler im **Fehlereingang**, unternimmt der Knoten nichts und übergibt den Fehler einfach an den **Fehlerausgang**. Tritt während der Ausführung des Knotens ein Fehler auf, enthält andernfalls der **Fehlerausgang** Informationen über den Fehler.

Beide Knoten sind in Bezug auf den Referenzeingang polymorph. Wenn Sie eine Anwendungs- oder VI-Refnum mit diesem Eingang verbinden, paßt sich der Knoten automatisch an diesen Datentyp an und stellt nur diejenigen Operationen zur Verfügung, die für diesen Typ gültig sind.

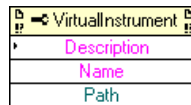
Mit dem Eigenschaftsknoten können Sie verschiedene Eigenschaften einer Anwendung oder eines VIs abrufen (lesen) oder einstellen (schreiben). Sie können die Eigenschaften aus dem Popup-Menü des Knotens wählen, indem Sie mit dem Bedienwerkzeug einfach auf ein Eigenschaftsterminal klicken oder das Popup-Menü des Terminals aufrufen.



Sie können mit einem einzigen Knoten mehrere Eigenschaften abrufen oder setzen. Wenn Sie einen Eigenschaftsknoten vergrößern, werden neue Terminals hinzugefügt.

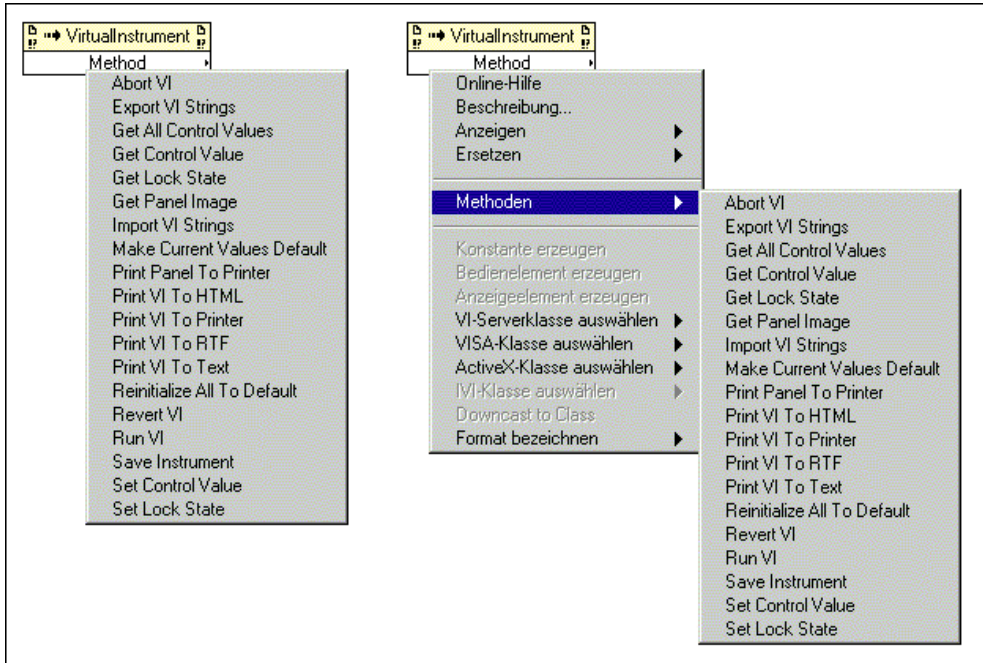


Sie können in einem Eigenschaftsknoten Eigenschaften lesen und schreiben (siehe nachstehende Abbildung). Die Eigenschaften, die Sie lesen, werden durch einen kleinen Richtungspfeil auf der rechten Seite gekennzeichnet, während die Eigenschaften, die Sie schreiben, links einen kleinen Richtungspfeil haben. Sie wählen, ob Eigenschaften gelesen oder geschrieben werden sollen, indem Sie **In Lesen ändern** oder **In Setzen ändern** aus dem Popup-Menü wählen.

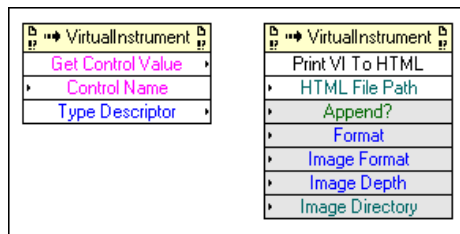


Der Knoten wird von oben nach unten ausgeführt. Tritt im Verlauf der Knotenausführung ein Fehler auf, werden die verbleibenden Eigenschaften ignoriert, und es wird ein Fehler zurückgegeben. Der Fehlerstring enthält Informationen darüber, welche Eigenschaft den Fehler verursacht hat.

Mit dem Methodenknoten können Sie Aktionen oder Methoden an einer Anwendung oder einem VI durchführen. Im Gegensatz zum Eigenschaftsknoten führt ein einzelner Methodenknoten nur eine einzige Methode an einer Anwendung oder einem VI durch. Sie greifen auf die verfügbaren Methoden zu, indem Sie mit dem Bedienwerkzeug auf ein Methodenterminal klicken oder indem Sie das Popup-Menü des Terminals aufrufen, welches dem des Eigenschaftsknotens entspricht.



Der Name der Methode ist stets das erste Terminal in der Parameterliste. Wenn die Funktion einen Wert zurückgibt, ist das Ergebnis der Wert des Methodenterminals. Es wird zum Beispiel ein Wert von der Methode **Get Control Value** zurückgegeben. Ansonsten besitzt das Methodenterminal keinen Wert, wie **Print VI To HTML**.



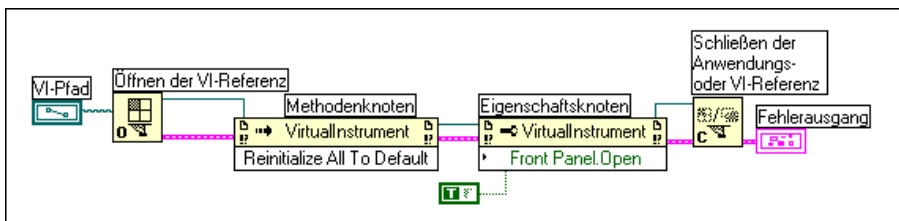
Der Methodenknoten listet die Parameter untereinander auf, wobei die optionalen Parameter am Ende in grauen Feldern aufgeführt sind. In der obigen Abbildung ist **HTML File Path** ein erforderlicher Parameter, während **Append?**, **Format**, **Image Format**, **Image Depth** und **Image Directory** optionale Parameter sind.

Beispiele für Eigenschaften und Methoden der VI-Klasse und Anwendungsklasse

In den folgenden Abschnitten werden allgemeine Anwendungen der VI-Klasse und Anwendungsklasse behandelt. Andere Anwendungen, in denen Eigenschaften und Methoden der VI-Klasse und Anwendungsklasse verwendet werden, finden Sie unter `examples\general\viserver`.

Manipulation von Eigenschaften und Methoden der VI-Klasse

Sie können Eigenschaften eines VIs unabhängig von der Durchführung von Methoden an einem VI setzen oder abrufen. In manchen Anwendungen ist beides gewünscht, der Zugriff auf VI-Eigenschaften und die Durchführung einer Methode an einem VI. In dem folgenden Diagramm werden die Frontpanel-Objekte in einem VI auf die Standardwerte zurückgesetzt, und dann wird das Frontpanel geöffnet, um diese Standardwerte anzuzeigen.



Bevor Sie auf Eigenschaften und Methoden eines VIs zugreifen, müssen Sie durch die Ausführung von “Öffnen der VI Referenz” eine Referenz zu diesem VI erstellen. Um eine Methode zu einem VI aufzurufen, verwenden Sie den Methodenknoten. Sobald die Verbindung zwischen “Öffnen der VI Referenz” und dem Methodenknoten hergestellt ist, können Sie auf alle Methoden der VI-Klasse zugreifen. Sie können auch das Popup-Menü des Knotens aufrufen und **VI-Serverklasse auswählen** **Virtuelles Instrument** wählen, um Zugriff auf die Methoden der VI-Klasse zu erhalten.

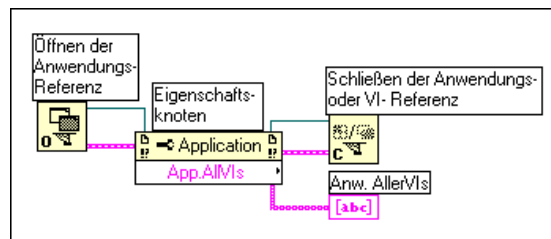


Der Eigenschaftsknoten funktioniert ähnlich wie der Methodenknoten. Sobald Sie eine VI-Referenz mit dem Eigenschaftsknoten verbunden haben, können Sie auf alle Eigenschaften der VI-Klasse zugreifen. Auch hier können Sie **VI-Serverklasse auswählen» Virtuelles Instrument** aus dem Popup-Menü des Knotens wählen.

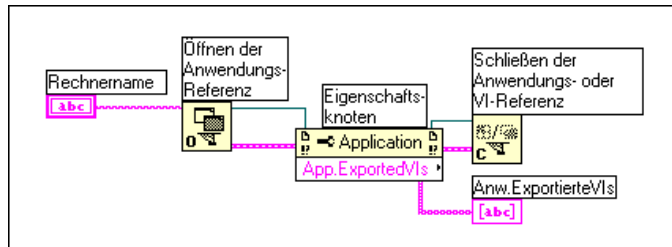
Um Eigenschaftswerte zu setzen, vergewissern Sie sich, daß sich der Pfeil auf der linken Seite befindet, indem Sie das Popup-Menü aufrufen und “In Setzen ändern” wählen. Prüfen Sie stets auf mögliche Fehler. Weder der Methodenknoten noch der Eigenschaftsknoten können ausgeführt werden, wenn vor der Ausführung ein Fehler auftritt. Wenn der Fehler in einem dieser Knoten auftritt, wird die Eigenschaft oder Methode, bei der der Fehler aufgetreten ist, in der Fehlermeldung angegeben.

Manipulation von Eigenschaften und Methoden der Anwendungsklasse

Sie können Eigenschaften auf einem lokalen oder Remote LabVIEW-Computer setzen oder abrufen, unabhängig von der Durchführung von Methoden an LabVIEW. Im folgenden Diagramm werden die lokal im Arbeitsspeicher befindlichen VIs auf dem Frontpanel in einem Stringarray angezeigt.



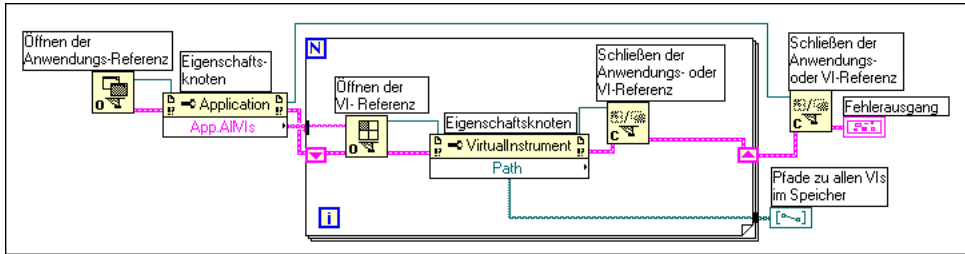
Wenn Sie die auf einem Remote-Computer im Arbeitsspeicher befindlichen VIs suchen wollen, verbinden Sie ein String-Bedienelement mit dem Eingang des Computernamens, und geben Sie die numerische IP-Adresse oder den Domännennamen des Computers ein (siehe nachstehende Abbildung). Sie müssen auch die Eigenschaft in **Exportierte VIs im Speicher** ändern, da sich die Eigenschaft **Alle VIs im Speicher** der vorherigen Abbildung auf lokale Versionen von LabVIEW und BridgeVIEW bezieht.



Prüfen Sie stets auf mögliche Fehler. Der Eigenschaftsknoten kann nicht ausgeführt werden, wenn vor der Ausführung ein Fehler auftritt. Wenn der Fehler im Eigenschaftsknoten auftritt, wird die Eigenschaft, bei der der Fehler aufgetreten ist, in der Fehlermeldung angegeben.

Manipulation von Eigenschaften und Methoden der VI-Klasse und Anwendungsklasse

Sie können Eigenschaften und Methoden der VI-Klasse und Anwendungsklasse separat verwenden. In manchen Anwendungen müssen Sie von beiden Klassen aus auf Eigenschaften und Methoden zugreifen. Im nachfolgenden Blockdiagramm werden die auf dem lokalen Computer im Arbeitsspeicher befindlichen VIs ermittelt und der Pfad jedes dieser VIs auf dem Frontpanel angezeigt. Um alle im Arbeitsspeicher befindlichen VIs zu finden, müssen Sie auf eine Eigenschaft vom Typ Anwendungsklasse zugreifen. Um die Pfade all dieser VIs zu ermitteln, müssen Sie auf eine Eigenschaft vom Typ VI-Klasse zugreifen. Die Anzahl der im Arbeitsspeicher befindlichen VIs bestimmt die Anzahl der Durchläufe der For-Schleife. “Öffnen der VI-Referenz” und “Schließen der VI-Referenz” müssen in der For-Schleife enthalten sein, da Sie für jedes im Arbeitsspeicher befindliche VI eine VI-Referenz benötigen. Es ist am besten, die Anwendungsreferenz erst dann zu schließen, wenn alle VI-Pfade gesammelt sind.



Streng typisierte VI-Refnums

Streng typisierte Refnums brauchen nur bei Anwendungen verwendet zu werden, in denen Sie ein VI dynamisch aufrufen. In solchen Anwendungen gibt es zwei Situationen, in denen sie verwendet werden. Die erste und vielleicht eher typische Situation wäre, daß Sie eine streng typisierte VI-Referenz als Parameter an ein SubVI übergeben möchten. In diesem Fall verbinden Sie das Bedienelement der streng typisierten VI-Refnum mit dem Anschlußfeld des VI und verbinden das Refnum-Terminal mit dem Eingang eines Knotens vom Typ "Aufruf bei Referenz". Der Refnum-Wert wird in diesem Fall dazu verwendet, festzustellen, welches VI aufgerufen wird.

Die zweite Situation tritt ein, wenn Sie eine streng typisierte VI-Refnum mit dem Eingang des Typ-Bezeichners der Funktion "Öffnen der VI Referenz" verbinden. In diesem Fall wird der Wert des Bedienelements ignoriert — nur der Typ der Refnum wird von dieser Funktion verwendet. Er wird dazu verwendet, um festzustellen, ob das geöffnete VI dasselbe Anschlußfeld besitzt wie das der streng typisierten VI-Refnum.

Um eine streng typisierte Refnum zu erstellen, legen Sie eine Refnum auf dem Frontpanel ab, indem Sie **Elemente»Pfad & RefNum»Anwendungs- oder VI-Refnum** wählen. Rufen Sie das Pop-up-Menü auf, und wählen Sie **VI-Serverklasse auswählen»Durchsuchen....** Das Dialogfeld **Wählen Sie ein VI** wird angezeigt und fordert Sie zur Auswahl eines VIs auf.

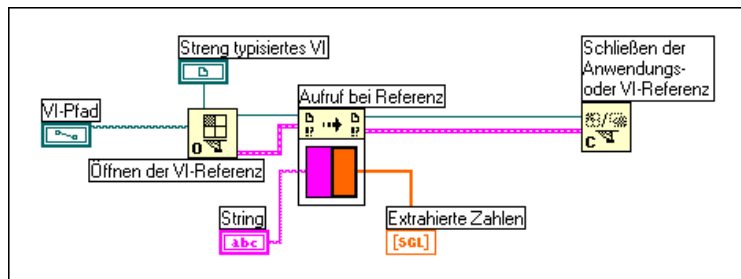


Beachten Sie, daß nur die Informationen über das Anschlußfeld gespeichert werden, obwohl Sie ein VI für streng typisierte Refnums angeben. Das bedeutet, daß keine permanente Verknüpfung zwischen der Refnum und dem VI hergestellt wird. Verwechseln Sie insbesondere die Auswahl eines VI-Anschlußfeldes nicht mit dem Erhalt einer Referenz zum gewählten VI. Sie geben mit dem Eingang **VI Pfad** der Funktion “Öffnen der VI Referenz” ein bestimmtes VI an.

Nachdem Sie Anschlußfelder für streng typisierte Refnums gewählt haben, wird das Anschlußfeld im Untermenü **VI-Serverklasse auswählen»** **Streng typisierte VIs** der VI-Refnum gespeichert. Wenn Sie BridgeVIEW oder LabVIEW beenden, bleibt diese Auswahl von Anschlußfeldern nicht bis zum nächsten Start der Anwendung erhalten.

Beispiel für streng typisierte VI-Refnums

Der einzige Anwendungsfall, bei dem Sie streng typisierte VI-Refnums verwenden, ist der dynamische Aufruf eines VIs. Das nachfolgende Blockdiagramm veranschaulicht dies anhand eines Beispiels.



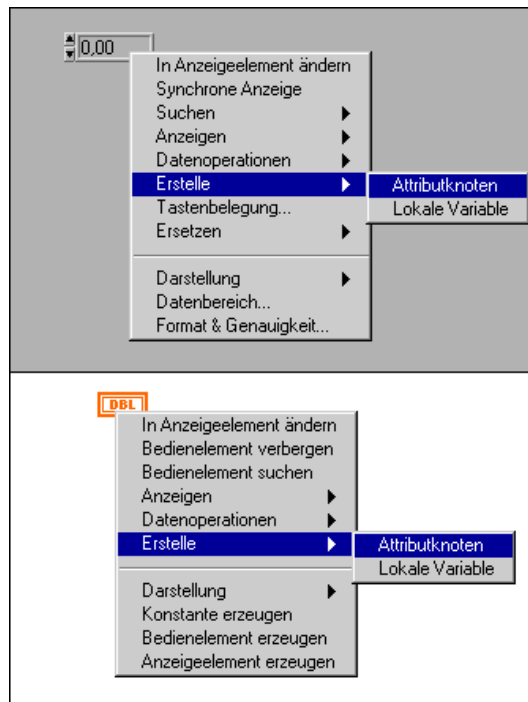
Sobald Sie **VI Pfad** und **Streng typisierte VIs** mit der Funktion “Öffnen der VI Referenz” verbinden, können Sie die Funktion “Öffnen der VI Referenz” mit dem Knoten “Aufruf bei Referenz” verbinden. Das Anschlußfeld wird automatisch im Knoten “Aufruf bei Referenz” angezeigt. Sie können dann beliebige Eingangswerte und Ausgangsanzeigen mit dem Anschlußfeld verbinden. Der Knoten “Aufruf bei Referenz” ruft das vom VI-Referenzeingang angegebene VI, wobei alle angegebenen Eingangswerte verwendet werden. Dieser Knoten ist nützlich, wenn Sie nicht alle SubVIs auf einmal in den Arbeitsspeicher laden wollen.

Attributknoten

Dieses Kapitel beschreibt die Verwendung von Attributknoten, um Attribute von Frontpanel-Bedienelementen programmatisch zu setzen und zu lesen. Einige der nützlichen Attribute sind zum Beispiel Anzeigefarben, Sichtbarkeit von Bedienelementen, Menüstrings für ein Ring-Bedienelement, Plotfarben für Graphen und Diagramme und Cursor für Graphen.

Erstellen von Attributknoten

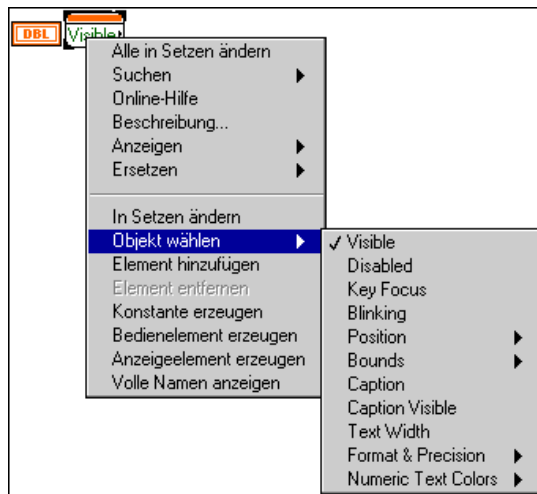
Sie erstellen einen Attributknoten durch Auswahl des Menüpunkts **Erstellen»Attributknoten** aus dem Popup-Menü eines Frontpanel-Bedienelements oder des Terminals.



Durch Auswählen dieses Menüpunkts wird im Blockdiagramm nahe dem Terminal eines Bedienelements ein neuer Knoten erstellt (siehe nachstehende Abbildung). Ist mit dem Bedienelement ein Label assoziiert, wird dieses Label als Ausgangslabel des Attributknotens verwendet. Sie können das Label nach der Erstellung des Knotens ändern.

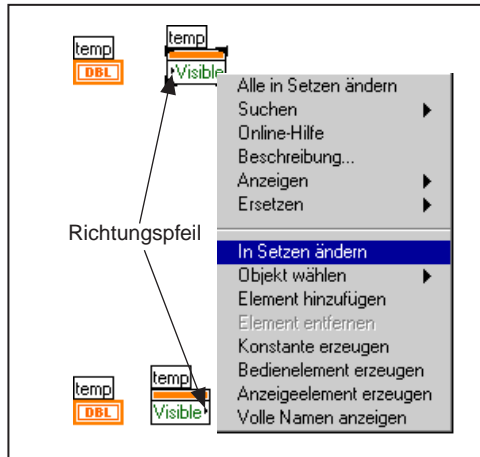


Wenn Sie das Popup-Menü eines Attributterminals aufrufen und anschließend **Objekt wählen** auswählen, sehen Sie ein Menü mit Attributen, die Sie vom Bedienelement aus schreiben oder lesen können (siehe nachstehende Abbildung). Sie können ein Shortcut zur Attributliste verwenden, indem Sie mit dem Bedienwerkzeug auf den Attributknoten klicken.

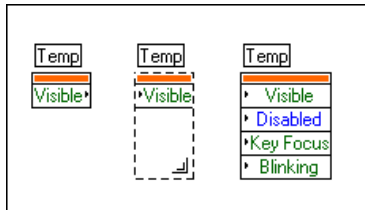


Sie haben die Wahl, Attribute zu lesen oder zu schreiben, indem Sie aus dem Popup-Menü des Attributknotens entweder den Menüpunkt **In Lesen ändern** oder **In Setzen ändern** wählen (siehe nachfolgende Abbildung).

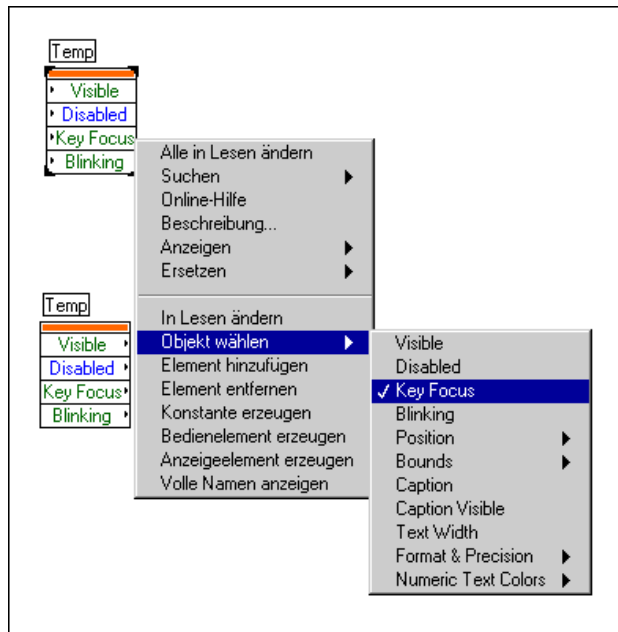
Sie können ein Attribut schreiben (setzen), wenn sich der kleine Richtungspfeil auf der linken Seite des Terminals befindet. Befindet sich der Pfeil auf der rechten Seite des Terminals, können Sie ein Attribut lesen.



Sie können mehrere Attribute mit demselben Knoten lesen oder schreiben, indem Sie den Attributknoten vergrößern. Neue Terminals werden mit zunehmender Größe des Terminals hinzugefügt. Die Ausführungsreihenfolge eines Attributknotens verläuft von oben nach unten.

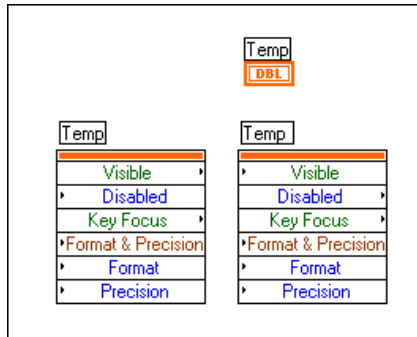


Sie ordnen ein Terminal einem bestimmten Attribut zu, indem Sie mit dem Bedienwerkzeug auf das Terminal klicken und ein Attribut aus dem Pop-up-Menü des Attributknotens auswählen.

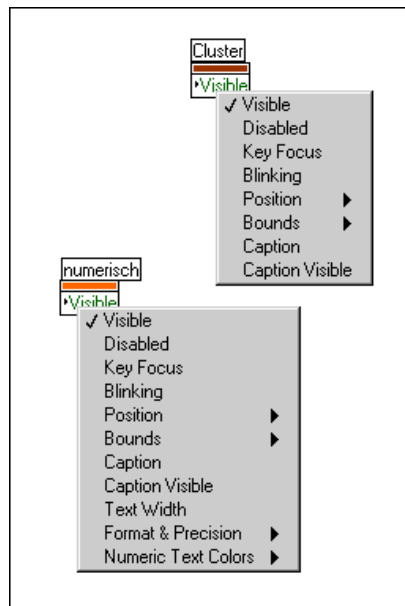


Sie können mehrere Attributknoten durch Klonen eines bestehenden Knotens erstellen, oder indem Sie den Menüpunkt **Erstellen» Attributknoten** erneut auswählen. Um einen bestehenden Knoten zu klonen, klicken Sie auf den betreffenden Knoten, und ziehen Sie ihn mit dem Positionierwerkzeug bei gedrückter <Strg>-Taste (**Windows**), <Optionstaste> (**Macintosh**), <Meta>-Taste (**Sun**) oder <Alt>-Taste (**HP-UX**). Wenn Sie einen Attributknoten jedoch mit den Befehlen aus dem Menü **Bearbeiten** kopieren und einfügen, wird eine neue Kopie sowohl des Attributknotens als auch des dazugehörigen Bedienelements erstellt.

Jede Kopie eines Attributknotens kann unabhängig voneinander Attributterminals zum Lesen oder Schreiben haben.

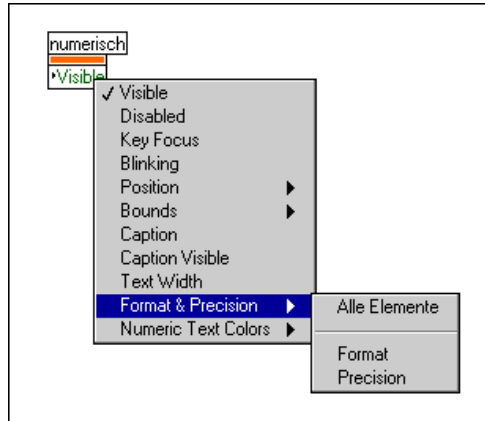


Die folgende Abbildung zeigt die Attribute für einen Cluster und ein numerisches Bedienelement innerhalb des Clusters.



Manche Bedienelemente, wie z.B. der Graph, verfügen über eine große Anzahl von Attributen, die Sie lesen oder setzen können. Manche dieser Attribute sind in Kategorien gruppiert und in Untermenüs aufgelistet, wie z.B. die Kategorie **Format & Genauigkeit** für ein numerisches Bedienelement. Sie haben die Wahl, alle Attribute auf einmal zu setzen, indem Sie den Menüpunkt **Alle Elemente** aus dem Untermenü wählen. Sie

können auch ein oder mehrere Elemente einzeln setzen, indem Sie die betreffenden Attribute auswählen. Der Menüpunkt **Format & Genauigkeit** eines numerischen Bedienelements wird in der nachstehenden Abbildung als Beispiel gezeigt.



Nachdem Sie einen Attributknoten erstellt haben, ändern sich die Menüpunkte **Bedienelement suchen** bzw. **Terminal suchen** des jeweiligen Popup-Menüs in Untermenüs, die Ihnen dabei helfen, Attributknoten zu suchen. Der Attributknoten besitzt analog dazu Optionen, mit denen das mit diesem verbundene Bedienelement und Terminal gesucht werden kann.

Ein Anwendungsbeispiel eines Attributknotens finden Sie in `examples\general\attribute.llb`.

Hilfe zu Attributknoten

Das Hilfefenster und die *Online-Referenz* (Menü **Hilfe**) sind unschätzbare Werkzeuge bei der Verwendung von Attributknoten. Sie können damit Beschreibungen, Datentypen und gültige Werte für Attribute suchen. Weitere Informationen hierzu finden Sie im Abschnitt *Hilfe zu Attributen* in Kapitel 1, *Einführung in die Programmierung in G*.

Grundattribute

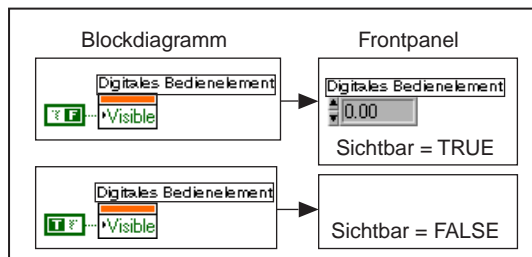
Es gibt viele Attribute, die zum Ändern der verschiedenen Frontpanel-Objekte in Ihrer Anwendung Verfügung stehen. Dieser Abschnitt beschäftigt sich mit den Attributen Sichtbar, Deaktiviert, Fokus, Blinken, Position und Grenzen, die nahezu allen Frontpanel-Objekten gemein sind.



Das Attribut “Visible” (Sichtbar)

Sie können die Sichtbarkeit eines Objekts auf dem Frontpanel mit dem Attribut “Visible” lesen oder schreiben. Das damit verbundene Objekt ist sichtbar, wenn das Attribut den Wert TRUE hat, und verborgen, wenn es den Wert FALSE hat.

In der nachstehenden Abbildung ist das digitale Bedienelement auf einen unsichtbaren Zustand eingestellt. Ein boolescher Wert TRUE macht das Bedienelement, wie gezeigt, wieder sichtbar.

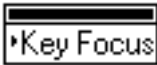
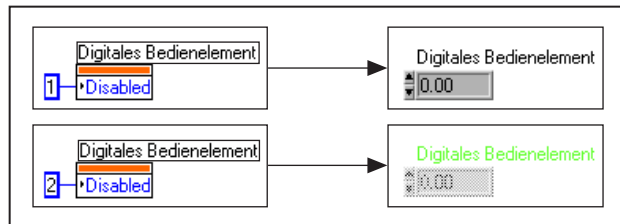


Das Attribut “Disabled” (Deaktiviert)

Durch die Implementierung des Attributs “Disabled” können Sie steuern, ob ein Benutzer Zugriff auf ein Objekt hat. Mit dem Wert Null wird ein Objekt aktiviert, so daß der Benutzer es ausführen kann. Der Wert 1 deaktiviert das Objekt, wodurch die Ausführung verhindert wird. Ein Wert 2 deaktiviert das Objekt und markiert es als nicht bedienbar.

Sie können den Benutzerzugriff auf ein digitales Bedienelement deaktivieren. Das Aussehen des Bedienelements ändert sich dadurch, daß es deaktiviert wird, im ersten Beispiel nicht. Im zweiten Beispiel wird der

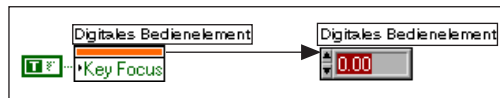
Benutzerzugriff auf das digitale Bedienelement deaktiviert und dieses als nicht bedienbar gekennzeichnet.



Das Attribut “Key Focus” (Fokus)

Mit dem Attribut “Key Focus” können Sie einem Bedienelement den Fokus zuweisen oder überprüfen, ob es gegenwärtig den Fokus hat. Ein Bedienelement mit Fokus verhält sich so, als hätten Sie mit der Tabulatortaste zu diesem Bedienelement gewechselt und dieses damit aktiviert. Bei den meisten Bedienelementen können Sie Werte in das Bedienelement eingeben, indem Sie diese mit der Tastatur eintippen. Sie können den Fokus auf dem Frontpanel auch dadurch setzen, daß Sie im Ausführungsmodus die <Tab>-Taste drücken oder die mit dem Bedienelement verknüpfte (mit dem Menüpunkt **Tastenbelegung** zugewiesene) Zugriffstaste drücken.

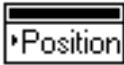
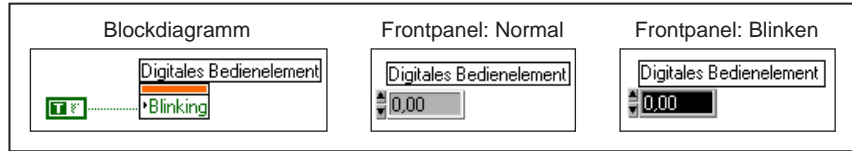
Sie können den Fokus einem digitalen Bedienelement zuweisen. Sie können dann einen neuen Wert in das Bedienelement eingeben, ohne daß Sie dieses mit der Maus auswählen müssen.



Das Attribut “Blinking” (Blinken)

Mit dem Attribut “Blinking” können Sie den Blinkstatus eines Objekts lesen oder setzen. Wenn Sie das Attribut auf TRUE setzen, blinkt ein Frontpanel-Objekt. Blinkrate und Farben werden im Dialogfeld **Voreinstellungen** eingestellt. Wenn Sie dieses Attribut auf FALSE setzen, blinkt das Objekt anschließend nicht mehr.

In der nachfolgenden Abbildung ist das Anzeigeelement auf dem Frontpanel auf Blinken eingestellt.

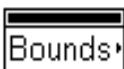
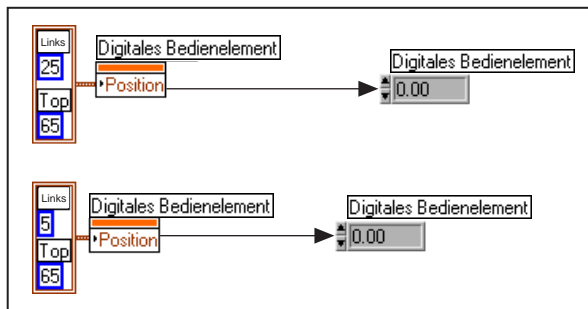


Das Attribut "Position"

Sie können die Position der linken oberen Ecke eines Objekts auf dem Frontpanel mit dem Attribut "Position" setzen oder lesen. Die Position wird in Pixeleinheiten in Bezug auf den Ursprung des Panelfensters bestimmt, bei dem es sich anfangs um die linke obere Ecke des Fensters handelt. Wenn Sie das Fenster gerollt haben, kann sich der Ursprung an einer anderen Stelle befinden. Dieses Attribut besteht aus einem Cluster aus zwei vorzeichenlosen Long-Integer-Werten. Das erste Element im Cluster (links) ist die Position der linken Kante des Bedienelements, und das zweite Element im Cluster (oben) ist die Position der oberen Kante des Bedienelements in Bezug auf den Ursprung des Panelfensters.



Durch die Ausführung oder Verwendung des Attributnotens "Position" ändert sich die Position des digitalen Bedienelements auf dem Frontpanel in der nachstehenden Abbildung.



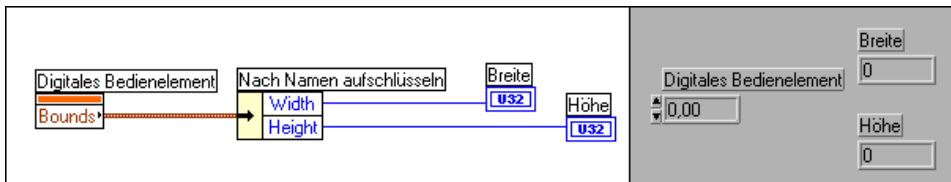
Das Attribut "Bounds" (Grenzen) (schreibgeschützt)

Das Attribut "Bounds" liest die Grenze eines Objekts auf dem Frontpanel in Pixeleinheiten. Der Wert umfaßt das Bedienelement und alle seine Teile, einschließlich Label, Legende, Maßstab usw. Das Attribut besteht aus einem Cluster aus zwei vorzeichenlosen Long-Integer-Werten. Das erste Element im Cluster (Breite) ist die Breite des Objekts in Pixeln, und das zweite Element im Cluster (Höhe) ist die Höhe des Objekts in Pixeln.



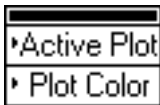
Das Attribut “Bounds” ist *schreibgeschützt*. Es kann die Größe eines Bedien- oder Anzeigeelements auf dem Frontpanel *nicht* verändern. Die meisten Objekte verfügen über andere Attribute zur Größenanpassung, wie z.B. das Attribut “Plot Area Size” (Zeichenbereich) von Graphen und Diagrammen. Dieses Attribut ist nützlich, wenn Sie die Gesamtgröße eines Bedienelements einschließlich aller optionaler Bestandteile lesen müssen, damit Sie andere Bedienelemente im Verhältnis zu diesem betreffenden Bedienelement positionieren können.

Sie können die Grenzen des digitalen Bedienelements feststellen (siehe nachstehende Abbildung).



Beispiele für Sonderattribute von Bedien- oder Anzeigeelementen

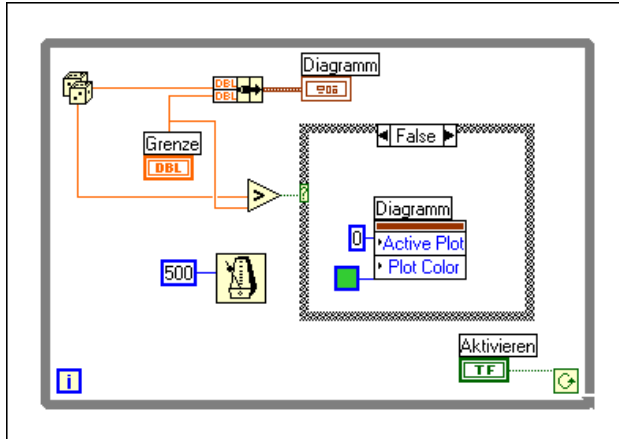
In den folgenden Abschnitten erhalten Sie einen Einblick in übliche Anwendungen von Attributknoten. Im Verzeichnis `general` finden Sie weitere Anwendungen des Attributknotens.



Ändern der Plotfarbe eines Diagramms

Die links gezeigten Attribute setzen oder lesen den aktiven Plot (die Hauptdiagonale, für die spezifische Attribute gesetzt oder gelesen werden) und die Plotfarbe des aktiven Plots. “Active Plot” ist ein Integer-Wert, der angibt, welcher Plot in einem Diagramm mit mehreren Plots der aktive sein soll. Plotfarbe ist ein Integer-Wert, der die gewünschte Farbe repräsentiert. Sie können auf die Plotfarbe zugreifen, indem Sie **Plot Info»Plot Color** aus der Attributliste wählen.

Dieser Attributknoten ändert die Plotfarbe des mit dem Attribut “Aktiver Plot” angegebenen Plots. Im folgenden Beispiel ändert sich die Farbe des Plots für eine Zufallszahl, wenn die Werte eine vom Benutzer gesetzte Grenze überschreiten. Beachten Sie, daß der aktive Plot angegeben wird, bevor die Plotfarbe geändert wird.



Einstellen der Strings eines booleschen Attributs

Dieses boolesche Attribut setzt oder liest die Label eines booleschen Bedienelements. Der Eingang ist ein Array aus bis zu vier Strings, die den Zuständen False, True, True Tracking und False Tracking entsprechen.

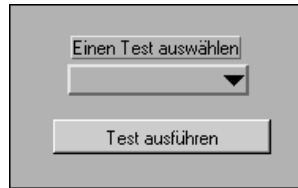
- True und False: Zustände EIN und AUS des booleschen Elements.
- True und False Tracking: Vorläufige Übergangsebenen zwischen booleschen Zuständen. True Tracking ist der Übergangszustand, wenn das boolesche Element von True in False übergeht. False Tracking ist der Übergang von False zu True, wenn das boolesche Attribut geändert wird. Tracking gilt nur für boolesche Elemente bei den mechanischen Aktionen "Schaltet wenn losgelassen" und "Latch wenn losgelassen". Diese mechanischen Aktionen befinden sich in einem Übergangszustand, bis Sie die Maustaste loslassen. Die Textstrings "True Tracking" und "False Tracking" werden während des Übergangszustandes angezeigt.

Sie können die Anzeigestrings für boolesche Bedienelemente auf die Stringoptionen **Stopp**, **Ausführen**, **Stopp?** und **Ausführen?** setzen.

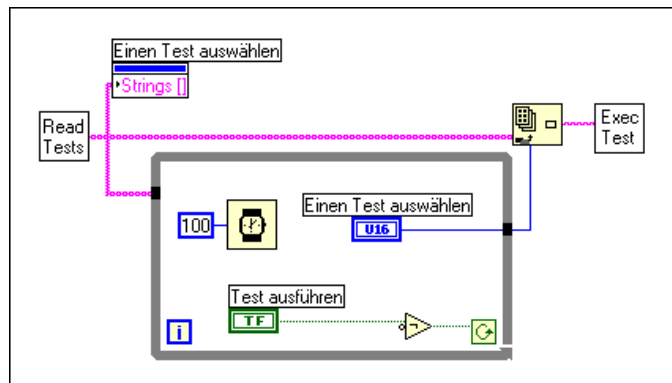
Einstellen der Strings für Ring-Steuerungen

Die Ring-Steuerung ist eine Steuerung eines Popup-Menüs, die den numerischen Wert des gegenwärtig gewählten Elements enthält. Sie können sie dazu verwenden, dem Benutzer eine Liste von Optionen anzubieten. Wenn die Optionen erst bei Run-Time bestimmt werden können, lassen sich die Optionen mit dem Attributknoten setzen. Weitere Informationen über Ring-Steuerungen finden Sie in Kapitel 13, *Bedien- und Anzeigeelemente vom Typ Liste und Ring*.

Im folgenden Beispiel wird dem Benutzer ein Panel mit einem Ring-Bedienelement angeboten, das eine Liste von Tests enthält. Der Benutzer wählt einen Text und klickt zum Fortfahren auf die Taste **Test ausführen**.



Das in der nachstehenden Abbildung gezeigte Blockdiagramm liest eine Liste von gültigen Tests aus einer Datei und übergibt die Liste, die als Array aus Strings dargestellt wird, an einen Attributknoten für das Ring-Bedienelement. Das Diagramm wird dann immer wieder durchlaufen und wartet darauf, daß der Benutzer auf die Taste **Test ausführen** klickt. Dies gibt dem Benutzer die Möglichkeit, einen Test aus dem Ring-Bedienelement zu wählen oder Informationen für andere Bedienelemente anzugeben. Wenn der Benutzer einen Test wählt, wird der String, der dem numerischen Wert des Ring-Bedienelements entspricht, gelesen und an das VI übergeben, das dann den Test ausführt.

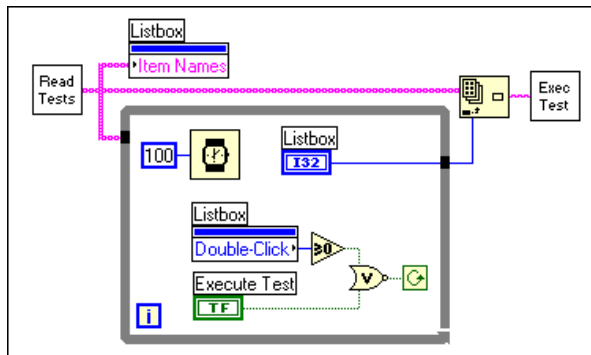


Verwendung eines Doppelklick-Listenelementes

Das Attribut “Doppelklick” ist ein schreibgeschütztes Attribut speziell für Listenfelder. Dieses Attribut zeigt an, in welches Listenfeld auf dem Frontpanel doppelgeklickt worden ist. Der Wert des Attributs “Doppelklick” wird gesetzt, wenn Sie auf ein Element doppelklicken oder nach der Auswahl eines Elements <Eingabe> (**Windows** oder **UNIX**) oder <Return> (**Macintosh** und **Sun**) drücken. Der Doppelklickwert ist –1, wenn kein Doppelklick erfolgt ist. Er wird auf –1 zurückgesetzt, nachdem er mit

dem Attributknoten gelesen worden ist. Er wird auch dann auf –1 zurückgesetzt, wenn Sie ein anderes Element auswählen oder eines der anderen Attribute setzen.

Die folgende Abbildung zeigt ein Beispiel für die Verwendung des Attributknotens “Doppelklick”, um zu bestimmen, welcher Test auszuführen ist.



Die Schleife wird angehalten, wenn Sie auf die Taste **Test ausführen** klicken oder wenn Sie auf ein Element doppelklicken, wobei in diesem Fall das Attribut “Doppelklick” statt –1 die Elementnummer zurückgibt.

Selektive Optionsauswahl für Benutzer

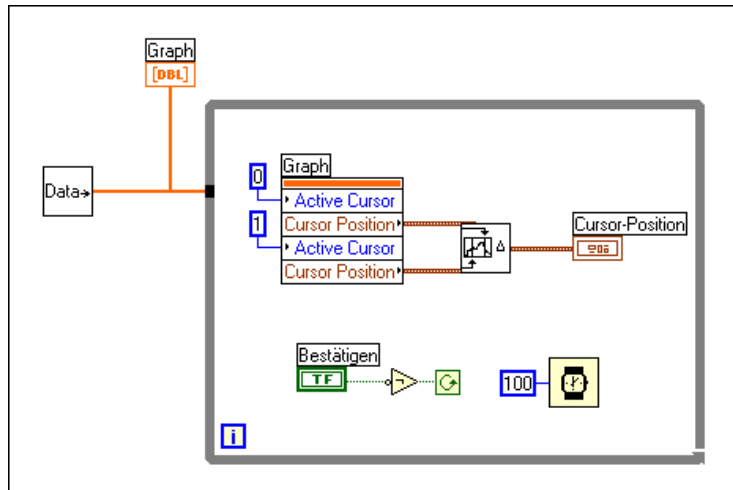
Wenn Benutzer eine Auswahl treffen, wollen Sie womöglich verschiedene Optionen anbieten. Hierfür gibt es drei Möglichkeiten.

1. Eine Option besteht darin, SubVIs für Popup-Menüs zu verwenden. Sie können SubVIs mit den Optionen erstellen, die Sie dem Benutzer zur Auswahl anbieten wollen. Durch die Verwendung der Menüpunkte **Frontpanel nach dem Aufruf anzeigen** und **Nach Abarbeitung schließen** aus **VI-Einstellungen** » **Ausführungsoptionen** bei der Erstellung des SubVIs kann eines dieser SubVIs beim Aufruf geöffnet sein.
2. Eine andere Methode, Optionen anzubieten, besteht darin, die Option `visible` des Attributknotens zu verwenden, um Bedienelemente selektiv anzuzeigen und zu verbergen.
3. Die dritte Methode zur Darstellung von Optionen bietet sich mit der Option `disabled` des Attributknotens, mit der Bedienelemente selektiv deaktiviert werden können. Wenn ein Bedienelement deaktiviert ist, kann der Benutzer den Wert nicht ändern.

Programmatisches Lesen der Cursor von Graphen

Sie können Attribute verwenden, um auf Informationen in einem Plot eines Graphen mit mehreren Plots oder einem Thumb eines Schiebereglers mit mehreren Thumbs zuzugreifen, aber Sie müssen angeben, welches Objekt ausgeführt wird. Die mehrfachen Cursor in einem Graphen sind ein gutes Beispiel für ein Attribut, das aktiviert werden muß, bevor vom Diagramm aus darauf zugegriffen werden kann.

Das folgende Blockdiagramm zeigt ein VI, das Daten in einem Graphen anzeigt und die Position der Cursor des Graphen programmatisch liest.



Beim Durchlaufen der While-Schleife aktiviert das VI zuerst den Cursor “Kleinsten Wert” und liest seinen numerischen Wert. Als nächstes aktiviert und liest das VI den Cursor “Größter Wert”. Dann berechnet das VI Informationen über die Cursorauswahl, und zeigt sie auf dem Frontpanel an. Wenn Sie die Bestätigungstaste drücken, beendet das VI die Schleife und bestätigt die Cursorpositionen.

Anwendungsbeispiele für das programmatische Lesen der Cursor von Graphen finden Sie in `examples\general\graphs\zoom.11b`.

Globale und lokale Variablen

Dieses Kapitel beschreibt, wie globale und lokale Variablen definiert und verwendet werden. Verwenden Sie globale Variablen, um von mehreren VIs aus leicht auf einen bestimmten Satz von Werten zuzugreifen. Lokale Variablen dienen innerhalb eines einzigen VIs einem ähnlichen Zweck.

Globale und lokale Variablen sind fortgeschrittene G-Konzepte. Sie sollten sich vor der Anwendung dieser Elemente unbedingt mit diesem Kapitel vertraut machen.

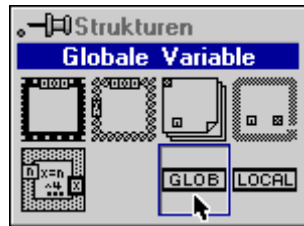
Beispiele zur Verwendung von globalen und lokalen Variablen finden Sie in `examples\general\globals.llb` und `examples\general\locals.llb`.

Globale Variablen

Eine globale Variable ist ein integriertes G-Objekt. Wenn Sie eine globale Variable erstellen, wird automatisch eine besondere Art von VI erstellt. Sie können diesem VI Frontpanel-Bedienelemente hinzufügen, die die Datentypen der darin enthaltenen globalen Variablen definieren.

Es gibt zwei Möglichkeiten, um mehrere globale Variablen zu erstellen. Sie können mehrere VIs mit je einem Element erstellen, oder ein VI mit mehreren globalen Variablen, indem Sie mehrere Frontpanel-Bedienelemente auf dem Frontpanel der einen globalen Variable definieren. Der Ansatz eines VIs mit mehreren globalen Variablen ist effizienter, da Sie verwandte Variablen gruppieren.

Sie können eine globale Variable erstellen, indem Sie die globale Variable aus der Palette **Funktionen»Strukturen** wählen und in das Diagramm einfügen.



Globaler Knoten

Im Blockdiagramm erscheint ein Knoten für die globale Variable.

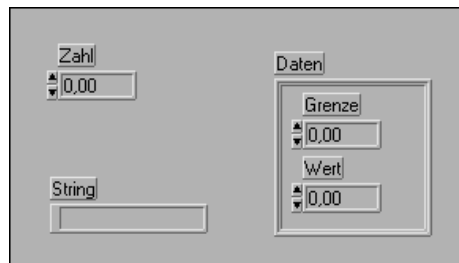
Doppelklicken Sie auf den Knoten, um dessen Frontpanel zu öffnen. Mit diesem Frontpanel definieren Sie die Datentypen für eine oder mehrere globale Variablen.



Hinweis

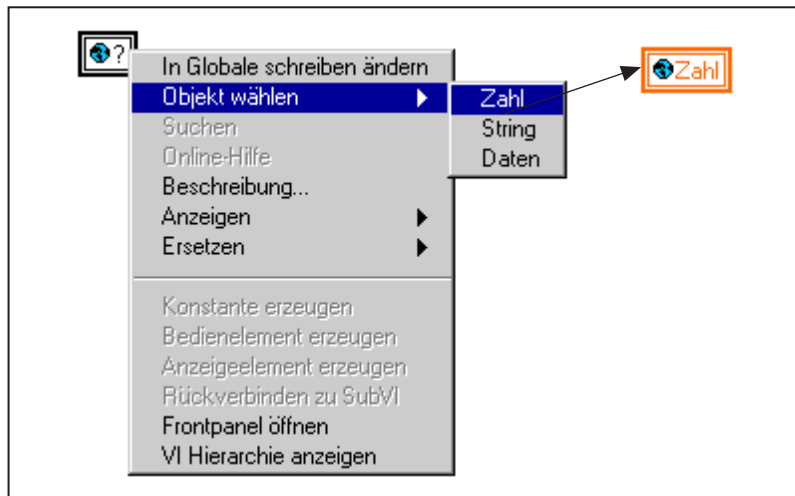
Weisen Sie jedem Bedienelement einen Namen (Label) zu, da Sie mit dem Namen auf eine bestimmte globale Variable Bezug nehmen müssen. Sobald Sie die Datentypen für die globalen Variablen definiert haben, speichern Sie das globale VI.

Die folgende Abbildung zeigt ein Beispiel eines Frontpanels, das drei globale Variablen beschreibt — eine Zahl, einen String und einen aus zwei Werten bestehenden Cluster.



Sobald Sie eine globale Variable in ein Blockdiagramm plaziert und ein Frontpanel dafür definiert haben, ist der Knoten mit dem Frontpanel verknüpft. Da ein einziges VI mit globalen Variablen mehrere globale Variablen definieren kann, müssen Sie wählen, auf welche globale Variable Sie von einem bestimmten Knoten aus zugreifen wollen. Wählen Sie eine globale Variable, indem Sie das Popup-Menü des Knotens aufrufen und das Element aus dem Menü **Objekt wählen** wählen (siehe nachstehende

Abbildung). Oder klicken Sie mit dem Bedienwerkzeug auf den Knoten, und wählen Sie das gewünschte Element.



Sie können in eine globale Variable schreiben oder daraus lesen. Mit 'in eine globale Variable schreiben' ist gemeint, daß sich der Wert der globalen Variable ändert; mit 'aus einer globalen Variable lesen' ist gemeint, daß Sie auf die globale Variable als Datenquelle zugreifen. Wenn Sie in eine globale Variable schreiben oder diese lesen wollen, wählen Sie den Menüpunkt **In Globale schreiben ändern** bzw. **In Globale lesen ändern** aus dem Popup-Menü der globalen Variable.

Globale Variablen können von einem beliebigen im Arbeitsspeicher befindlichen VI geschrieben oder gelesen werden. Es ist wichtig zu wissen, an welchen Stellen in Ihrer Anwendung gelesen und geschrieben wird, damit eine globale Variable nicht unerwartet geändert wird. Bei G-Programmen können viele Vorgänge gleichzeitig ablaufen, und es ist möglicherweise nicht immer leicht zu wissen, wann von verschiedenen parallelen Diagrammen auf eine globale Variable zugegriffen wird. Dieses Problem mit wetteifernden Zugriffen auf eine freigegebene Ressource wie eine globale Variable tritt bei Multithreaded-Software sogar noch stärker hervor. Es gibt viele legitime Verwendungen für globale Variablen, aber der unüberlegte Gebrauch kann manchmal zu Situationen führen, in denen Debugging schwer ist. Verwenden Sie sie daher umsichtig.

Nachdem Sie ein VI mit globalen Variablen gespeichert haben, können Sie die globalen Variablen mit **Funktionen»VI auswählen...** in andere VIs plazieren. Wenn Sie ein VI mit globalen Variablen aus dem Dateidialogfeld wählen, plaziert G den Knoten der globalen Variablen in das Diagramm.

Sie können eine globale Variable auch vom Hierarchiefenster aus klonen, kopieren und einfügen oder durch Ziehen kopieren.

Lokale Variablen

Mit einer lokalen Variable können Sie eines der Bedien- oder Anzeigeelemente auf dem Frontpanel des VIs schreiben oder lesen. Das Schreiben in eine lokale Variable hat dieselbe Wirkung wie die Übergabe von Daten an ein Terminal, mit der Ausnahme, daß Sie die Variable schreiben bzw. lesen können, obwohl es sich dabei um ein Bedienelement bzw. Anzeigeelement handelt. Außerdem ist eine beliebige Anzahl von lokalen Variablenreferenzen zu einem bestimmten Frontpanel-Bedienelement möglich, von denen sich einige im Schreibmodus und andere im Lesemodus befinden können.

Mit einer lokalen Variablenreferenz kann ein Frontpanel-Bedienelement sogar tatsächlich sowohl als Eingang als auch als Ausgang verwendet werden.

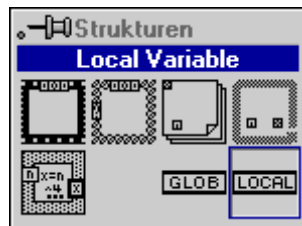


Hinweis

Sie müssen sicherstellen, daß das Label, das Sie für Ihre lokale Variable wählen, mit einem Objekt des Frontpanels verbunden ist. Wenn die lokale Variable nicht mit einem Objekt auf dem Frontpanel verbunden ist, funktioniert es nicht.

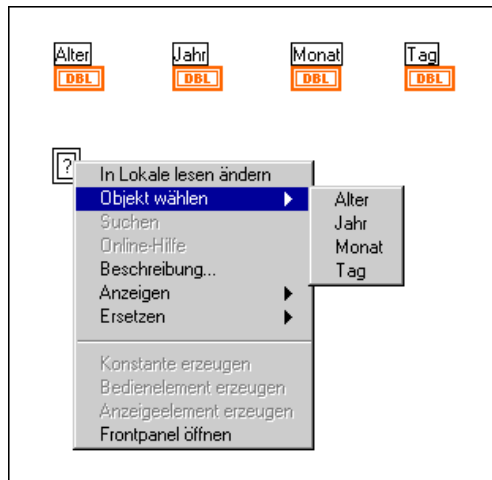


Die leichteste Methode, eine lokale Variable zu erstellen, besteht darin, das Popup-Menü des Bedienelements auf dem Frontpanel oder des Terminals aufzurufen und **Erstelle»Lokale Variable** zu wählen. Dabei erscheint automatisch eine lokale Variable im Blockdiagramm. Eine andere Möglichkeit besteht darin, die lokale Variable unter **Funktionen»Strukturen** auszuwählen (siehe nachstehende Abbildung).

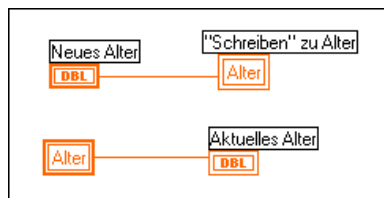


Ein Knoten, der ähnlich wie der einer globalen Variablen aussieht, wird angezeigt. Sie können das Popup-Menü des Knotens aufrufen oder mit dem Bedienwerkzeug darauf klicken, um das Bedienelement, das Sie lesen oder setzen wollen, aus einer Liste von Bedienelementen der höchsten Ebene zu wählen (siehe nachfolgende Abbildung). Sie können auch festlegen, ob Sie das Bedienelement schreiben oder von ihm lesen wollen, indem Sie

entweder den Menüpunkt **In Lokale schreiben ändern** oder den Menüpunkt **In Lokale lesen ändern** wählen.



Die folgende Abbildung zeigt den Zugriff mehrerer lokaler Variablen auf dasselbe Bedienelement, wobei jede eine andere Aufgabe hat, und zwar entweder Lesen oder Schreiben. Sie können die Variable `Alter` zweimal im Diagramm verwenden, einmal zum Schreiben und ein zweites Mal zum Lesen.



Vergessen Sie nicht, eine Sequenz für den Zugriff auf lokale Variablen oder globale Variablen festzulegen, um wie im vorstehenden Beispiel die gewünschten Ergebnisse zu erzielen. Wenn Sie die Sequenz nicht selbst festlegen, gibt es keine Garantie dafür, daß "Schreiben" in `Alter` vor "Lesen" von `Alter` stattfindet.

Fortgeschrittene G-Themen

Dieser Abschnitt enthält Informationen über fortgeschrittene Funktionen in G und Techniken, die zur Erstellung virtueller Instrumente verwendet werden können.

Teil IV, *Fortgeschrittene G-Themen*, enthält folgende Kapitel.

- Kapitel 24, *Benutzerspezifische Bedienelemente und Typendefinitionen*, bietet eine Einführung in benutzerspezifische Bedienelemente und Typendefinitionen.
- Kapitel 25, *Aufrufen von Code aus anderen Sprachen*, beschreibt verschiedene Methoden zum Aufrufen von Code, der in anderen Sprachen geschrieben wurde.
- Kapitel 26, *Überblick über das G-Ausführungssystem*, beschreibt das VI-Multitasking und die Ausführung von VIs.
- Kapitel 27, *Verwalten Ihrer Anwendungen*, beschreibt, wie die Dateien in Ihren G-Anwendungen verwaltet werden können.
- Kapitel 28, *Leistung*, ist in drei Abschnitte unterteilt. Der erste Abschnitt beschreibt den Performance Profiler, eine Funktion, mit der Daten zur Ausführungszeit Ihres VI angezeigt und Singlethread-, Multithread- und Mehrprozessor-Anwendungen überwacht werden können. Im zweiten Abschnitt werden die Faktoren beschrieben, die sich auf die Ausführungszeit auswirken. Im dritten Abschnitt werden Faktoren beschrieben, die sich auf die Speicherauslastung auswirken.
- Kapitel 29, *Portierbarkeit und Lokalisierungsthemen*, behandelt Themen im Zusammenhang mit der Übertragung von VIs zwischen Plattformen und die VI-Lokalisierung.

Benutzerspezifische Bedienelemente und Typendefinitionen

Dieses Kapitel stellt benutzerspezifische Bedienelemente und Typendefinitionen vor.

Sie können ein Bedien- oder Anzeigeelement auf einem Frontpanel anpassen, um es besser auf Ihre Anwendung abzustimmen. Nehmen wir z.B. an, Sie wollen einen booleschen Schalter erzeugen, der ausgeschaltet ein geschlossenes Ventil und eingeschaltet ein offenes Ventil anzeigt, ferner einen Schieberegler mit der Skala auf der rechten anstatt auf der linken Seite, oder ein Ring-Bedienelement mit vordefinierten Text- oder Bildelementen.

Sie können ein Bedien- oder Anzeigeelement, das Sie angepaßt haben, wie VIs in einem Verzeichnis oder einer VI-Bibliothek speichern. Dann können Sie dieses Bedienelement auf anderen Frontpanels verwenden. Sie können ferner ein Icon für Ihr benutzerspezifisches Bedienelement erstellen und den Namen und das Icon des Bedienelements der Palette **Elemente** hinzufügen.

Benötigen Sie dasselbe Bedienelement an vielen Stellen in Ihren VIs, können Sie einen Prototypen dieses Bedienelements, eine sogenannte Typendefinition, erstellen. Wenn Sie an der Typendefinition eine Änderung vornehmen, aktualisieren Sie automatisch auch alle VIs, die sie verwenden.

Sie können auch ein benutzerspezifisches Bedienelement in einem Blockdiagramm verwenden. Dadurch wird eine Konstante mit demselben Datentyp wie das Bedienelement erstellt. Wenn Sie eine Typendefinition in einem Blockdiagramm verwenden, wird die resultierende Konstante automatisch aktualisiert, wenn Sie an der Typendefinition eine Änderung vornehmen.

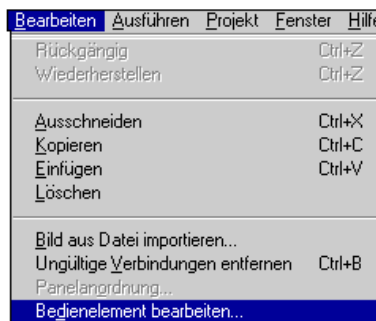
Die folgenden Abschnitte erläutern, wie diese und andere benutzerspezifische Veränderungen an Bedienelementen vorgenommen werden.

Benutzerspezifische Bedienelemente

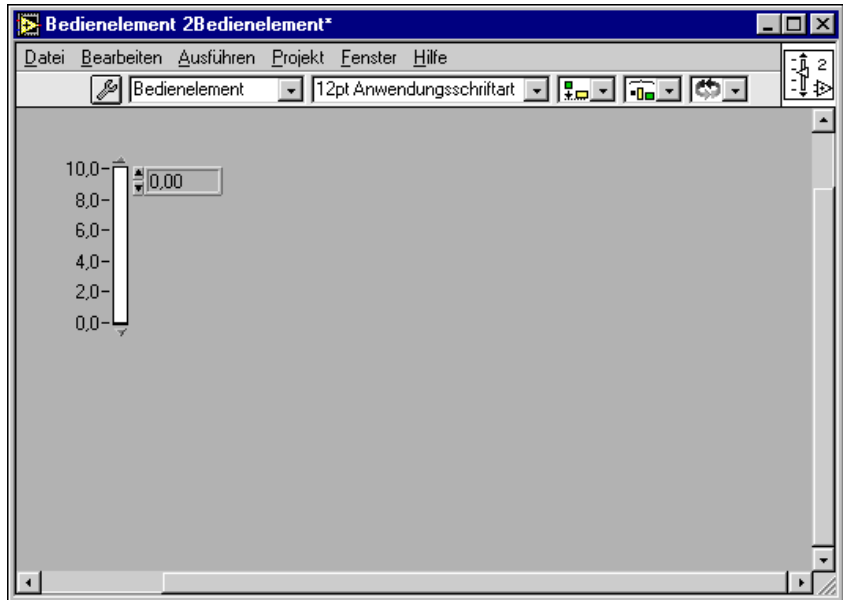
Verwendung des Bedienelement-Editors

Vergewissern Sie sich, daß Sie sich zum Anpassen eines Bedienelements im Bearbeitungsmodus befinden. Plazieren Sie ein Bedienelement, das dem zu erstellenden am nächsten kommt, auf das Frontpanel. Um beispielsweise einen Schieberegler mit der Skala auf der rechten Seite zu erstellen, beginnen Sie damit, einen beliebigen vertikalen Schieberegler auf das Frontpanel zu plazieren.

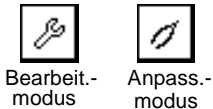
Wählen Sie den Schieberegler mit dem Positionierwerkzeug aus, und wählen Sie dann **Bearbeiten»Bedienelement bearbeiten...**. Diese Option ist nur verfügbar, wenn Sie ein Bedienelement auswählen. Sie können immer nur ein Bedienelement eines Frontpanels auf einmal bearbeiten.



Ein Fenster wird geöffnet, das eine Kopie des Bedienelements anzeigt. Dieses Fenster, das in der nachfolgenden Abbildung zu sehen ist, wird als Bedienelement-Editor bezeichnet und enthält den Titel Bedienelement *N*; dieser Name bleibt dem Fenster des Bedienelement-Editors solange zugewiesen, bis Sie das Bedienelement speichern und ihm damit einen permanenten Namen zuweisen.



Das Fenster des Bedienelement-Editors sieht wie ein Frontpanel aus, es wird aber nur zum Bearbeiten und Speichern eines einzigen Bedienelements verwendet; es besitzt kein Blockdiagramm und kann nicht ausgeführt werden.



Bearbeit.-
modus

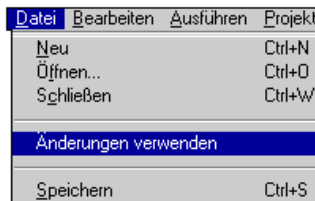
Anpass.-
modus

Der Bedienelement-Editor besitzt einen Bearbeitungsmodus und einen Anpassungsmodus. Der aktuelle Modus wird durch eine Taste in der Symbolleiste angezeigt (siehe Abbildungen links). Der Bedienelement-Editor befindet sich im Bearbeitungsmodus, wenn er geöffnet wird. Im Bearbeitungsmodus können Sie, wie im Bearbeitungsmodus eines beliebigen Frontpanels, die Größe oder Farbe eines Bedienelements ändern und Optionen aus seinem Popup-Menü wählen. Und im Anpassungsmodus können Sie die Bestandteile eines Bedienelements einzeln ändern. Der Anpassungsmodus wird später in diesem Kapitel ausführlich beschrieben.

Sobald Sie ein Bedienelement bearbeitet haben, können Sie es anstelle des ursprünglichen Bedienelements auf dem Frontpanel verwenden, das Sie konstruiert haben, als Sie den Bedienelement-Editor geöffnet haben. Sie können es auch unter anderen Frontpanels speichern.

Verwenden von Änderungen eines benutzerspezifischen Bedienelements

Wenn Sie daran gehen, das ursprüngliche Bedienelement des Frontpanels durch das neue benutzerspezifische Bedienelement zu ersetzen, wählen Sie **Datei»Änderungen verwenden** aus dem Hauptmenü des Bedienelement-Editors.



Wenn das ursprüngliche Frontpanel die einzige Stelle ist, an der Sie das benutzerspezifische Bedienelement verwenden, können Sie das Fenster des Bedienelement-Editors schließen, ohne das Bedienelement zu speichern. Stellen Sie sicher, daß Sie das ursprüngliche VI mit dem benutzerspezifischen Bedienelement gespeichert haben, um Ihre Arbeit zu bewahren. Wenn Sie das benutzerspezifische Bedienelement in Zukunft auf anderen Frontpanels verwenden wollen, müssen Sie es entsprechend der Anleitung im Abschnitt *Speichern benutzerspezifischer Bedienelemente* in diesem Kapitel speichern.

Die Option **Änderungen verwenden** ist erst verfügbar, wenn Sie Änderungen an dem Bedienelement vorgenommen haben. **Änderungen verwenden** ist deaktiviert, wenn es kein ursprüngliches Bedienelement gibt, das zu aktualisieren ist. Dies ist der Fall, wenn Sie das ursprüngliche Bedienelement ersetzen oder löschen, wenn Sie das ursprüngliche Frontpanel schließen oder wenn Sie ein zuvor gespeichertes benutzerspezifisches Bedienelement mit **Datei»Öffnen** öffnen.

Gültige benutzerspezifische Bedienelemente



nicht OK-Taste

Enthält der Bedienelement-Editor mehr als ein Bedienelement, wird die Taste **nicht OK** angezeigt. Bei einem gültigen benutzerspezifischen Bedienelement muß es sich um ein einzelnes Bedienelement handeln, obgleich es später in einen Cluster weiterer Bedienelemente aufgenommen werden kann. Die Taste **nicht OK** erscheint vorübergehend, wenn Sie Bedienelemente in ein oder aus einem Cluster oder Array verschieben. Um eine Erklärung des Fehlers zu erhalten, klicken Sie auf die Taste **nicht OK**.

Befinden sich mehrere Bedienelemente im Bedienelement-Editor, besagt die Fehlermeldung folgendes:

Mögliche Fehler sind unter anderem, daß entweder mehrere Bedienelemente oder keine Bedienelemente gegeben sind.

Speichern benutzerspezifischer Bedienelemente

Wenn Sie Ihr benutzerspezifisches Bedienelement auf anderen Frontpanels verwenden wollen, wählen Sie **Datei»Speichern unter...** aus dem Hauptmenü des Bedienelement-Editors. Sie speichern ein Bedienelement auf dieselbe Weise wie ein VI, in einem Verzeichnis oder in einer VI-Bibliothek. Ein Verzeichnis oder eine VI-Bibliothek enthält Bedienelemente, VIs oder beides.

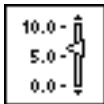
Wenn Sie das Fenster des Bedienelement-Editors schließen, ohne die Änderungen am Bedienelement zu speichern, wird ein Dialogfeld angezeigt, das Sie fragt, ob Sie das Bedienelement speichern wollen.

Verwendung von benutzerspezifischen Bedienelementen

Wenn Sie das benutzerspezifische Bedienelement speichern, können Sie es auf anderen Frontpanels verwenden, indem Sie **Elemente»Ein Bedienelement auswählen...** aus dem Frontpanel eines beliebigen VI wählen. Sie können es auch in Blockdiagrammen verwenden, indem Sie **Funktionen»VI auswählen...** aus dem Blockdiagramm eines beliebigen VIs wählen. Wenn Sie ein benutzerspezifisches Bedienelement in einem Blockdiagramm verwenden, erstellen Sie eine Konstante mit demselben Datentyp wie das benutzerspezifische Bedienelement.

Weitere Informationen über das Hinzufügen eines benutzerspezifischen Bedienelements zur Palette "Elemente" finden Sie im Abschnitt [Anpassen der Elemente- und Funktionenpaletten](#) in Kapitel 7, [Anpassen Ihrer Umgebung](#).

Erstellen von Icons



Bedienelement-Icon

Wenn Sie die Absicht haben, Ihr Bedienelement der Palette **Elemente** hinzuzufügen, oder wenn das Bedienelement eine Typendefinition ist, erstellen Sie vor dem Speichern ein Icon, das das Bedienelement darstellt. Doppelklicken Sie auf das Iconquadrat in der rechten oberen Ecke des Fensters des Bedienelement-Editors, oder rufen Sie sein Popup-Menü auf, und wählen Sie **Icon bearbeiten...**, um ein Icon für das Bedienelement zu erstellen. Dieses Icon repräsentiert das Bedienelement in der Palette **Elemente**, und falls es sich bei dem Bedienelement um eine

Typendefinition handelt, im Hierarchiefenster. Weitere Informationen hierzu finden Sie im Abschnitt *Typendefinitionen* in diesem Kapitel und im Abschnitt *Verwendung des Hierarchiefensters* in Kapitel 3, *Verwendung von SubVIs*.

Unabhängige Instanzen von benutzerspezifischen Bedienelementen

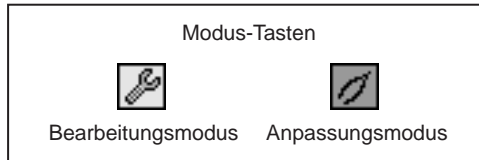
Sie können ein beliebiges zuvor gespeichertes benutzerspezifisches Bedienelement öffnen, indem Sie **Datei»Öffnen** wählen. Ein benutzerspezifisches Bedienelement wird stets im Fenster des Bedienelement-Editors geöffnet.

Änderungen, die Sie an einem benutzerspezifischen Bedienelement vornehmen, wenn Sie dieses öffnen, wirken sich nicht auf das VI aus, das das Bedienelement verwendet. Wenn Sie ein benutzerspezifisches Bedienelement auf einem Frontpanel verwenden, besteht keine Verbindung zwischen dieser Instanz des benutzerspezifischen Bedienelements und der Datei oder VI-Bibliothek, in der es gespeichert ist; jede Instanz ist eine separate, unabhängige Kopie.

Sie können jedoch eine Verbindung zwischen Bedienelement-Instanzen auf verschiedenen Frontpanels oder Blockdiagrammen von VIs und dem Prototypen des Bedienelements erstellen. Hierzu müssen Sie das benutzerspezifische Bedienelement als Typendefinition oder strikte Typendefinition speichern. Dann wirken sich alle Änderungen, die Sie am Prototypen vornehmen, auf alle Instanzen des Bedienelements in allen VIs aus, die es verwenden. Weitere Informationen hierzu finden Sie im Abschnitt *Typendefinitionen* später in diesem Kapitel.

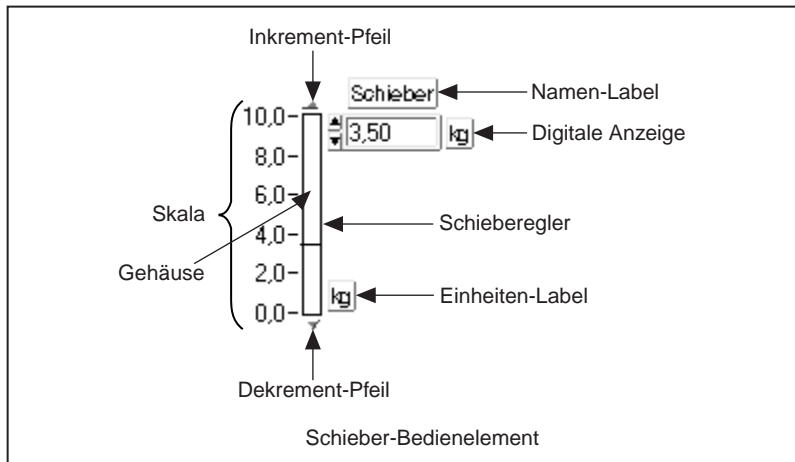
Anpassungsmodus

Sie können im Anpassungsmodus des Bedienelement-Editors umfangreichere Änderungen an einem Bedienelement vornehmen. Sie wechseln zwischen Bearbeitungs- und Anpassungsmodus, indem Sie in der Symbolleiste des Bedienelement-Editors auf die Modustaste klicken oder indem Sie **Anpassungsmodus** oder **Bearbeitungsmodus** aus dem Menü **Ausführen** wählen, während der Bedienelement-Editor das aktive Fenster ist (siehe nachstehende Abbildungen).



Unabhängige Bestandteile

Alle Bedienelemente setzen sich aus kleineren Bestandteilen zusammen. Ein Bedienelement vom Typ "Schieber" besteht beispielsweise aus einer Skala, einem Gehäuse, einem Schieberegler, den Inkrement- und Dekrementpfeilen, einer digitalen Anzeige und einem Namen-Label. Die folgende Abbildung zeigt die Bestandteile eines Schiebers.



Wenn Sie im Bedienelement-Editor zum Anpassungsmodus wechseln, werden die Bestandteile Ihres Bedienelements unabhängig voneinander. Sie können dann an jedem Teil Änderungen vornehmen, ohne daß dadurch

andere Teile beeinträchtigt werden. Wenn Sie zum Beispiel mit dem Positionierwerkzeug auf die Skala des Schiebers klicken und diese ziehen, bewegt sich nur die Skala. Sie können Teile auswählen und mit den Ringen **Objekte ausrichten** bzw. **Objekte einteilen** aus der Symbolleiste ausrichten oder einteilen oder mit dem Ring **Neuordnen** aus der Symbolleiste ihre Schichtungsreihenfolge ändern. Der Anpassungsmodus zeigt alle Teile des Bedienelements, auch diejenigen, die im Bearbeitungsmodus verborgen sind, wie z.B. den Namen-Label oder den Radix auf einem digitalen Bedienelement.

Da die Teile eines Bedienelements voneinander getrennt sind, können Sie den Wert eines Bedienelements im Anpassungsmodus weder ausführen noch ändern. Wie Sie sehen, ist das Bedienwerkzeug deaktiviert. Das Verbindungswerkzeug ist im Bedienelement-Editor immer deaktiviert, da Sie weder das Blockdiagramm verwenden, noch Bedienelemente mit einem Anschlußfeld verbinden.

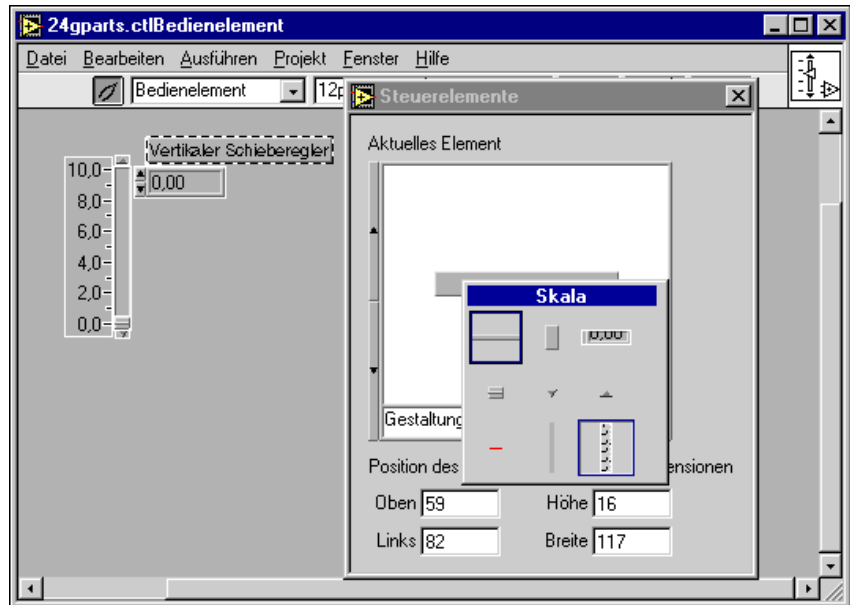
Das Elemente-Fenster des Bedienelement-Editors

Um Ihnen zu helfen, Teile von Bedienelementen in der Größe zu verändern und zu positionieren, wählen Sie **Fenster»Das Elemente-Fenster anzeigen**. Das floatende Fenster, das erscheint, identifiziert die einzelnen Elemente des Bedienelements und zeigt Ihnen die genaue Position und Größe jedes Teils. Die Anzeige *Aktuelles Element* im Elementfenster zeigt ein Bild des Elements, das gegenwärtig im Fenster des Bedienelement-Editors gewählt ist. Ein Menü mit allen Elementen erhalten Sie durch Klicken auf die aktuelle Elementanzeige.

Sie können auch durch die Elemente des Bedienelements rollen, indem Sie auf den Inkrement- oder Dekrementpfeil des aktuellen Elements klicken. Wenn Sie das in der aktuellen Elementanzeige dargestellte Element ändern, wird dieses auf dem Bedienelement des Bedienelement-Editors ausgewählt. Wenn Sie ein anderes Element des Bedienelements im Bedienelement-Editor auswählen oder ändern oder dessen Popup-Menü aufrufen, ändert sich auch das in der aktuellen Elementanzeige gezeigte Element.

Die nachstehende Abbildung zeigt links das Fenster des Bedienelement-Editors, das rechts vom Fenster "Steuerelemente" überlagert ist. Der Namen-Label des Schiebers ist das aktuelle Element, und es ist im Fenster des Bedienelement-Editors ausgewählt. Das Fenster "Steuerelemente" zeigt das Menü mit Elementen, die Sie sehen, wenn Sie auf die Anzeige des

aktuellen Elements klicken. Dieses Beispiel zeigt, daß der Namen-Label das aktuelle Element ist und daß Sie jedoch im Begriff sind, zur Skala zu wechseln.



Das Fenster “Steuerelemente” zeigt Ihnen die genaue Position und Größe des aktuellen Elements an. Diese Werte sind in Pixel angegeben. Wenn Sie ein Element im Bedienelement-Editor verschieben oder in der Größe verändern, werden die Position und Größe im Elementfenster dementsprechend aktualisiert. Sie können die Werte für die Position und Größe auch direkt in das Fenster “Steuerelemente” eingeben, um das Element im Bedienelement-Editor zu verschieben oder in der Größe anzupassen. Dies ist nützlich, wenn Sie zwei Elemente genau gleich groß machen oder ein Element am anderen ausrichten müssen. In der vorstehenden Abbildung zeigt das Elementfenster die Position und Größe des Namen-Labels des Schiebers an — die linke obere Ecke des Labels befindet sich an den Pixelkoordinaten (74, 84), und das Label ist 15 Pixel hoch und 64 Pixel breit.

Das Fenster “Steuerelemente” wird ausgeblendet, wenn Sie zu einem anderen Fenster wechseln. Es wird erneut angezeigt, wenn Sie zum Bedienelement-Editor zurückkehren.

Popup-Menüs für verschiedene Elemente im Anpassungsmodus

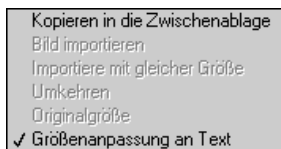
Im Anpassungsmodus wird das Popup-Menü für das Bedienelement im gesamten durch ein Popup-Menü für jedes Element ersetzt. Wenn Sie das Popup-Menü eines Elements aufrufen, sehen Sie ein Menü mit einigen Optionen, die im Bearbeitungsmodus verfügbar sind, und einige Optionen, die nur im Anpassungsmodus verfügbar sind. Unterschiedliche Elemente besitzen unterschiedliche Popup-Menüs. Es gibt drei Grundtypen von Elementen, die angepaßt werden können.

- Gestalterische Elemente, wie z.B. das Gehäuse, der Schieberegler und die Inkrement- und Dekrementpfeile eines Schiebers, sind am häufigsten. Gestalterische Elemente zeigen ein Bild an.
- Textelemente, wie z.B. das Namen-Label des Schiebers. Textelemente bestehen aus einem Bild für den Hintergrund (gewöhnlich nur ein Rechteck) und Text.
- Bedienelemente als Elemente, wie z.B. das numerische Bedienelement, das für die digitale Anzeige des Schiebers verwendet wird. Drehknöpfe, Meßgeräte und Diagramme verwenden für eine digitale Anzeige auch ein numerisches Bedienelement. Manche Bedienelemente sind sogar noch komplizierter. Der Graph verwendet beispielsweise ein Array aus Clustern für das Cursor-Anzeigelement.

In den folgenden Abschnitten werden die verschiedenen Elemente und die Optionen ihrer Popup-Menüs näher beschrieben.

Gestalterische Elemente

Ein gestalterisches Element weist keine dynamische Benutzerinteraktion oder Indikation auf. Die nachstehende Abbildung zeigt ein Popup-Menü eines gestalterischen Elements, wie z.B. eines Schiebergehäuses. Um das Popup-Menü eines gestalterischen Elements aufzurufen, müssen Sie sich im Anpassungsmodus befinden. Sie müssen direkt auf das Element selbst klicken, nicht das Bild des Elements im Fenster "Steuerelemente".



Die folgende Übersicht beschreibt die auf dem Popup-Menü verfügbaren Optionen.

- **Kopieren in die Zwischenablage** — Überträgt eine Kopie des Bildes des betreffenden Elements in die Zwischenablage. Wenn Sie für das Schiebergehäuse **Kopieren in die Zwischenablage** wählen, enthält die Zwischenablage ein Bild eines langen, schmalen Einfassungsrechtecks. Dieses Zwischenablagebild kann in ein beliebiges Frontpanel eingefügt oder mit **Bild importieren** als das Bild für ein anderes Element importiert werden. Diese Bilder sind nichts anderes als die **Gestaltung** in der Palette **Elemente**.

Wenn Sie einfache Vielecke wie das Gehäuserechteck für andere Elemente benötigen, gibt es mehrere Vorteile dafür, von anderen Elementen kopierte Bilder zu kopieren, anstatt sie in einem Paint-Programm zu erstellen. Bestehenden Elementen oder Gestaltungen entnommene Bilder sehen besser aus als Bilder, die in einem Paint-Programm erstellt werden, wenn Sie deren Größe verändern. Ein in einem Paint-Programm gezeichnetes Bild kann beispielsweise nur gleichmäßig wachsen, wobei die Fläche vergrößert, aber auch die Umrandung verstärkt wird. Ein von einem Element wie einem Schiebergehäuse kopiertes Rechteck behält bei der Größenanpassung dieselbe dünne Umrandung bei.

Ein weiterer Vorteil ist, daß integrierte Elemente sowohl auf Farb- als auch auf Schwarz/Weiß-Monitoren grundsätzlich gleich aussehen.

Außerdem können Bilder, die Elementen oder Gestaltungen entnommen sind, mit dem Farbwerkzeug farbig gestaltet werden. Von anderen Quellen importierte Bilder behalten hingegen die Farben bei, mit denen sie importiert wurden, da diese Farben Teil der Bilddefinition sind.

- **Bild importieren** — Ersetzt das aktuelle Bild eines gestalterischen Elements mit dem gegenwärtig in der Zwischenablage befindlichen Bild. Benutzen Sie diese Option, um das Aussehen Ihrer Bedienelemente individuell anzupassen. Sie können beispielsweise Bilder eines offenen und geschlossenen Ventils für einen booleschen Schalter importieren. Befindet sich kein Bild in der Zwischenablage, ist die Option **Bild importieren** deaktiviert.

Im Bedienelement-Editor gibt es einen Shortcut für den Import von Bildern in ein boolesches Bedienelement. Wenn Sie sich im Bearbeitungsmodus befinden, können Sie aus dem Popup-Menü eines booleschen Elements **Bild importieren»TRUE** oder **Bild importieren»FALSE** wählen. Dabei wird das Bild sowohl in den

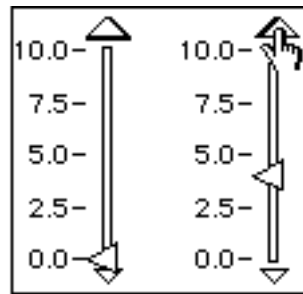
normalen Zustand als auch den entsprechenden Übergangszustand importiert. Weitere Informationen über Übergangszustände finden Sie im Abschnitt *Gestalterische Elemente mit mehreren Bildern* weiter unten in diesem Kapitel.

Im Anpassungsmodus können Sie ebenfalls verschiedene Bilder für Übergangszustände importieren. Rufen Sie hierzu zunächst das Popup-Menü der Taste auf, und benutzen Sie **Bildobjekt**, um das dritte Bild zu ändern. Befindet sich das Bild **TRUE»FALSE** in der Zwischenablage, rufen Sie das Popup-Menü nochmals auf, und wählen Sie **Bild importieren**. Wiederholen Sie diese Schritte für das vierte Bild (**FALSE»TRUE**).

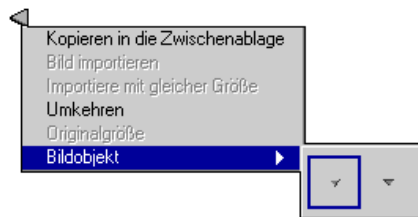
- **Importiere mit gleicher Größe** — Ersetzt das aktuelle Bild, behält aber die Originalgröße des Elements bei, indem das Bild in der Zwischenablage dementsprechend verkleinert oder vergrößert wird. Befindet sich kein Bild in der Zwischenablage, ist **Importiere mit gleicher Größe** deaktiviert.
- **Umkehren** — Stellt das ursprüngliche Erscheinungsbild des Elements wieder her. Die Position des Elements wird dabei nicht geändert. Wenn Sie das Fenster des Bedienelement-Editors öffnen, indem Sie von einem Frontpanel **Bedienelement bearbeiten** wählen, stellt der Editor das Element in der Form wieder her, in der es auf dem Frontpanel abgebildet ist. Wenn Sie das Fenster des Bedienelement-Editors mit **Datei»Öffnen...** öffnen, ist **Umkehren** deaktiviert.
- **Originalgröße** — Setzt das Bild eines Elements auf die Originalgröße zurück. Dies ist bei Bildern nützlich, die Sie aus anderen Anwendungen importieren und dann in der Größe verändern. Manche dieser Bilder sehen nach der Größenanpassung nicht so gut aus wie das Original, so daß Sie sie möglicherweise in ihrer Originalgröße wiederherstellen wollen, um dieses Problem zu beheben. Wenn Sie kein Bild importieren, ist **Originalgröße** deaktiviert.

Gestalterische Elemente mit mehreren Bildern

Manche gestalterischen Elemente besitzen mehrere Bilder, die zu verschiedenen Zeitpunkten angezeigt werden. Diese verschiedenen Bilder haben alle dieselbe Größe und dieselben Farben. Der Inkrementpfeil des Schiebers ist beispielsweise ein Bild eines Dreiecks, das normalerweise leicht vom Hintergrund abgehoben dargestellt wird. Er besitzt ein weiteres Bild, ein vertieftes Dreieck, das erscheint, wenn Sie mit dem Bedienwerkzeug darauf klicken, um den Wert des Schiebers zu erhöhen. Die nachstehende Abbildung zeigt die zwei Bilder eines Inkrement-Pfeiles in Aktion.



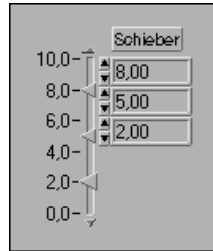
Das Popup-Menü eines gestalterischen Elements mit mehreren Bildern beinhaltet auch die Option **Bildobjekt** (siehe nachstehende Abbildung).



Bildobjekt zeigt alle Bilder an, die zu einem gestalterischen Element gehören. Das gegenwärtig angezeigte Bildobjekt wird von einem dunklen Rand umgeben. Wenn Sie ein Bild importieren, ändern Sie lediglich das aktuelle Bildobjekt. Um ein Bild eines der anderen Bildobjekte zu importieren, wählen Sie zuerst das Bildobjekt aus, und importieren Sie dann das neue Bild.

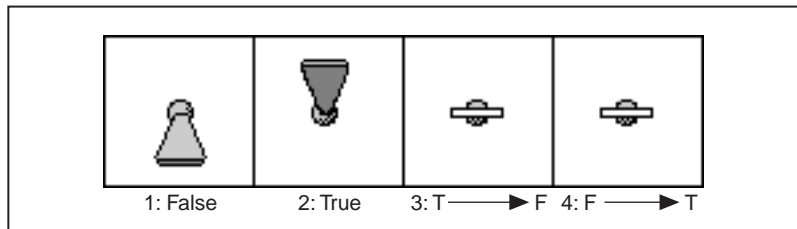
Gestalterische Elemente mit unabhängigen Bildern

Ein gestalterisches Element mit mehreren Bildern kann Bilder verschiedener Größen beinhalten, die jeweils verschiedene Farben haben können. Der Schieber verwendet beispielsweise zwei Bilder mit unterschiedlichen Größen, um anzuzeigen, welcher Schieberegler eines mehrwertigen Schiebers aktiv ist. Im folgenden Beispiel wird durch ein größeres Dreieck angezeigt, daß der mittlere Schieberegler aktiv ist.

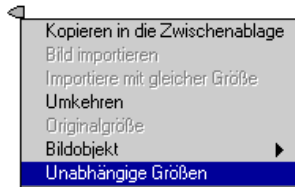


Ein boolescher Schalter besitzt ebenfalls mehrere Bilder. Jedes Bild kann eine andere Größe und unterschiedliche Farben haben. Ein boolescher Schalter besitzt vier verschiedene Bilder — das erste zeigt den Zustand FALSE, das zweite den Zustand TRUE. Das dritte und vierte Bild werden verwendet, wenn Sie die mechanische Aktion eines booleschen Bedienelements entweder auf **Schaltet wenn losgelassen** oder **Latch, wenn losgelassen** einstellen.

Der Wert des booleschen Schalters ändert sich bei diesen beiden mechanischen Aktionen erst, wenn Sie die Maustaste loslassen. Vom Zeitpunkt des Mausclicks bis zum Loslassen der Maustaste wird das dritte oder vierte Bild als Übergangszustand des booleschen Schalters angezeigt. Das dritte Bild gilt für den Übergangszustand TRUE-FALSE, und das vierte für den Übergangszustand FALSE-TRUE. Die folgende Abbildung zeigt einen Umschalter, das dritte und vierte Bild ist gleich, was allerdings nicht immer der Fall ist.



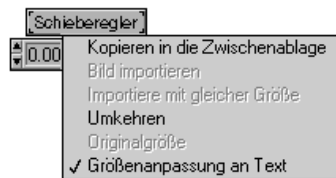
Wenn ein gestalterisches Element Bilder unterschiedlicher Größen haben kann, ist im Popup-Menü des Elements die Option **Unabhängige Größen** gegeben (siehe nachstehende Abbildung).



Unabhängige Größen ist eine Option, die Sie nur im Anpassungsmodus einschalten können, wenn Sie jedes Bild individuell verschieben und in der Größe verändern wollen, ohne die anderen Bilder des gestalterischen Elements zu ändern. Normalerweise ist diese Option nicht aktiviert; wenn Sie das aktuelle Bild des gestalterischen Elements verschieben oder in der Größe verändern, werden die anderen Bilder um denselben Abstand verschoben bzw. in ihrer Größe proportional angepaßt.

Textelemente

Ein Textelement ist ein Bild mit etwas Text. Das Popup-Menü für ein Textelement, wie z.B. ein Namen-Label, beinhaltet einige Menüpunkte, die mit denen im Popup-Menü eines gestalterischen Elements identisch sind. Die anderen Menüpunkte in diesem Menü sind dieselben wie die im Popup-Menü für Text im Bearbeitungsmodus eines Frontpanels. Die nachstehende Abbildung zeigt ein Beispiel für ein Popup-Menü eines Textelements.



Nur das Hintergrundbild des Textelements ist im Elementfenster zu sehen, der Text selbst nicht. Das Hintergrundbild kann angepaßt werden, der Text nicht.

Bedienelemente als Elemente

Ein Bedienelement kann andere Bedienelemente als Elemente enthalten. Ein übliches Beispiel hierfür ist die digitale Anzeige auf einem Schieber, Drehknopf, Meßgerät oder einem Diagramm. Es besteht keinerlei Unterschied zwischen der digitalen Anzeige und dem gewöhnlichen digitalen Bedienelement eines Frontpanels, mit der Ausnahme, daß die digitale Anzeige als Element eines anderen Bedienelements gebraucht wird.

Die digitale Anzeige setzt sich auch aus Elementen zusammen. Wenn Sie das ursprüngliche Bedienelement im Bedienelement-Editor bearbeiten, verhält sich die digitale Anzeige wie ein Einzelement, so daß Sie dessen Elemente einzeln weder ändern noch verschieben können. Sie können jedoch den Bedienelement-Editor für die digitale Anzeige ändern und sie dort anpassen.

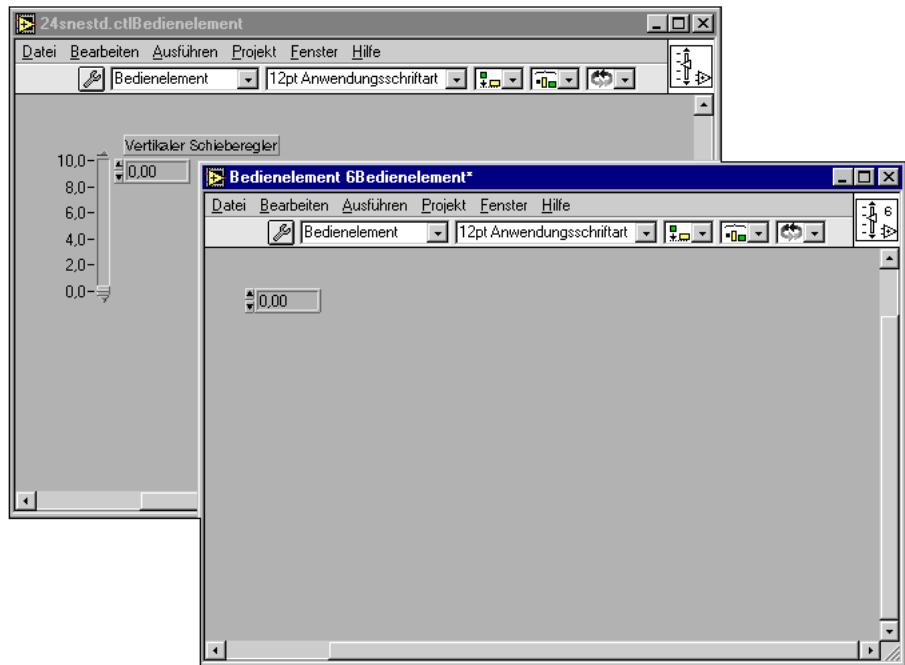
Um ein Bedienelement anzupassen, das Bestandteil eines anderen Bedienelements ist, öffnen Sie den Bedienelement-Editor. Sie können das Fenster des Bedienelement-Editors für das Element direkt vom ursprünglichen Frontpanel aus öffnen, wenn es im Bearbeitungsmodus vom Hauptbedienelement aus separat gewählt werden kann. Die digitale Anzeige kann zum Beispiel separat vom Schieber-Bedienelement aus gewählt werden. Sie können dann **Bearbeiten»Bedienelement bearbeiten...** wählen.

Im Anpassungsmodus können Sie das Fenster des Bedienelement-Editors für das Element stets vom Fenster des Bedienelement-Editors für das Hauptbedienelement aus öffnen. Wählen Sie das Element im Bedienelement-Editor aus und anschließend **Bearbeiten»Bedienelement bearbeiten...** Bedienelement-Editoren können auf diese Weise unbegrenzt ineinander verschachtelt werden, aber die meisten Bedienelemente benutzen andere Bedienelemente nur auf der höchsten Ebene als Elemente. Eine Ausnahme stellt hier der Graph dar, der komplizierte Bedienelemente als Elemente verwendet, die wiederum andere Bedienelemente als Elemente verwenden.

Sie können für das Hauptbedienelement, dessen Anpassung bereits im Gange ist, kein zweites Fenster des Bedienelement-Editors öffnen.

Die nachstehende Abbildung zeigt links den Bedienelement-Editor für den Schieber und rechts ein Fenster des Bedienelement-Editors für die digitale Anzeige. Sie brauchen sich nicht im Anpassungsmodus zu befinden, um

ein verschachteltes Fenster des Bedienelement-Editors zu öffnen, es sei denn, das Element des Bedienelements läßt sich im Bearbeitungsmodus nicht auswählen.



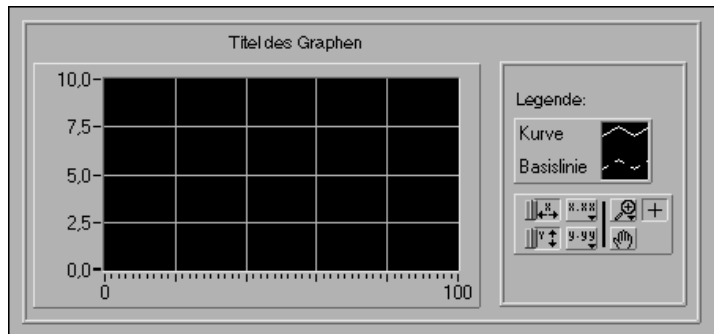
Hinzufügen von gestalterischen Elementen zu benutzerspezifischen Bedienelementen

Wenn Sie im Bedienelement-Editor ein benutzerspezifisches Bedienelement erstellen, können Sie mit gestalterischen Elementen oder Textelementen noch weitere Änderungen daran vornehmen.

Wenn Sie ein Bild oder Text aus der Zwischenablage einfügen, mit dem Beschriftungswerkzeug ein Label erstellen oder ein Bild aus **Elemente»Gestaltung** wählen, wird dieses Bild oder dieser Text Teil des Bedienelements und wird zusammen mit dem Bedienelement angezeigt, wenn Sie es auf ein Frontpanel plazieren. Dies ist im Bearbeitungs- oder Anpassungsmodus im Bedienelement-Editor möglich. Sie können das neue Element wie jedes andere Element auch verschieben, in der Größe verändern oder seine Schichtungsreihenfolge ändern. Das neue Objekt erscheint im Anpassungsmodus als Gestaltungselement im Fenster “Steuerelemente”.

Sie können im Bedienelement-Editor auch Gestaltungselemente löschen.

Die nachstehende Abbildung zeigt einen benutzerspezifischen Graphen mit Gestaltungselementen, einschließlich dem Label *Titel des Graphen*, dem Label *Legende:* und dem Kasten um die Legendelemente herum.



Wenn Sie ein benutzerspezifisches Bedienelement auf anderen Frontpanels verwenden, können Sie zwar die Größe beliebiger Gestaltungselemente verändern, Sie können sie aber nicht verschieben.

Vorbehalte gegen benutzerspezifische Bedienelemente

Es gibt einige Punkte, die Sie bei der Erstellung benutzerspezifischer Bedienelemente beachten müssen.

- Bilder, die auf einer Plattform erstellt worden sind, sehen etwas anders aus, wenn sie auf einer anderen Plattform geladen werden (dies gilt für Bilder, die in einen Pict-Ring importiert oder auch auf einem beliebigen Frontpanel als Hintergrund verwendet werden). Ein Bild mit einer unregelmäßigen Form oder einem transparenten Hintergrund könnte auf einer anderen Plattform einen einfarbig weißen Hintergrund haben. Siehe hierzu Abschnitt [Bildunterschiede](#) in Kapitel 29, [Portierbarkeit und Lokalisierungsthemen](#).
- Der Bedienelement-Editor kann nur das Aussehen eines Bedienelements ändern; er kann nicht das Verhalten eines Bedienelements ändern. Dies hat zwei Auswirkungen.
 - Sie können die Art der Anzeige eines Bedienelements nicht ändern.
 - Sie können die Verhaltensweise eines Bedienelements bei der Bearbeitung und insbesondere bei der Größenanpassung nicht ändern.

Wenn Sie zum Beispiel ein Ring-Bedienelement verlängern, ändert sich auch die Höhe der Inkrement- und Dekrementpfeile. Wenn Sie die Inkrement- und Dekrementpfeile verschieben, damit sie unten im Ring nebeneinander angeordnet sind, verlängert der Ring sie weiter, wenn er verlängert wird, wodurch merkwürdige Ergebnisse erzielt werden.

- Benutzerspezifische Bedienelemente sehen oft richtig aus, verhalten sich aber trotzdem gelegentlich unzuverlässig. Wenn Sie mit einem Bedienelement selber zufrieden sind, nicht aber mit seinem unzuverlässigen Bearbeitungsverhalten, lesen Sie die Ausführungen über strenge Typendefinitionen im nächsten Abschnitt, [Typendefinitionen](#), um herauszufinden, wie die Bearbeitung eingeschränkt werden kann.

Typendefinitionen

Eine *Typendefinition* ist ein Prototyp eines Bedienelements. Sie verwenden den Bedienelement-Editor zum Erstellen eines Prototypen oder einer Typendefinition. Typendefinitionen sind nützlich, wenn Sie dieselbe Art Bedienelement in vielen VIs verwenden. Sie können das Bedienelement als Typendefinition speichern und diese Typendefinition in all Ihren VIs verwenden. Wenn Sie dieses Bedienelement dann ändern, können Sie die einzelne Typendefinitionsdatei aktualisieren, anstatt das Bedienelement in jedem VI, das es verwendet, zu aktualisieren.

Allgemeine Typendefinition: gültige Datentypen

Eine Typendefinition zwingt das Bedienelement überall dort, wo es verwendet wird, zum selben Datentyp. Verwenden Sie eine Typendefinition, wenn Sie ein Bedienelement desselben Datentyps an vielen Stellen verwenden wollen und wenn Sie den Datentyp womöglich überall dort, wo es verwendet wird, automatisch ändern wollen. Nehmen wir zum Beispiel an, daß Sie eine Typendefinition eines digitalen Bedienelements vom Typ Double erstellen wollen und diese Typendefinition anschließend in vielen verschiedenen VIs verwenden. Später ändern Sie die Typendefinition in ein digitales Bedienelement vom Datentyp 16-Bit Integer-Wert. Wenn Sie die Typendefinition ändern, aktualisieren Sie automatisch alle VIs, die diese Typendefinition verwenden. Nur diejenigen Instanzen der Typendefinition, die Sie speziell so eingerichtet haben, daß sie nicht automatisch aktualisiert werden sollen, werden nicht automatisch aktualisiert. Weitere Informationen hierzu finden Sie im Abschnitt [Verwendung von Typendefinitionen](#) weiter unten in diesem Kapitel.

Sie können auch einen Cluster, wie z.B. einen Cluster aus zwei Integer-Werten und einem String, als Typendefinition anlegen. Wenn Sie die Typendefinition in einen Cluster aus zwei Integer-Werten und einem String ändern, können Sie die Typendefinition überall dort aktualisieren, wo sie verwendet wird.

Solange der Datentyp dem des Prototypen entspricht, können Name, Beschreibung, Standardwert, Größe, Farbe oder sogar der Stil des Bedienelements (z.B. ein Drehknopf statt eines Schiebers) bei der Typendefinition abweichen.

Strikte Typendefinition: alle Parameter müssen übereinstimmen

Eine Typendefinition kann auch erzwingen, daß fast *alle* Parameter des Bedienelements in allen Anwendungsfällen identisch sein müssen, nicht nur der Datentyp, sondern auch seine Größe, Farbe und sein Aussehen. Man spricht in diesem Fall von einer *strikten Typendefinition*.

Die einzigen Aspekte eines Bedienelements, die von einem Prototypen einer strikten Typendefinition abweichen können, sind der Name, die Beschreibung und der Standardwert. Nehmen wir zum Beispiel an, Sie erstellen eine strikte Typendefinition, bei der es sich um ein digitales Bedienelement mit einem roten Rahmen vom Datentyp Double handelt. Wie die allgemeine Typendefinition wird automatisch jedes VI, das diese Typendefinition verwendet, automatisch aktualisiert, wenn Sie die strikte Typendefinition in einen Integer-Wert ändern. Im Gegensatz zur allgemeinen Typendefinition erfordern andere Änderungen in der strikten Typendefinition jedoch, wie z.B. die Änderung des roten Randes in einen blauen, daß auch VIs, die diese strikte Typendefinition verwenden, aktualisiert werden müssen. Ferner können Sie die automatische Aktualisierung einer Instanz einer strikten Typendefinition nicht deaktivieren.



Hinweis

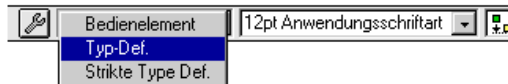
Viele Attributknoten sind bei strikten Typendefinitionen nicht verfügbar. Die einzigen Attribute, die bei strikten Typendefinitionen verfügbar sind, sind diejenigen, die das Aussehen des Bedienelements betreffen, wie z.B. Sichtbar, Deaktiviert, Fokus, Blinken, Position und Grenze.

Typendefinitionen im Blockdiagramm

Wenn Sie eine Typendefinition in einem Blockdiagramm verwenden, hat sie stets die Erscheinungsform einer Konstante, nicht eines Bedien- oder Anzeigeelements. Eine Instanz einer strikten Typendefinition in einem Blockdiagramm fungiert daher als eine Instanz einer allgemeinen Typendefinition. Sie wird nur dann automatisch aktualisiert, wenn sich der Datentyp der strikten Typendefinition ändert.

Erstellen von Typendefinitionen

Sie erstellen eine Typendefinition, indem Sie den Ring auf der Symbolleiste in einem Fenster des Bedienelement-Editors einstellen (siehe nachstehende Abbildung). Richten Sie das Bedienelement wie gewünscht ein, und wählen Sie im Fenster des Bedienelement-Editors **Datei» Speichern**.



Mit **Datei»Öffnen...** können Sie eine beliebige Typendefinition öffnen und speichern. Eine Typendefinition wird stets in einem Fenster des Bedienelement-Editors geöffnet. Alle Änderungen, die Sie in einer Typendefinition vornehmen, wirken sich auf alle VIs aus, die sie verwenden.

Verwendung von Typendefinitionen

Fügen Sie allgemeine Typendefinitionen und strikte Typendefinitionen wie jedes andere benutzerspezifische Bedienelement auch auf dem Frontpanel oder im Blockdiagramm eines VIs ein. Sie können eine Typendefinition auf dem Frontpanel oder im Blockdiagramm wie jedes andere Bedienelement oder jede andere Konstante bearbeiten und ausführen.

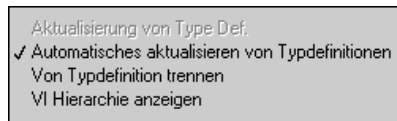


Hinweis

Sie können auf dem Frontpanel nur den Namen, die Beschreibung oder den Standardwert einer strikten Typendefinition ändern.

Ein Bedienelement ist als Typendefinition erkennbar, wenn es im Popup-Menü die Optionen für Typendefinitionen (siehe nachstehende Abbildung) enthält. Sie können eine strikte Typendefinition auf dem Frontpanel oder

im Blockdiagramm erkennen, da Sie sie erstens dort nicht bearbeiten können, und zweitens fehlen im Popup-Menü die meisten Optionen.



Für jede Typendefinition, die Sie auf einem Frontpanel oder in einem Blockdiagramm verwenden, behält das VI eine Verbindung zu der Datei oder VI-Bibliothek bei, in der sie gespeichert ist. Sie können diese Verbindung im Einsatz sehen, wenn Sie eine Typendefinition auf ein Frontpanel oder in ein Blockdiagramm einfügen und die Typendefinition und anschließend **Bearbeiten»Bedienelement bearbeiten...** auswählen. Der Bedienelement-Editor, der geöffnet wird, ist die gespeicherte Typendefinition, mit dem Namen, den Sie ihr gegeben haben, anstatt des allgemeinen Bedienelements *N*.

Aktualisieren von Typendefinitionen

Ihre G-Entwicklungsumgebung stellt sicher, daß die Datentypen überall dort, wo eine Typendefinition verwendet wird, gleich sind. Sie stellt auch sicher, daß alles an einer strikten Typendefinition in jedem Frontpanel, bei dem sie verwendet wird, gleich ist. Sie können automatisch alle allgemeinen Typendefinitionen oder strikten Typendefinitionen auf Ihrem Frontpanel korrigieren, wobei Sie fehlerhafte Instanzen auf dem Frontpanel mit einer in der Datei oder VI-Bibliothek gespeicherten exakten Kopie und im Blockdiagramm mit einer Konstante ersetzen können, deren Datentyp dem des Bedienelements entspricht, das in der Datei oder der VI-Bibliothek gespeichert ist.

Wenn Sie eine Instanz einer Typendefinition auf dem Frontpanel in größerem Rahmen bearbeiten, wie z.B. durch farbliche Gestaltung und Größenanpassung, wollen Sie unter Umständen auf diese automatische Aktualisierungsfunktion verzichten. Sie können in diesem Fall das Popup-Menü der Typendefinition auf dem Frontpanel aufrufen und die Option **Automatisches Aktualisieren von Typdefinitionen** ausschalten. Statt einer automatischen Aktualisierung dieser Typendefinition bei Bedarf weist das VI einen unterbrochenen Ausführungspfeil auf, und die Typendefinition auf dem Frontpanel ist deaktiviert. Sie können das VI erst ausführen, wenn Sie die Typendefinition berichtigt haben, indem Sie entweder die Option **Aktualisierung von Type Def.** aus dem Popup-Menü wählen oder den Datentyp der Typendefinition entsprechend ändern. Die Option **Automatisches Aktualisieren von Typdefinitionen** ist im Popup-

Menü einer strikten Typendefinition nicht verfügbar, da diese stets automatisch aktualisiert wird.

Wenn Sie eine Typendefinition verwenden, können Sie ihr einen anderen Standardwert zuweisen. Wenn sich jedoch der Datentyp der Typendefinition ändert, werden alle Standarddaten anhand des Prototypen so aktualisiert, daß der alte Standardwert nicht in den neuen Datentyp konvertiert werden kann, wie z.B. beim Ersetzen eines numerischen Elements durch einen String. Andernfalls bleiben die individuellen Standardwerte erhalten.

Wenn Sie eine Typendefinition mit Hilfe von **Konstante erzeugen**, **Bedienelement erzeugen** oder **Anzeigeelement erzeugen** verbinden, wird die Typendefinition anhand des Prototypen aktualisiert. Weitere Informationen über die Befehle **Konstante erzeugen**, **Bedienelement erzeugen** oder **Anzeigeelement erzeugen** finden Sie in Kapitel 17, *Einführung in das Blockdiagramm*.

Suchen von Typendefinitionen

Da ein VI eine Verbindung zu allen Typendefinitionen aufrecht erhalten muß, muß die Datei oder VI-Bibliothek, die die Typendefinition enthält, für die Ausführung eines VIs, welches sie verwendet, verfügbar sein. Wenn Sie ein VI öffnen und eine vom VI benötigte Typendefinition nicht gefunden werden kann, werden die Instanzen dieser Typendefinition im VI deaktiviert und der Ausführungspfeil unterbrochen dargestellt. Um dieses Problem zu beheben, müssen Sie entweder die richtige Typendefinition finden und öffnen oder das Popup-Menü der deaktivierten Instanz aufrufen und **Von Typdefinition trennen** wählen. Durch die Trennung von der Typendefinition werden die Datentypbeschränkungen der Instanz entfernt, wodurch sie zu einem gewöhnlichen Bedienelement bzw. einer gewöhnlichen Konstante wird. Sie können die Verbindung nur wiederherstellen, wenn Sie die Typendefinition finden und das Bedienelement damit ersetzen.

Cluster-Typendefinitionen

Wenn Sie eine Typendefinition oder strikte Typendefinition verwenden, bei der es sich um einen Cluster handelt, ist es ratsam, statt der Funktionen "Bündeln" und "Aufschlüsseln" die Funktionen "Nach Namen bündeln" und "Nach Namen aufschlüsseln" im Blockdiagramm dazu zu verwenden, um auf die Elemente des Clusters zuzugreifen. Diese Funktionen rufen Elemente des Clusters nach dem Namen anstatt der Clusteranordnung ab und werden nicht beeinträchtigt, wenn Sie die Elemente neu anordnen oder der Cluster-Typendefinition neue Elemente hinzufügen. Wenn Sie ein

Element, das mit “Nach Namen bündeln” oder “Nach Namen aufschlüsseln” abgefragt wird, löschen, müssen Sie das Blockdiagramm anpassen. Weitere Informationen hierzu finden Sie in der Beschreibung dieser Funktionen in Kapitel 14, *Bedien- und Anzeigeelemente vom Typ Array und Cluster*.

Aufrufen von Code aus anderen Sprachen

Dieses Kapitel beschreibt verschiedene Methoden, mit denen Code, der in anderen Sprachen geschrieben ist, aufgerufen werden kann. Zu diesen Methoden zählen plattformspezifische Protokolle, die Erstellung eines Code Interface Nodes (CIN) zum Aufrufen von Code, der speziell für die Verbindung mit VIs geschrieben worden ist, und die Verwendung eines Knotens für den Aufruf externer Bibliotheken, um Dynamic Link Libraries (.DLL-Dateien) unter Windows, Code Fragments auf dem Macintosh und freigegebene Bibliotheken auf UNIX aufzurufen. Eine weitere Methode stellt der LabWindows/CVI-Funktions-Panel-Konverter dar, mit dem ein Instrumententreiber, der in LabWindows/CVI geschrieben ist, konvertiert werden kann.

Ausführen anderer Anwendungen von Ihren VIs aus

Sie können von Ihren VIs aus andere Anwendungen ausführen. Die Methoden laufen unter Windows und UNIX anders ab als auf dem Macintosh.

(Windows, UNIX) Sie benutzen zum Ausführen anderer Anwendungen von Ihren VIs aus "System Exec". Sie können das einfache System Exec-VI aus der Palette **Funktionen»Kommunikation** dazu verwenden, eine Befehlszeile von Ihrem VI aus auszuführen. Die Befehlszeile kann alle beliebigen Parameter enthalten, die von der zu startenden Anwendung unterstützt wird.

Wenn Sie über TCP/IP (oder in Windows über DDE) auf die Anwendung zugreifen können, dürften Sie in der Lage sein, Daten oder Befehle an die Anwendung zu übergeben. Ziehen Sie die Referenzunterlagen der betreffenden Anwendung hinzu, um herauszufinden, in welchem Rahmen diese kommunikationsfähig ist. Wenn Sie LabVIEW-Benutzer sind, können Sie weitere Einzelheiten über Verfahren zur Verwendung von netzwerkfähigen VIs zur Übertragung von Informationen an andere Anwendungen in Kapitel 20, *Einführung in die Kommunikation*, und Kapitel 21, *TCP und UDP im LabVIEW Benutzerhandbuch* nachlesen.

(Macintosh) Sie benutzen zum Ausführen anderer Anwendungen von Ihren VIs aus Apple Event-VIs. Apple Events ist ein Macintosh-spezifisches Protokoll für die Kommunikation zwischen Anwendungen. Sie können damit Befehle von Anwendungen an Anwendungen senden bzw. andere Anwendungen starten. Wenn Sie LabVIEW-Benutzer sind, können Sie weitere Einzelheiten über unterschiedliche Methoden der Verwendung von Apple Event-VIs in G zum Starten und Steuern anderer Anwendungen in Kapitel 21, *TCP und UDP im LabVIEW Benutzerhandbuch* nachlesen.

Verwendung des Aufrufs externer Bibliotheken

Sie können die meisten freigegebenen Bibliotheken (unter Windows sind dies DLLs oder “Dynamic Link Libraries”, auf dem Macintosh Code Fragments und unter UNIX Shared Libraries) mit dem Knoten zum “Aufruf externer Bibliotheken” aufrufen. Der Knoten “Aufruf externer Bibliotheken” umfaßt eine große Anzahl von Datentypen und Aufrufkonventionen. Sie können mit dem Knoten Funktionen der meisten Standardbibliotheken und benutzerspezifischen Bibliotheken aufrufen.

Der Knoten “Aufruf externer Bibliotheken” ist am besten geeignet, wenn Sie bestehenden Code aufrufen wollen oder wenn Sie mit dem Prozeß der Erstellung einer DLL (Windows), von Code Fragments (Macintosh) oder freigegebenen Bibliotheken (UNIX) vertraut sind. Da eine Bibliothek ein Format verwendet, das über mehrere Entwicklungsumgebungen hinweg standardisiert ist, können Sie beinahe jede Entwicklungsumgebung zur Erstellung einer Bibliothek verwenden, die dann von G aufgerufen werden kann. Konsultieren Sie die Dokumentation Ihres Compilers, um festzustellen, ob dieser standardmäßige freigegebene Bibliotheken generieren kann.

Eine ausführlichere Beschreibung dieses Knotens finden Sie im Abschnitt [Aufruf externer Bibliotheken](#) weiter unten in diesem Kapitel.

In einem Multithread-Betriebssystem können Sie mehrere Aufrufe an eine dynamische Funktionsbibliothek (.DLL-Datei) oder freigegebene Bibliothek gleichzeitig durchführen. Standardmäßig werden alle Bibliothek-Aufrufknoten, einschließlich der Knoten aus vorherigen Versionen, im Thread der Benutzerschnittstelle ausgeführt. Bevor Sie einen Bibliothek-Aufrufknoten als ablaufinvariant konfigurieren, vergewissern Sie sich, daß die aufgerufene Funktion von mehreren Threads gleichzeitig ausgeführt werden kann. Weitere Informationen hierzu finden Sie im Abschnitt [Multithreading](#) in Kapitel 26, [Überblick über das G-Ausführungssystem](#).

Verwendung von CINs (Code Interface Nodes)

Für Anwendungen, bei denen Sie die höchste Leistung erfordern oder mit denen Sie arbiträre Datenstrukturen an C-Code übergeben wollen, können Sie einen Code Interface Node (CIN) erstellen. Durch die Verwendung von CINs können Sie Code aufrufen, der speziell für Verknüpfungen zu VIs der Sprache G geschrieben ist.

Der CIN ist eine sehr allgemeine Methode des Aufrufs von C-Code von G aus. Sie können arbiträr komplexe Datenstrukturen an einen und von einem CIN übergeben. In manchen Fällen erzielen Sie bei der Verwendung von CINs eine höhere Leistung, da Datenstrukturen im selben Format, in dem sie in G gespeichert worden sind, an das CIN übergeben werden.

Um dieses Leistungsniveau zu erzielen, müssen Sie jedoch lernen, wie ein CIN erstellt wird. Dies setzt voraus, daß Sie ein geschulter C-Entwickler sind und sich genügend Zeit nehmen, um den erforderlichen CIN zu erstellen. Da CINs enger mit G verknüpft sind, gibt es außerdem Beschränkungen bei den Compilern, die Sie verwenden können.

Wenn Sie LabVIEW-Benutzer sind und Informationen über die Erstellung von Code Interface Nodes benötigen, konsultieren Sie das Handbuch *LabVIEW Code Interface Reference Manual*, das im PDF-Format (Portable Document Format) auf den Programmdisketten oder der CD Ihrer Software enthalten ist.

Aufruf externer Bibliotheken

Mit dem Aufruf externer Bibliotheken können Sie eine 16-Bit DLL (Windows 3.1), eine 32-Bit DLL (Windows 95/NT), ein Macintosh Code Fragment oder eine freigegebene Bibliothek (UNIX) direkt aufrufen.

(Macintosh) Beim Aufruf externer Bibliotheken wird der Macintosh Code Fragment Manager (CFM) verwendet. Dies ist der Standard auf allen PowerMac-Computern. 680x0 Macintosh-Computer verwenden die CFM-Erweiterung. Außerdem kann der Knoten für den Aufruf externer Bibliotheken auf 680x0 Macintosh-Computern keine Funktionen mit veränderlichen Argumenten aufrufen. Freigegebene Bibliotheken verhalten sich auf dem Macintosh anders als auf anderen Plattformen. Eine Datei kann durchaus mehrere Codefragmente enthalten, die jeweils einen anderen Namen haben.

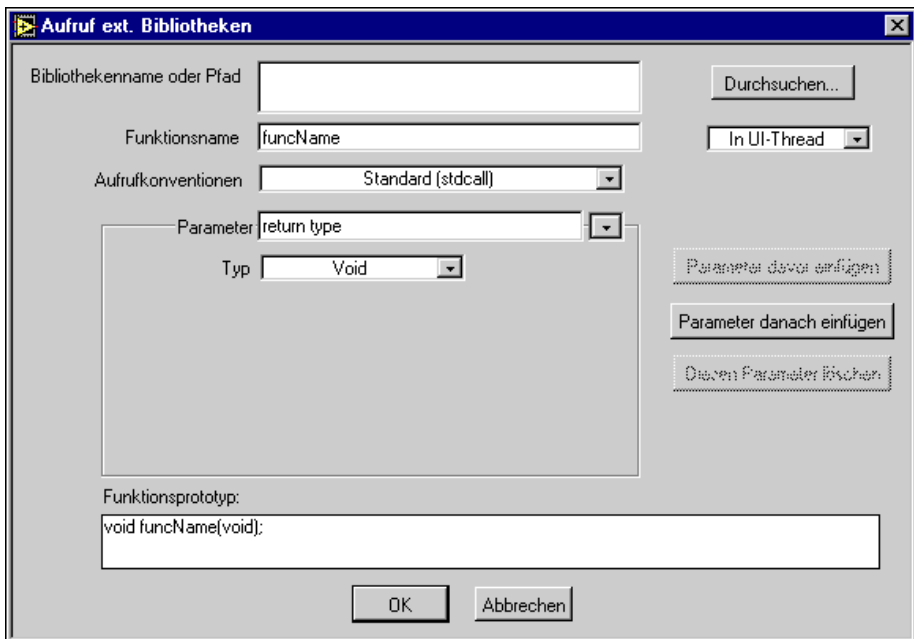


Der Aufruf externer Bibliotheken (siehe Abbildung links) ist über die Palette **Funktionen»Erweitert** verfügbar.

Wenn Sie auf den Aufruf externer Bibliotheken doppelklicken oder aus dem Popup-Menü **Konfigurieren...** wählen, zeigt G ein Dialogfeld an, in dem Sie die Bibliothek, Funktion, Parameter und den Rückgabewert für den Knoten sowie unter Windows geltende Aufrufkonventionen angeben können. Wenn Sie auf die Taste **OK** klicken, fügt der Knoten automatisch die richtige Anzahl von Terminals hinzu und stellt die Terminals auf die richtigen Datentypen ein.

Der Rückgabewert kehrt zum rechten Terminal des oberen Terminalpaars des Knotens zurück. Gibt es keinen Rückgabewert, bleibt dieses Terminalpaar ungebraucht. Jedes weitere Terminalpaar entspricht einem Parameter in der Liste der Funktionsparameter. Sie übergeben einen Wert an die Funktion, indem Sie eine Verbindung zum linken Terminal eines Terminalpaars herstellen. Sie lesen den Wert eines Parameters nach dem Funktionsaufruf mit einer Verbindung, die vom rechten Terminal eines Terminalpaars ausgeht.

Die nachstehende Abbildung zeigt das Dialogfeld “Aufruf externer Bibliotheken”.



Parallel zur Auswahl von Objekten im Dialogfeld zeigt eine Anzeige, ein sogenannter Funktionsprototyp, unten den C-Prototypen der gewählten Funktion an.

**Hinweis**

Aufrufe von freigegebenen Bibliotheken, die mit C++ geschrieben und kompiliert worden sind, wurden nicht getestet und sind eventuell nicht möglich.

Unter Windows 3.1 kann eine obere Grenze von 29 4-Byte-Argumenten mit der Funktion "Aufruf externer Bibliotheken" übergeben werden. Double-Fließkommamaparameter, die nach dem Wert übergeben werden, sind 8-Byte-Mengen und zählen als zwei Argumente. Sie sind also auf 14 Double-Fließkommamaparameter beschränkt, die nach Wert übergeben werden.

Unter Windows 3.1 können sich nicht mehr als zwei Knoten für den Aufruf externer Bibliotheken im Arbeitsspeicher befinden, die dieselbe Funktion innerhalb derselben DLL mit verschiedenen Argumenten aufrufen.

Unter Windows 3.1 müssen 16-Bit DLLs vorliegen, und unter Windows 95/NT 32-Bit DLLs. Wollen Sie eine 16-Bit DLL von Windows 95/NT aus aufrufen, müssen Sie sie entweder als 32-Bit DLL neu kompilieren oder eine "thunking" DLL erstellen. Informationen zu thunking DLLs entnehmen Sie der Microsoft-Dokumentation.

Aufrufkonventionen (Windows)

Verwenden Sie die Aufrufkonventionen des Rings, um die Aufrufkonventionen für die Funktion auszuwählen. Die standardmäßige Aufrufkonvention für Windows 3.1 ist Pascal, und für Windows 95/NT Stdcall. Dieser Standard entspricht der Aufrufkonvention der meisten DLLs. Die Alternative dazu sind die C-Aufrufkonventionen. Die geeigneten Aufrufkonventionen können Sie der Dokumentation zur betreffenden DLL entnehmen.

Parameterlisten

Der Aufruf externer Bibliotheken besitzt anfangs keine Parameter und einen Rückgabewert "void". Sie können auf die Tasten **Parameter davor einfügen** und **Parameter danach einfügen** klicken, um der Funktion Parameter hinzuzufügen. Zum Löschen eines Parameters können Sie auf die Taste **Diesen Parameter löschen** klicken.

Mit dem Parameter-Ring können Sie verschiedene Parameter oder den Rückgabewert auswählen. Sie können dem Parameter dann einen deskriptiveren Namen zuweisen. Der Parametername hat keine Auswirkung auf den Aufruf, er wird aber an Ausgangsverbindungen weitergeleitet. Deskriptive Namen erleichtern den Wechsel zwischen Parametern.

Geben Sie mit dem Typ-Ring den Typ jedes Parameters an. Der Rückgabetyt ist auf **Void** (d.h. die Funktion gibt keinen Wert zurück), **Numerisch** oder **String** beschränkt.

Bei Parametern stehen **Numerisch**, **Array**, **String** oder **An Typ anpassen** zur Auswahl.

Wenn Sie ein Objekt aus dem Typ-Ring wählen, sehen Sie weitere Objekte, mit denen Sie Einzelheiten über den Datentyp und die Übergabe der Daten an die Bibliothek-Funktion angeben können. Der Knoten **Aufruf externer Bibliotheken** verfügt wegen der Vielfalt von Datentypen, die bei verschiedenen Bibliotheken erforderlich ist, über verschiedene Objekte für Datentypen. Konsultieren Sie die Dokumentation über die aufzurufende Bibliothek, um festzustellen, welcher Datentyp zu verwenden ist.

- **Void**—Der Typ `void` wird nur beim Rückgabewert angenommen. Diese Option ist bei Parametern nicht verfügbar. Verwenden Sie diesen Typ, wenn Ihre Funktion keine Werte zurückgibt.
- **Numerisch**—Bei numerischen Datentypen müssen Sie mit dem Datentyp-Ring den genauen numerischen Typ angeben. Die folgenden Optionen sind gegeben.
 - Vorzeichenbehaftete und vorzeichenlose Versionen von 8-Bit, 16-Bit und 32-Bit Integer-Werten
 - 4-Byte Single-Zahlen
 - 8-Byte Double-Zahlen

Extended-Zahlen und komplexe Zahlen können nicht verwendet werden. Sie werden in Standardbibliotheken im allgemeinen nicht verwendet.

Sie müssen auch den Format-Ring verwenden, um anzugeben, ob Sie den Wert oder einen Zeiger des Werts übergeben wollen.



Hinweis

Unter Windows 3.1 können Sie als Rückgabetyt weder Single- noch Double-Datentypen verwenden. Es gibt keine Standardmethode für DLLs zur Rückgabe von Single- und Double-Zahlen. Fließkommarückgabewerte werden von jedem Compiler unterschiedlich implementiert. Wenn Sie eine Single- oder Double-Zahl zurückgeben müssen, tun Sie dies in Form eines Parameters, nicht als Rückgabewert.

- **Arrays**—Sie können den Datentyp von Arrays, (mit denselben Optionen wie bei numerischen Datentypen), die Anzahl von Dimensionen und das Format, mit dem Sie das Array übergeben, angeben. Verwenden Sie die Option **Format**, um zu wählen, ob Sie einen Zeiger zum Array oder ein **Array-Handle** übergeben wollen,

einen Zeiger auf einen Zeiger eines 4-Byte-Werts für jede Dimension, gefolgt von den Daten. Wenn Sie **Array-Datenzeiger** wählen, übergeben Sie die Arraydimension als separate(n) Parameter.

**Hinweis**

Unter Windows 3.1 verwendet der Knoten für den Aufruf externer Bibliotheken nur das Format des Array-Datenzeigers für Arrays. Außerdem können Sie unter Windows 3.1 angeben, daß die Daten mit einem Zeiger Huge übergeben werden. Sie können einen Zeiger Huge verwenden, wenn Sie mehr als 64 Kilobyte an Daten im Array übergeben müssen. Schalten Sie diese Option nur ein, wenn die DLL, die Sie aufrufen, den Datenzeiger Huge erwartet. Wenn Sie versuchen, einen Zeiger Huge an eine Funktion zu übergeben, die einen normalen Zeiger erwartet, kann die Anwendung abstürzen.

**Vorsicht**

Versuchen Sie nicht, die Größe eines Arrays mit Systemfunktionen, wie z.B. "realloc", anzupassen. Dies kann zu einem Systemabsturz führen.

Strings — Sie können das Stringformat für Strings angeben. Die Optionen für das Stringformat lauten C, Pascal oder G.

**Vorsicht**

Legen Sie Ihrer Auswahl des Stringformats den Stringtyp zugrunde, den die Bibliothekfunktion erwartet. Die meisten Standardbibliotheken erwarten entweder einen C-String (C-String, dem ein Nullzeichen folgt) oder einen Pascal-String (String, dem ein Längenbyte vorangeht). Wenn die Bibliothekfunktion, die Sie aufrufen, speziell für G geschrieben ist, empfiehlt sich womöglich das Format "String-Handle", bei dem es sich um einen Zeiger auf einen Zeiger auf vier Bytes Längeninformation handelt, dem Stringdaten folgen.

**Hinweis**

Unter Windows 3.1 können Sie weder das Format "String Handle" verwenden, noch einen String zurückgeben. Auf Plattformen, die einen Rückgabetyt "String" unterstützen, wird der String sofort in einen Puffer kopiert. Die Zuordnung des Strings wird nicht aufgehoben.

**Vorsicht**

Versuchen Sie nicht, die Größe eines Strings mit Systemfunktionen, wie z.B. "realloc", anzupassen. Dies kann zu einem Systemabsturz führen.

An Typ anpassen — erlaubt Ihnen, arbiträre Datentypen der Sprache G an DLLs zu übergeben. Sie werden auf dieselbe Weise wie an einen CIN übergeben. Dies bedeutet folgendes:

- Skalare werden nach Referenz übergeben (ein Zeiger auf den Skalar wird an die Bibliothek übergeben).
- Arrays und Strings werden als Handle übergeben (Zeiger auf einen Datenzeiger). Siehe hierzu das Handbuch *LabVIEW Code Interface Reference Manual*, das auf den Programmdisketten bzw. der CD Ihrer

Software im PDF-Format (portable document format) im Lieferumfang enthalten ist.

- Cluster werden nach Referenz übergeben.
- Skalare Elemente in Arrays oder Clustern sind in-line. Ein Cluster, der ein numerisches Element enthält, wird beispielsweise als Zeiger an eine Struktur übergeben, die ein numerisches Element enthält.
- Cluster innerhalb von Arrays sind in-line.
- Strings und Arrays innerhalb von Clustern werden mit einem Handle abgefragt.



Hinweis

Unter Windows 3.1 können Sie An Typ anpassen nicht verwenden.

Aufrufen von Funktionen, die andere Datentypen erwarten

In manchen Fällen könnten Sie eine Funktion antreffen, die einen Datentyp erwartet, den G nicht verwendet. Sie können den Aufruf externer Bibliotheken beispielsweise nicht dazu verwenden, einen arbiträren Cluster oder ein Array nichtnumerischer Daten zu übergeben.

Je nach Datentyp sind Sie möglicherweise in der Lage, die Daten durch Erstellen eines Strings oder Arrays aus Bytes zu übergeben, der bzw. das ein binäres Abbild der zu sendenden Daten enthält. Sie können binäre Daten erstellen, indem Sie Datenelemente in Strings abbilden und diese verketteten.

Eine andere Option besteht darin, eine Bibliotheksfunktion zu schreiben, die Datentypen annimmt, die von G verwendet werden, und zwar mit Parametern, die Datenstrukturen erzeugen, die die Bibliotheksfunktion erwartet, und dann die Bibliotheksfunktion aufrufen.

Schließlich können Sie statt dessen auch einen Code Interface Node schreiben. Code Interface Nodes können arbiträre Daten annehmen, sind aber schwieriger zu meistern, da Sie Datenformate und -speicherung in G verstehen müssen.

LabWindows/CVI-Funktions-Panel-Konverter



Hinweis

Diese Funktion steht auf dem Macintosh nicht zur Verfügung.

Der LabWindows/CVI-Funktions-Panel-Konverter automatisiert den Prozeß der Konvertierung von Instrumententreiber, die in LabWindows/CVI geschrieben sind, damit sie in G verwendet werden können. Ein LabWindows/CVI-Instrumententreiber besteht aus einer C-Quelle und Headerdateien, einer DLL (Dynamic Link Library) (**Windows**) oder freigegebenen Bibliothek (**UNIX**), welche den kompilierten Code enthält, sowie einer LabWindows/CVI-spezifischen Datei, die als Funktions-Panel (FP)-Datei bezeichnet wird. Funktions-Panel-Dateien werden in LabWindows/CVI verwendet, damit Benutzer bei der Bearbeitung einer Anweisung eines C-Funktionsaufrufs Argumente und Rückgabewerte angeben können, indem sie Werte in ein Pop-up-Fenster eintragen.

Der LabWindows/CVI Funktions-Panel-Konverter konvertiert jede Funktion in ein VI, wobei der Typ, die Datendarstellung und Platzierung von Bedienelementen auf VI-Frontpanels anhand der Informationen in der CVI FP-Datei bestimmt werden. Jedes generierte VI verwendet einen Knoten für den Aufruf externer Bibliotheken im Blockdiagramm, um die passende C-Routine in der passenden Treiberbibliothek aufzurufen.

Obgleich viele Instrumententreiber ohne Aufrufe von Bibliotheksfunktionen vollständig im Diagramm geschrieben werden, hat der LabWindows/CVI Funktions-Panel-Konverter in manchen Fällen seine Vorteile. Bei Instrumenten, bei denen kein G-Treiber verfügbar ist, macht es dieses Werkzeug relativ leicht, einen CVI-Treiber zu nutzen, falls einer existiert. Hat man jedoch die Wahl, ist ein echter G-Treiber, der aus G-Diagrammen besteht, vorzuziehen, da er für Kunden leicht zu überprüfen und zu ändern ist und er in der G-Umgebung ein gutes Multitasking-Verhalten aufweist. Bibliothekaufrufe sind synchron, so daß VIs, die parallel ausgeführt werden, für die Dauer des Aufrufs angehalten werden.

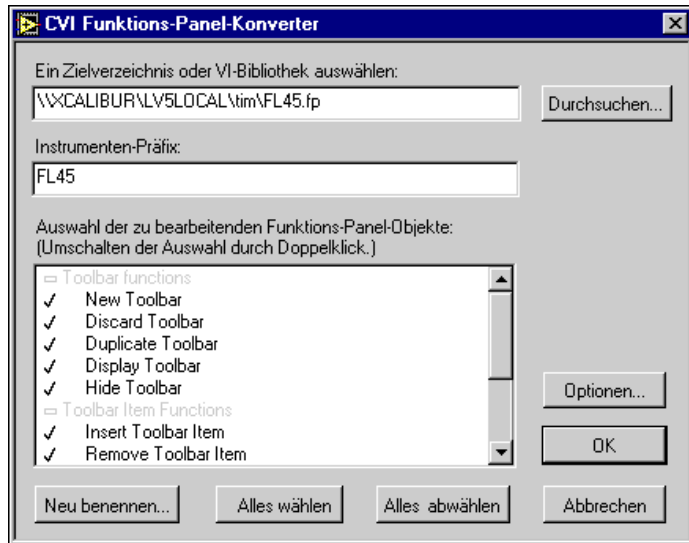


Hinweis

Alle CVI .dll-Dateien erfordern Run-Time-Unterstützung. Installieren Sie die CVI Run-Time-Engine, um VIs zu verwenden, die beim Konvertierungsprozeß erzeugt werden.

Konvertierungsprozeß

Wenn Sie **Datei»CVI FP Datei konvertieren...** wählen, erscheint ein Dialogfeld, in dem Sie die LabWindows/CVI Funktions-Panel-Datei auswählen müssen. Wenn Sie eine FP-Datei auswählen, sehen Sie das folgende Dialogfeld, in dem Sie angeben, wo die neuen VIs gespeichert und welche Treiberfunktionen konvertiert werden sollen.



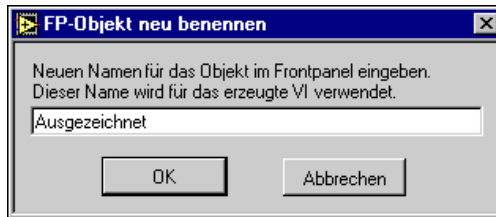
Sie geben das Zielverzeichnis oder die VI-Bibliothek ins oberste Textfeld ein. Das empfohlene Ziel wird angezeigt, und Sie können den Pfad ändern, um die neuen VIs an einem beliebigen anderen Speicherort abzulegen. Mit der Taste **Durchsuchen...** können Sie das Ziel wie gewohnt mit einem Dateialogfeld bestimmen.

Das Textfeld **Instrumenten-Präfix** zeigt den Instrumentenpräfix gemäß der Funktions-Panel-Datei an. Dieser Präfix wird von CVI allen C-Funktionsnamen in der Funktions-Panel-Baumstruktur vorangestellt. Der Konverter stellt diesen Präfix dann ebenfalls den Namen der von ihm generierten VIs voran. Der vorgeschlagene Präfix wird basierend auf der DLL angezeigt, Sie können diesen String aber beliebig ändern oder löschen.

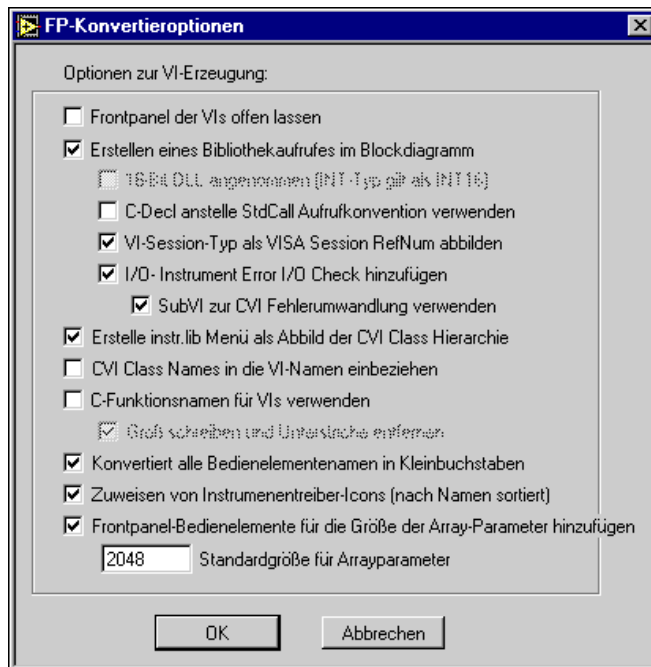
Der Rest des Dialogfeldes enthält die Auswahl der zu konvertierenden Funktions-Panel-Objekte. Ein Listenfeld listet alle Objekte in der Funktions-Panel-Baumstruktur wie in LabWindows/CVI nach Klassen hierarchisch eingerückt auf. Die Klassennamen werden auch aufgelistet, sie sind aber als nicht bedienbar markiert und nicht auswählbar. Anfangs

sind alle Knoten, die in der Funktions-Panel-Datei gefunden werden, ausgewählt. Ausgewählte Objekte sind mit einem Häkchen und Klassennamen mit einem Rechteck markiert.

Durch Doppelklicken auf ein Objekt wird die Auswahl umgeschaltet. Mit den Tasten **Alles wählen** und **Alle Markierungen aufheben** kann der jeweilige Vorgang bequem für alle Objekte auf einmal durchgeführt werden. Sie können auch einmal auf ein Objekt klicken, um dieses hervorzuheben, und können dann auf **Umbenennen** klicken. Dadurch wird das folgende einfache Dialogfeld zum Umbenennen eines FP-Objekts eingeblendet.



Mit der Taste **Optionen...** wird das folgende Dialogfeld aufgerufen.



Es stehen folgende Konvertierungsoptionen zur Verfügung.

Frontpanel der VIs offen lassen — Der Konverter behält VIs nach Abschluß der Konvertierung im Arbeitsspeicher und läßt ihre Frontpanels offen, anstatt jedes VI nach der Speicherung auf dem Datenträger zu schließen (standardmäßig ausgeschaltet).

Erstellen eines Bibliothekaufrufes im Blockdiagramm — Der Konverter fügt einen Knoten für den Aufruf externer Bibliotheken in das Blockdiagramm jedes VI ein und stellt alle Frontpanel-Verbindungen ordnungsgemäß her. Ist diese Option nicht eingeschaltet, wird nur das Frontpanel erstellt, und es wird nichts in das Blockdiagramm eingefügt (standardmäßig eingeschaltet).

16-Bit DLL angenommen — (Windows 3.1) Der Konverter behandelt den Integertyp des LabWindows/CVI Funktions-Panels beim Erstellen des Typendeskriptors, der die Argumente des Aufrufs externer Bibliotheken beschreibt, als einen INT16-Typ, und nicht als INT32-Typ. Dies ist beim Aufruf von DLLs notwendig, die mit Compilern von Drittanbietern, wie z.B. Borland C, erstellt worden sind (standardmäßig eingeschaltet).

I/O-Instrument Error I/O Check hinzufügen — Der Konverter legt im Blockdiagramm jedes erstellten VI Fehlerbehandlungscode ab. Wenn Sie diese Option wählen, werden Fehlereingang- und Fehlerausgang-Cluster auf dem Frontpanel abgelegt; sie werden unterhalb aller vom LabWindows/CVI Funktions-Panels abgeleiteten Bedien- und Anzeigeelemente positioniert. Die Frontpanel-Terminals und der Aufruf externer Bibliotheken im Blockdiagramm werden in eine Case-Struktur aufgenommen, die nur ausgeführt wird, wenn das Statusfeld des Fehlereingangs FALSE anzeigt, womit ein fehlerloser Zustand angezeigt wird (standardmäßig eingeschaltet).

SubVI zur CVI Fehlerumwandlung verwenden — Der Konverter legt ein SubVI im Blockdiagramm jedes VI ab, um LabWindows/CVI-artigen Fehlercodes G-artige Fehlercluster zuzuordnen, die für die Zufuhr an den allgemeinen Fehlerbehandler geeignet sind. Der Integer-Rückgabewert vom Knoten für den Aufruf externer Bibliotheken, der Fehlereingang-Cluster und der Name des aktuellen VIs werden an das SubVI übergeben. Stellt das SubVI einen Fehler fest, wird er an den Fehlerausgang übergeben und der Status auf TRUE gesetzt. Wird eine Warnung festgestellt, wird sie an den Fehlerausgang übergeben und der Status auf FALSE gesetzt. Andernfalls wird der Fehlereingang an den Fehlerausgang übergeben. Ist diese Option nicht gewählt, wird kein SubVI abgelegt, und das Blockdiagramm wird so konstruiert, daß im Fehlerausgang nur dann ein

Fehler angezeigt wird, wenn der Rückgabewert des Aufrufs externer Bibliotheken kleiner als Null ist (standardmäßig eingeschaltet).

CVI Class Names in die VI-Namen einbeziehen — Erstellt entweder basierend auf dem Funktions-Panel-Namen oder dem C-Funktionsnamen für jede Instrumententreiberoption automatisch einen Namen für jedes VI. Ist diese Option aktiviert, wird der Klassenname, der mit dem LabWindows/CVI Funktions-Panel verbunden ist, automatisch dem für die Funktion generierten Namen vorangestellt.

C-Funktionsnamen für VIs verwenden — Normalerweise konstruiert der Konverter die Namen der VIs direkt anhand der Funktions-Panel-Objektnamen, indem er einfach den Instrumentenpräfix mit der Erweiterung `.vi` kombiniert. Leider ergeben sich hierbei nicht immer eindeutige Namen für alle Objekte, da LabWindows/CVI nicht vorschreibt, daß die “Zweigelemente” der Funktions-Panel-Baumstruktur eindeutige Namen aufweisen müssen. Das Einstellen dieser Option hat zur Folge, daß der Konverter VI-Namen aus den eigentlichen C-Funktionsnamen, denen die FP-Objekte entsprechen, konstruiert und somit eindeutige Ergebnisse gewährleistet.

Um Probleme zu vermeiden, die durch doppelte Namen hervorgerufen werden, prüft der Konverter, wenn er die Liste der anzuzeigenden Funktions-Panel-Objekte erstellt, ob alle Objekte eindeutig sind. Nicht eindeutige Objekte werden mit einem Nullsymbol gekennzeichnet und sind nicht durch Doppelklicken auswählbar. Um Probleme zu vermeiden, die durch doppelte Namen hervorgerufen werden, ruft der Konverter vor dem Öffnen des Dialogfeldes eine Warnungsmittteilung auf, wenn er Namenskonflikte feststellt, und schaltet automatisch die Option **C-Funktionsnamen für VIs verwenden** ein. Wenn Sie es vorziehen, statt dessen die Funktions-Panel-Namen zu verwenden, können Sie diese Option manuell ausschalten und die Objekte, die Namenskonflikte darstellen, einzeln umbenennen. Umbenannte Objekte behalten die vom Benutzer eingegebenen Namen auch dann bei, wenn die Option **C-Funktionsnamen für VIs verwenden** geändert wird.

Groß schreiben und Underscores entfernen — Da Namen, die anhand von C-Funktionsnamen gebildet werden, weniger kommunikativ als diejenigen sind, die von Funktions-Panel-Objekten abgeleitet werden, versucht diese Option, sie “attraktiver” zu gestalten, indem die Anfangsbuchstaben in Großbuchstaben umgewandelt und Unterstriche durch Leerzeichen ersetzt werden. Da Abkürzungen nicht erweitert werden, sind die Ergebnisse jedoch möglicherweise immer noch nicht so

gefällig wie die FP-Namen, aber dies ist bei doppelten FP-Objektnamen, die Sie nicht individuell umbenennen wollen, unumgänglich.

Konvertiert alle Bedienelementennamen in Kleinbuchstaben — Wandelt Bedienelementennamen in Anlehnung an VXI *Plug&Play*-Standards in Kleinbuchstaben um.

Zuweisen von Instrumententreiber-Icons (nach Namen sortiert) — Weist VIs Icons basierend auf dem Namen der generierten VIs zu. Der Konverter sucht im Namen der Funktion Schlüsselwörter wie z.B. initialisieren, schließen, Selbsttest, konfigurieren oder messen und verwendet das entsprechende Icon. Werden keine Schlüsselwörter gefunden, wird ein Standardicon verwendet. Außerdem werden bis zu sieben Zeichen des Instrumentenpräfixes links oben auf das Icon gestempelt.

Standardgröße für Arrayparameter — Wenn eine DLL-Funktion eines Instrumententreibers ein Array ausgibt, muß der Arbeitsspeicher, in den die Funktion das Array schreibt, vorab reserviert und an die DLL übergeben werden. Mit dieser Option können Sie die Standardgröße in Elementen auswählen, die für Arrays reserviert werden soll. Ist ein Argument des Knotens für den Aufruf externer Bibliotheken ein Array, legt der Konverter eine Funktion "Array initialisieren" ab, um ein Array für die Übergabe in den Knoten zu erstellen. Die Anfangsgröße solcher Arrays wird mit der Option **Standardgröße für Arrayparameter** angegeben. In der .out-Datei wird eine Warnung generiert, so daß Sie VIs, die dieses Konstrukt enthalten, leicht finden und Sonderfälle individuell behandeln können.

Wählen Sie im Hauptdialogfeld des LabWindows/CVI Funktions-Panel-Konverters **OK**. Dadurch wird ein Dateidialogfeld aufgerufen, in dem Sie die entsprechende Bibliothek zu der zu konvertierenden FP-Datei auswählen (wenn die Option **Erstellen eines Bibliothekaufrufes im Blockdiagramm** aktiviert ist). Bei Drücken der Abbruchtaste bleibt lediglich der Bibliothekspfad im Knoten "Aufruf externer Bibliotheken" unbestimmt. Die Konvertierung wird dadurch nicht abgebrochen.

Nachdem Sie auf **OK** klicken, blendet der Konverter ein Status-Dialogfeld mit der Statusanzeige ein, in dem der Name jedes neu erstellten VI bei der Erstellung angezeigt wird. Eine Protokolldatei mit dem Namen `präfix.out` wird erstellt, in der alle erstellten VIs sowie Warnungen oder Fehler aufgelistet werden, die während der Konvertierung aufgetreten sind. Sollten in der Tat Warnungen oder Fehler auftreten, wird Ihnen mitgeteilt, im Anschluß an die Konvertierung diese Datei zu überprüfen.

Überblick über das G-Ausführungssystem

Dieses Kapitel beschreibt das Multitasking und die Ausführung von VIs.

Mit dem G-Ausführungssystem können Sie mehrere VIs gleichzeitig ausführen. Außerdem kann es in einem bestimmten VI mehrere parallele Zweige geben, die auch wieder gleichzeitig ausgeführt werden können.

Beim normalen Gebrauch besteht kein Grund, sich mit den Einzelheiten des Multitasking zu beschäftigen. Sie können sich Teile eines Diagramms und mehrere VIs als parallel ablaufend vorstellen. Mit der Multitasking-Fähigkeit des Ausführungssystems können Sie ein VI bearbeiten oder in Einzelschritten durchgehen, während die Ausführung anderer VIs fortgesetzt wird. Und falls Sie versehentlich ein VI mit einer Endlosschleife erstellen, wird dadurch nicht Ihr Computer gesperrt oder die Ausführung anderer VIs unmöglich gemacht.

In manchen Anwendungen, bei denen Timing kritischer ist, ist ein besseres Verständnis der Funktionsweise des Multitaskingsystems durchaus wichtig. Dieses Thema wird im folgenden Abschnitt behandelt.

Multitasking

Die meisten Computer haben nur einen Prozessor, d.h. es wird immer nur ein Task auf einmal ausgeführt. Multitasking wird dadurch erzielt, daß ein Task eine bestimmte Zeit lang ausgeführt wird und sich dann mit anderen Tasks abwechselt. Solange das Zeitintervall für jeden Task klein genug ist, besteht der Eindruck, daß mehrere Tasks gleichzeitig ablaufen.

Die Betriebssysteme Windows 3.1 und Macintosh nutzen hauptsächlich eine Form des Multitasking, die als kooperatives Multitasking bezeichnet wird, bei der jedes Programm so geschrieben sein muß, daß es von Zeit zu Zeit Ressourcen an andere Tasks freigibt. Die meisten Anwendungen verbringen eine beträchtliche Zeit damit, auf Ereignisse wie z.B. Benutzeraktionen zu warten, so daß allein dadurch normalerweise eine Reihe von Gelegenheiten gegeben ist, um Ressourcen an andere Tasks

abzutreten. Wenn ein beliebiger Task jedoch zu lange beschäftigt ist, ohne Ressourcen freizugeben, erhalten andere Tasks nicht die Gelegenheit zur Ausführung. Kooperatives Multitasking funktioniert nur dann, wenn alle Programme mit Multitasking im Sinn geschrieben worden sind. In der Praxis sind die meisten Programme nicht sehr gut beim kooperativen Multitasking, wenn sie allgemeine Tasks wie z.B. das Drucken, Ziehen von Fenstern oder Speichern von Dateien durchführen.

Viele Betriebssysteme unterstützen eine andere Form des Multitasking, die als preemptives Multitasking bezeichnet wird. Beim preemptiven Multitasking kümmert sich das Betriebssystem um die Schaltung und zeitliche Planung von Tasks. Jedem Task wird eine begrenzte Zeitspanne zur Ausführung eingeräumt. Ist die Zeit für einen bestimmten Task abgelaufen, wird ein Task zum Anhalten gezwungen, während ein anderer die Ausführung aufnimmt. Dieses Umschalten erfolgt zu arbiträren Zeitpunkten, ohne jegliche Codierung von seiten des Entwicklers.

Multithreading

Wenn eine Anwendung einen Task durchführen muß, gibt es gewöhnlich mehrere Arten, dieses Problem anzupacken. Eine Möglichkeit ist, den Task schrittweise abzuschließen. Die meisten Tasks können jedoch in kleinere Tasks unterteilt werden, die theoretisch wiederum parallel ausgeführt werden können. Wenn Sie beispielsweise ein Dokument drucken, beginnt die Anwendung den Druck im Hintergrund, so daß Sie andere Dokumente bearbeiten können. Multithreading ist der Fachausdruck für die Anwendung des Multitasking in einer einzelnen Anwendung, wobei die einzelnen Tasks in kleinere Tasks aufgeteilt werden, damit sie parallel ausgeführt werden können.

Wie beim kooperativen Multitasking können Sie Programme so entwerfen, daß sie ein kooperatives Multithreading innerhalb einer Anwendung ausführen. Jeder Thread muß so geschrieben sein, daß er immer wieder Ressourcen freisetzt, damit andere Threads oder Tasks ausgeführt werden können. Da alle Threads in einer Multithread-Anwendung als Teil desselben Programms entwickelt werden, hat der Entwickler dabei mehr Kontrolle, um sicherzustellen, daß die Kooperation stattfindet. Es ist jedoch schwierig, dies einheitlich in einer Anwendung durchzuführen, und es ist daher wahrscheinlich, daß die meisten Anwendungen mit einem kooperativen Multithreading etwas an Parallelität einbüßen.

Manche Betriebssysteme bieten einen Mechanismus für preemptives Multithreading. Das Betriebssystem verwaltet die Kontextschalter zwischen den Threads des Programms in genau derselben Weise, wie es die anderen Programme im preemptiven Multitaskingsystem verwaltet. Das Ergebnis kann eine bessere Aufteilung des Prozessors unter den aktiven Threads des Programms sein.

Das G-Ausführungssystem

Das G-Ausführungssystem benutzt auf Betriebssystemen, die dies unterstützen, das preemptive Multithreading sowie das kooperative Multithreading. Selbst auf Systemen mit preemptivem Multithreading wird eine begrenzte Anzahl von Threads verwendet, so daß in bestimmten Fällen auf kooperatives Multithreading zurückgegriffen wird.

(Windows 95/NT, Solaris 2 und PowerMAX) Die Anwendung wird als Multithread-System ausgeführt. Das Ausführungssystem führt VIs mit preemptivem Multitasking unter Verwendung von Threads aus. Es ist jedoch nur eine begrenzte Anzahl von Threads verfügbar, so daß es bei äußerst parallelen Anwendungen auf kooperatives Multitasking zurückgreift, wenn die verfügbare Threads aufgebraucht sind. Außerdem verwaltet das Betriebssystem das preemptive Multitasking zwischen der Anwendung und anderen Tasks.

(Macintosh und Windows 3.1) Die Anwendung wird als Singlethread-System ausgeführt. Das Ausführungssystem führt VIs mit kooperativem Multitasking unter Verwendung eines eigenen Zeitplanungssystems aus. Die Anwendung steht mit anderen Anwendungen in einer kooperativen Multitasking-Beziehung, indem es von Zeit zu Zeit Ressourcen freisetzt.

(HP-UX und Solaris 1) Die Anwendung wird als Singlethread-System ausgeführt. Das Ausführungssystem führt VIs mit kooperativem Multitasking unter Verwendung eines eigenen Zeitplanungssystems aus. Das Betriebssystem erledigt das preemptive Multitasking zwischen der Anwendung und anderen Tasks.

Grundlegendes Ausführungssystem

Die folgende Beschreibung gilt sowohl für das Singlethread- als auch das Multithread-Ausführungssystem.

Das Ausführungssystem führt eine Warteschlange aktiver Tasks. Bei drei parallel ausgeführten Schleifen wird beispielsweise zu jedem beliebigen Zeitpunkt ein Task ausgeführt, während die beiden anderen in der Warteschlange warten. Wenn man annimmt, daß alle Tasks dieselbe Priorität haben, wird ein Task eine gewisse Zeit lang ausgeführt. Dieser Task wird dann an das Ende der Warteschlange zurückgestellt, und der nächste Task wird eine Zeit lang ausgeführt. Wenn die Ausführung eines Tasks abgeschlossen ist, entfernt das Ausführungssystem ihn aus der Warteschlange.

Das Ausführungssystem führt das erste Element der Warteschlange aus, indem der generierte Code des VIs aufgerufen wird. Zu einem bestimmten Zeitpunkt prüft der generierte Code dieses VIs beim Ausführungssystem nach, ob einem anderen Task Ausführungszeit zugeteilt wird. Ist dies nicht der Fall, wird die Ausführung des VIs fortgesetzt.

Verwalten der Benutzerschnittstelle bei einer Singlethread-Anwendung

Neben der Ausführung von VIs muß das Ausführungssystem die Interaktion mit der Benutzerschnittstelle koordinieren. Wenn Sie auf eine Taste klicken, ein Fenster verschieben oder den Wert eines Schieber-Bedienelements ändern, erledigt das Ausführungssystem die Verwaltung dieses Vorgangs und sichert die Fortsetzung der VI-Ausführung im Hintergrund.

Das Singlethread-Ausführungssystem führt Multitasking durch, indem es zwischen der Beantwortung von Benutzeraktionen und ablaufenden VIs hin- und herschaltet. Wenn ein VI die Ausführung an das Ausführungssystem zurückgibt, prüft dieses, ob eine Behandlung irgendwelcher Ereignisse der Benutzerschnittstelle erforderlich ist. Ist dies nicht der Fall, kehrt das Ausführungssystem entweder zum VI zurück oder entnimmt der Warteschlange den nächsten Task.

Wenn Sie Tasten oder Pulldown-Menüs betätigen, kann die Durchführung dieses Vorgangs möglicherweise einige Zeit in Anspruch nehmen. Das Ausführungssystem behält ablaufende VIs im Hintergrund, indem es zwischen der Beantwortung Ihrer Interaktion mit dem Bedienelement oder Menü und ablaufenden VIs hin- und herschaltet.

Multithread-Anwendung und mehrere Ausführungssysteme

Multithread-Versionen besitzen mehrere Ausführungssysteme. Sie können VIs unter **Bevorzugtes Ausführungssystem** im Dialogfeld **VI-Einstellungen» Ausführungsoptionen** eines von sechs Ausführungssystemen zuweisen.



Die folgenden Ausführungssysteme stehen zur Auswahl.

- Benutzeroberfläche
- Standard
- Instrument I/O
- Datenerfassung
- Andere 1
- Andere 2

Der Zweck mehrerer Ausführungssysteme besteht darin, VIs, die unabhängig von anderen VIs auszuführen werden müssen, eine grobe Trennung zur Verfügung zu stellen. Standardmäßig werden VIs im Ausführungssystem "Standard" ausgeführt, das in anderen Threads als die Benutzeroberfläche ausgeführt wird. Das Ausführungssystem "Instrument I/O" hat den Zweck, VISA, GPIB und serielle I/O vom Eingreifen in andere VIs abzuhalten. Das Ausführungssystem "Datenerfassung" ist analog für die VIs der Datenerfassung eingestellt.

Das Ausführungssystem “Benutzeroberfläche” verhält sich in der Multithread- und in der Singlethread-Version genau gleich. Das System “Benutzeroberfläche” ist für die Benutzeroberfläche zuständig. Im Thread der Benutzeroberfläche können zwar VIs ausgeführt werden, das Ausführungssystem wechselt aber zwischen kooperativem Multitasking und der Beantwortung von Ereignissen der Benutzeroberfläche ab.

Jedes der anderen Ausführungssysteme hat seine eigene Warteschlange. Diese Ausführungssysteme sind nicht für die Verwaltung der Benutzeroberfläche zuständig. Immer wenn ein VI in einer dieser Warteschlangen ein Bedienelement aktualisieren muß, wird die Zuständigkeit an den Thread “Benutzeroberfläche” abgegeben.

Jedes Ausführungssystem, mit Ausnahme des Systems für die Benutzeroberfläche, besitzt außerdem zwei Threads, die für die Ausführung von VIs von der Warteschlange aus zuständig sind. Jeder Thread führt einen Task aus, so daß der zweite Thread weiterhin andere VIs in diesem Ausführungssystem ausführt, falls ein VI einen CIN aufruft. Da jedes Ausführungssystem über eine begrenzte Anzahl von Threads verfügt, erhalten Sie, sobald die Threads voll sind, wie auch in einem Singlethread-System wartende Tasks.

Während VIs, die Sie schreiben, korrekt ausgeführt werden, wenn Sie sie auf das Ausführungssystem “Standard” eingestellt lassen, ist es auch in Erwägung zu ziehen, sie auf ein anderes Ausführungssystem einzustellen. Wenn Sie beispielsweise Instrumententreiber entwickeln, ist es möglicherweise ratsam, Ihre VIs auf das Ausführungssystem “Instrument I/O” einzustellen.

Selbst wenn Sie VIs auf das Ausführungssystem “Standard” eingestellt lassen, besteht ein wichtiger Vorteil von Multithreading in der Abtrennung der Benutzeroberfläche in ihren eigenen Thread. Alle in der Benutzeroberfläche durchgeführten Aktivitäten (wie z.B. das Zeichnen auf dem Frontpanel, das Antworten auf Mausclicks usw.) kann stattfinden, ohne daß dem Blockdiagramm Ausführungszeit entzogen wird. Die Anzeige umfangreicher Informationen in einem Graphen hindert das Blockdiagramm nicht an der Ausführung von Code. Ebenso hindert die Ausführung einer langen Rechenroutine die Benutzeroberfläche nicht daran, Mausclicks oder Tastaturdateneingaben zu beantworten.

Computer mit mehreren Prozessoren ziehen sogar einen größeren Nutzen aus Multithreading. Auf einem System mit einem Prozessor unterbricht das Betriebssystem die Threads und teilt allen Threads des Prozessors Zeit zu. Bei einem Computer mit mehreren Prozessoren können Threads auf mehreren Prozessoren gleichzeitig ausgeführt werden, so daß mehrere Vorgänge in der Tat gleichzeitig ablaufen können.

Die Bezeichnungen Instrument I/O, Datenerfassung usw. sind lediglich Empfehlungen für die Art von Tasks, die Sie in diese Systeme aufnehmen könnten. I/O und DAQ (Datenerfassung) funktionieren in anderen Systemen, die Verwendung dieser Label hilft Ihnen jedoch dabei, Ihre Anwendung zu unterteilen und die Organisation zu verstehen.

Andere 1 und Andere 2 sind verfügbar, wenn es in Ihrer Anwendung andere Tasks gibt, die ihren eigenen Thread erfordern.

Synchrone/blockierende Knoten

Wie bereits erwähnt, gibt es einige synchrone Knoten, d.h. sie werden nicht mit anderen Knoten in einem Multitasking-System ausgeführt. In der Multithread-Version bedeutet dies, daß sie vollständig ausgeführt werden und daß der Thread, in dem sie ablaufen, bis zum Schluß ausschließlich von diesem Task in Anspruch genommen wird.

Code Interface Nodes (CINs), DLL-Aufrufe und alle Rechenfunktionen werden synchron ausgeführt. Die meisten Analyse-VIs und Datenerfassungs-VIs enthalten CINs und werden somit synchron ausgeführt. Eine Fast Fourier Transform (FFT)-Funktion wird ohne Unterbrechung bis zum Schluß ausgeführt, ganz gleich wie lange die FFT-Ausführung dauert.

Fast alle anderen Knoten sind asynchron. Strukturen, I/O-Funktionen, Timingfunktionen und SubVIs, die Sie erstellen, sind alle asynchron.

Die Funktionen Warten, Wartet auf Occurrence, Dialogfeld, GPIB-Funktionen, VISA-Funktionen und serielle VIs warten auf das Ende des Tasks, können dies aber ohne Belastung des Threads tun. Diese Tasks werden aus der Warteschlange entfernt, bis der Task abgeschlossen ist. Wenn er abgeschlossen ist (wenn der Benutzer z.B. eine Taste in einem Dialogfeld drückt, das von der Funktion "Dialogfeld" angezeigt wird), wird der Task an das Ende der Ausführungswarteschlange gesetzt.

Festlegen von Taskprioritäten

Es gibt zwei Methoden zum Festlegen der Priorität parallel ablaufender Tasks. Die eine besteht darin, die Prioritätseinstellung in den **VI-Einstellungen** zu ändern. Die andere besteht darin, die Wartefunktionen strategisch zu nutzen.

In den meisten Fällen wird die Priorität eines VIs von der Standardeinstellung abweichend nicht geändert. Es ist durchaus möglich, daß die Steuerung der Ausführungsreihenfolge mit Hilfe von Prioritäten nicht die erwarteten Ergebnisse erzeugt. Bei einer fehlerhaften Durchführung kann es sein, daß Tasks mit einer niedrigeren Priorität dabei völlig übergangen werden.

Wartefunktionen und die Festlegung von Taskprioritäten

Mit der Wartefunktion können Sie veranlassen, daß weniger wichtige Tasks weniger häufig ausgeführt werden. Wenn mehrere Schleifen beispielsweise parallel ausgeführt werden und Sie wollen, daß sie häufiger ausgeführt werden, fügen Sie die Wartefunktionen in die Tasks mit der niedrigeren Priorität ein. Damit wird anderen Tasks mehr Zeit überlassen.

Wie schon im Abschnitt *Synchrone/blockierende Knoten* in diesem Kapitel beschrieben, wird ein Diagramm, während es sich im Wartezustand befindet, gänzlich aus der Warteschlange entfernt, so daß andere Tasks ausgeführt werden können.

Es ist insbesondere angemessen, Wartefunktionen in Schleifen zu verwenden, die Abfragen an die Benutzeroberfläche senden. Eine Wartezeit von 100 bis 200 ms fällt kaum ins Gewicht, setzt aber Ressourcen für die Anwendung frei, damit andere Tasks wirkungsvoller gehandhabt werden können. Außerdem werden Ressourcen im Betriebssystem frei, so daß dieses anderen Threads oder Anwendungen mehr Zeit widmen kann.

VI-Einstellungen und Priorität

Sie ändern die Priorität eines VIs mit dem Menüpunkt **Priorität** im Dialogfeld **VI-Einstellungen**.



Es gibt sechs Prioritätsstufen, die im folgenden angefangen bei der niedrigsten bis zur höchsten Priorität aufgelistet sind.

- Hintergrundpriorität (niedrigste)
- normale Priorität
- über normaler Priorität
- hohe Priorität
- zeitkritische Priorität (höchste)
- Subroutine

Die ersten fünf Prioritäten verhalten sich ähnlich (niedrigste bis höchste), während die Subroutinenpriorität einige zusätzliche Merkmale aufweist. Die folgenden Punkte gelten für all diese Prioritäten mit Ausnahme der Subroutinenpriorität.

Prioritäten im Thread der Benutzeroberfläche und im Singlethread-Ausführungssystem

Im Thread der Benutzeroberfläche werden Prioritätsstufen beim Singlethread- und Multithread-Ausführungssystem gleich behandelt. Die folgenden Absätze beschreiben die Art und Weise, auf die Prioritäten bei diesen Konfigurationen behandelt werden.

Im Singlethread-System und im Thread der Benutzeroberfläche eines Multithread-Systems besitzt die oben beschriebene Warteschlange des Ausführungssystems mehrere Einstiegspunkte. VIs mit höherer Priorität werden vor VIs mit niedrigerer Priorität in die Warteschlange eingereiht. Wenn ein Task mit einer hohen Priorität ausgeführt wird und die Warteschlange nur Tasks einer niedrigeren Priorität enthält, wird der Task mit der höheren Priorität weiter ausgeführt. Wenn die Ausführungswarteschlange zum Beispiel zwei VIs jeder Prioritätsstufe enthält, teilen beispielsweise zuerst die zeitkritischen VIs die Ausführungszeit bis zu ihrem Abschluß ausschließlich. Dann teilen die VIs hoher Priorität die Ausführungszeit ausschließlich, bis auch sie beendet sind usw.

Die Ausnahme hierzu tritt ein, wenn die VIs höherer Priorität eine Funktion aufrufen, die wartet, wie z.B. die Funktion "Warten (ms)" (eine Liste der wartenden asynchronen Funktionen finden Sie im Abschnitt *Synchrone/blockierende Knoten* weiter oben in diesem Kapitel). In diesem Fall werden die VIs höherer Priorität aus der Warteschlange genommen, bis die Wartezeit oder I/O abgeschlossen ist, und anderen Tasks (möglicherweise mit niedrigerer Priorität) die Ausführung zugewiesen. Wenn die Wartezeit oder I/O abgeschlossen ist, fügt das Ausführungssystem den wartenden Task vor den Tasks mit niedrigerer Priorität wieder in die Warteschlange ein.

Wenn ein VI höherer Priorität außerdem ein SubVI niedrigerer Priorität aufruft, wird dieses SubVI für die Dauer des Aufrufs auf Prioritätsstufe des Aufrufenden befördert. Folglich brauchen Sie die Prioritätsstufen der SubVIs, die ein VI aufruft, nicht zu ändern, um die Prioritätsstufe anzuheben.

Wenden Sie Prioritäten mit Vorsicht an, da Tasks mit niedrigerer Priorität leicht von Tasks höherer Priorität zeitlich bedrängt werden. Wenn die Tasks höherer Priorität so ausgelegt sind, daß sie für eine lange Zeit ausgeführt werden, werden Tasks mit einer niedrigeren Priorität nur dann ausgeführt, wenn der Task höherer Priorität gelegentlich wartet oder I/O durchführt, so daß er aus der Warteschlange entfernt wird. Ziehen Sie bei der Verwendung

von Prioritäten in Erwägung, Wartezustände in weniger zeitkritischen Abschnitten des Diagramms einzufügen, um Zeit frei zu machen.

Prioritäten in anderen Ausführungssystemen des Multithread-Systems

Weiter oben in diesem Kapitel wurden sechs Ausführungssysteme beschrieben, die dem Menüpunkt "Ausführungssystem" im Dialogfeld **VI-Einstellungen** entsprechen. Tatsächlich gibt es für jede dieser sechs Kategorien ein separates Ausführungssystem pro Prioritätsstufe (ohne die Prioritätsstufe "Subroutine" und das Ausführungssystem "Benutzeroberfläche"). Jedes dieser priorisierten Ausführungssysteme besitzt seine eigene Warteschlange und zwei Threads, die auf die Behandlung von Diagrammen in der Warteschlange ausgelegt sind.

Statt sechs Ausführungssystemen gibt es ungeachtet der Priorität 1 für das System "Benutzeroberfläche" und 25 für die anderen Systeme (5 Systeme mal 5 für jede Prioritätsstufe).

Den Threads für jedes dieser Ausführungssysteme sind Prioritätsstufen im Betriebssystem basierend auf Ihrer Klassifizierung zugewiesen. Dies bedeutet bei normaler Ausführung, daß Tasks höherer Priorität mehr Zeit eingeräumt wird als Tasks niedrigerer Priorität. Wie bei den Prioritäten im obigen Thread "Benutzeroberfläche" kann es sein, daß bei Tasks höherer Priorität Tasks niedrigerer Priorität zu kurz kommen, wenn erstere ohne Wartezeiten für längere Zeitabstände ausgeführt werden.

Manche Betriebssysteme versuchen, der Bedrängung von Tasks niedrigerer Priorität entgegenzuwirken, indem sie die Prioritätsstufe von Tasks niedrigerer Priorität anheben. Somit erhalten Tasks niedrigerer Priorität auf Betriebssystemen mit Prioritätsaufwertung Gelegenheit zur Ausführung, selbst wenn ein Task höherer Priorität ununterbrochen ausgeführt werden soll. Verlassen Sie sich jedoch nicht darauf, da dies von Betriebssystem zu Betriebssystem unterschiedlich ist und Prioritäten von Tasks und das Verhalten bei der Prioritätsaufwertung auf manchen Betriebssystemen vom Benutzer angepaßt werden.

Mit dem System "Benutzeroberfläche" ist nur ein Thread verbunden. Dieser Thread ist auf dieselbe Betriebssystempriorität eingestellt wie die normale Priorität bei anderen Ausführungssystemen. Wenn Sie also ein VI im standardmäßigen Ausführungssystem mit über normaler Priorität ausführen, kommt dabei möglicherweise die Benutzeroberfläche zu kurz. Dies kann zu einer trägen oder sogar nicht antwortenden Benutzeroberfläche führen. Wenn Sie einem VI für die Ausführung

Hintergrundpriorität zuweisen, wird es mit niedrigerer Priorität ausgeführt als der Thread “Benutzeroberfläche”.

Wie in der obigen Diskussion zur Benutzeroberfläche hebt das Ausführungssystem die SubVI-Priorität für die Dauer des Aufrufs auf dieselbe Stufe des Aufrufenden an, wenn ein VI ein SubVI einer niedrigeren Priorität aufruft. Wenn einem VI und seinem SubVI in den VI-Einstellungen dasselbe Ausführungssystem zugewiesen ist, bleibt der Aufruf an das SubVI niedrigerer Priorität somit im selben priorisierten Ausführungssystem. Wenn das VI jedoch ein SubVI höherer Priorität aufruft, so wird dieser SubVI-Aufruf in einem anderen Ausführungssystem höherer Priorität ausgeführt.

Prioritätsstufe “Subroutine”

Wenn Sie ein VI auf die Priorität “Subroutine” (die höchste Stufe) einstellen, ist das Verhalten etwas anders. Die Motivation für die Prioritätsstufe “Subroutine” besteht darin, einem VI eine möglichst effektive Ausführung zu gestatten. Der Compiler kompiliert VIs der Priorität “Subroutine” so, daß sie keine Zeit mit anderen VIs teilen müssen.

Wenn ein VI mit der Priorität “Subroutine” ausgeführt wird, übernimmt es effektiv die Kontrolle über den Thread, in dem es ausgeführt wird (und es wird in demselben Thread ausgeführt wie der Aufrufer). Kein anderes VI kann in diesem Thread ausgeführt werden, bis die Ausführung des VIs mit der Priorität “Subroutine” abgeschlossen ist, selbst wenn das andere VI auch als Subroutine markiert ist. Im Singlethread-Ausführungssystem bedeutet dies, daß kein anderes VI ausgeführt wird. In Ausführungssystemen, die vom System “Benutzeroberfläche” abweichen, behandelt der Thread, in dem die Subroutine ausgeführt wird, keine anderen VIs, aber der zweite Thread des Ausführungssystems sowie andere Ausführungssysteme können weiterhin VIs ausführen.

Zusätzlich dazu, daß die Zeit nicht mit anderen VIs geteilt wird, ist die VI-Ausführung bei der Priorität “Subroutine” so rationalisiert, daß Bedien- und Anzeigeelemente eines Frontpanels nicht aktualisiert werden, wenn die Subroutine aufgerufen wird. Die Beobachtung eines VI-Frontpanels mit der Priorität “Subroutine” gibt keinerlei Aufschluß über seine Ausführung.

Ein VI mit der Priorität “Subroutine” kann andere VIs mit derselben Priorität aufrufen, es kann jedoch kein VI mit einer anderen Priorität aufrufen. Verwenden Sie die Priorität “Subroutine” in Situationen, in denen Sie den allgemeinen Overhead in einem SubVI minimieren wollen, das einfache Berechnungen durchführt.

Da Subroutinen nicht auf Interaktion mit der Ausführungswarteschlange ausgelegt sind, können sie auch keine Funktion aufrufen, die sie normalerweise aus der Warteschlange entfernt. Dies bedeutet, daß sie keine Wartefunktionen, I/O-Funktionen oder Dialogfeld-Funktionen aufrufen können.

Subroutinen verfügen über eine zusätzliche Option, die ihnen bei zeitkritischen Anwendungen helfen kann. Wenn Sie das Popup-Menü eines SubVIs aufrufen und Subroutinenaufruf übergehen, wenn beschäftigt wählen, überspringt das Ausführungssystem den Aufruf, wenn die Subroutine gegenwärtig in einem anderen Thread ausgeführt wird. Dies kann in zeitkritischen Schleifen von Vorteil sein, bei denen Operationen, die die Subroutine durchführt, risikolos übersprungen werden können und Sie die Verzögerung vermeiden wollen, die durch das Warten auf die Beendigung eines SubVIs entsteht. Wenn Sie die Ausführung eines SubVIs überspringen wollen, werden alle Ausgänge des SubVIs auf den Standardwert des betreffenden Datentyps eingestellt. Numerische Ausgänge werden also auf Null gesetzt, String- und Arrayausgänge sind leer und boolesche Ausgänge werden auf FALSE gesetzt. Beachten Sie, daß dies der Standard für diesen Typ ist, nicht der Standardwert für das Anzeigeelement auf dem SubVI-Frontpanel. Wenn Sie feststellen wollen, ob eine Subroutine auch tatsächlich ausgeführt wird, schreiben Sie vor, daß sie TRUE zurückgibt, wenn sie erfolgreich ausgeführt worden ist. Andernfalls wird automatisch der Standardwert FALSE zurückgegeben.

Ablaufinvariante Ausführung

Unter normalen Umständen kann das Ausführungssystem nicht gleichzeitig mehrere Aufrufe an dasselbe SubVI ausführen. Wenn Sie versuchen, ein SubVI aufzurufen, das von mehreren Stellen aus nicht ablaufinvariant ist, wird ein Aufruf ausgeführt, und der andere Aufruf wartet mit der Ausführung, bis der erste beendet ist. Wenn Sie ein VI ablaufinvariant machen (mit den VI-Einstellungen), pflegt jede Instanz des Aufrufs ihren eigenen Informationszustand. Das Ausführungssystem führt dann dasselbe SubVI gleichzeitig von mehreren Orten aus. Die ablaufinvariante Ausführung ist in den folgenden Situationen nützlich.

- Wenn ein VI einen angegebenen Zeitraum lang oder bis zum Timeout wartet.
- Wenn ein VI Daten enthält, die im Gegensatz zu einer globalen Variablen, bei der es sich um ein VI handelt, dessen Daten Sie freigeben wollen, nicht von mehreren Instanzen desselben VI geteilt werden sollen.

Sie beginnen die ablaufinvariante Ausführung über die **Ausführungsoptionen** im Dialogfeld **VI-Einstellungen**. Wenn Sie Ablaufinvarianz wählen, sind mehrere andere Menüpunkte nicht verfügbar.



Beachten Sie, daß die folgenden Optionen nicht verfügbar sind, wenn **Ablaufinvariante Ausführung** gewählt ist.

- Frontpanel beim Laden öffnen
- Bei Aufruf Frontpanel öffnen
- Highlight-Funktion
- Einzelschrittausführung

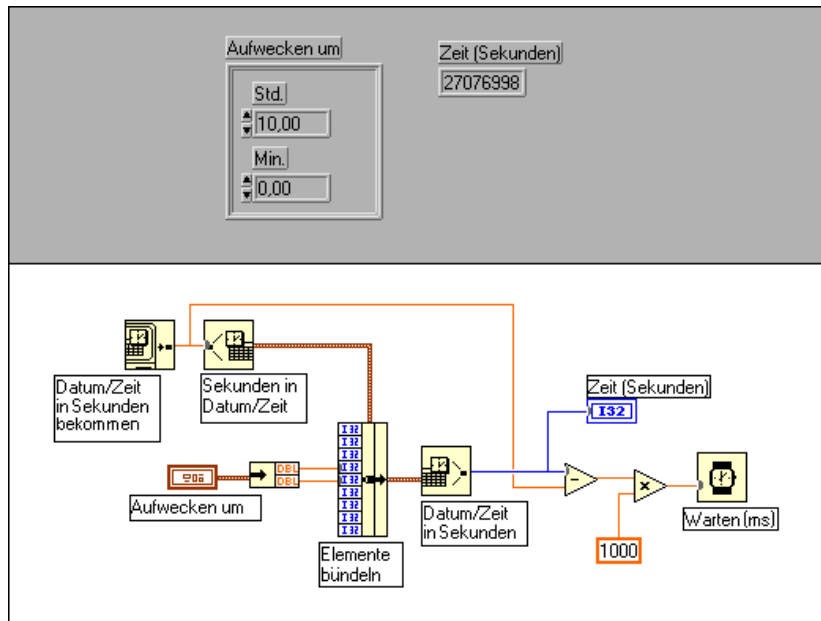
Diese Menüpunkte sind deaktiviert, da das SubVI bei jedem Aufruf zwischen verschiedenen Kopien der Daten und verschiedenen Ausführungszuständen wechseln muß, was es unmöglich macht, ständig den aktuellen Zustand anzuzeigen.

Anwendungsbeispiele der Ablaufinvarianz

Die Beispiele in den nächsten zwei Abschnitten veranschaulichen ablaufinvariante VIs.

Verwendung eines VIs mit Wartefunktion

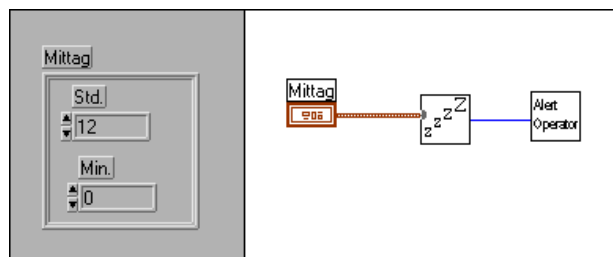
Das folgende ablaufinvariante Beispiel beschreibt ein VI namens Schläfchen, das Stunden und Minuten als Eingabe nimmt und wartet, bis diese Zeit eintrifft. Wenn Sie dies gleichzeitig an mehreren Orten verwenden wollen, muß das VI ablaufinvariant sein.



Die Funktion “Zeit in Sekunden abrufen” liest die aktuelle Zeit in Sekunden und die “Sekunden bis Datum/Uhrzeit” und konvertiert diesen Wert in einen Cluster aus Zeitwerten (Jahr, Monat, Tag, Stunde, Minute, Sekunde und Wochentag). Eine Bündelungsfunktion ersetzt die aktuelle Stunde und Minute mit Werten, die eine spätere Zeit desselben Tages darstellt, im Cluster-Bedienelement des Frontpanels “Aufwecken um”. Der angepaßte Datensatz wird dann wieder in Sekunden umgewandelt, und die Differenz zwischen der aktuellen Zeit in Sekunden und der Zeit in der Zukunft wird mit 1.000 multipliziert, um Millisekunden zu erhalten. Das Ergebnis wird an eine Wartefunktion übergeben.

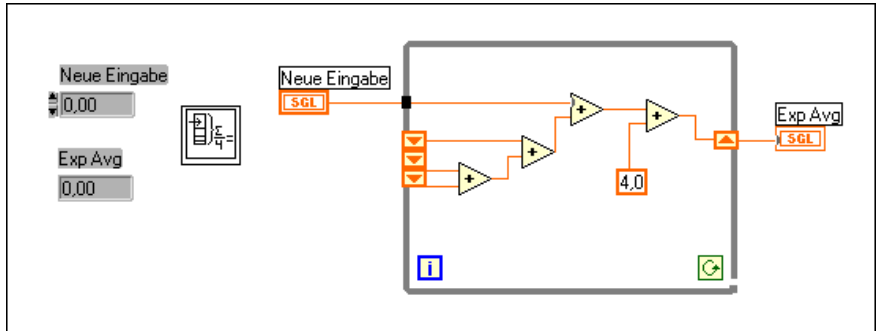
“Mittag” und “Vesper” sind zwei VIs, die “Schläfchen” als SubVI verwenden. Das VI “Mittag”, dessen Frontpanel und Blockdiagramm in der nachstehenden Abbildung zu sehen sind, wartet bis 12 Uhr und blendet ein Frontpanel ein, das den Bediener daran erinnert, Mittagspause zu machen. Das VI “Vesper” ruft ein Panel auf, das den Bediener daran erinnert, um 10 Uhr Vesper zu machen. Das VI “Vesper” und das VI “Mittag” sind mit Ausnahme der verschiedenen Pop-up-SubVIs identisch.

Damit “Mittag” und “Vesper” parallel ausgeführt werden können, muß “Schläfchen” ablaufinvariant sein. Wenn Sie andernfalls “Mittag” zuerst starten, wartet “Vesper” solange, bis “Schläfchen” um 12 Uhr aufwacht, was zwei Stunden zu spät ist.



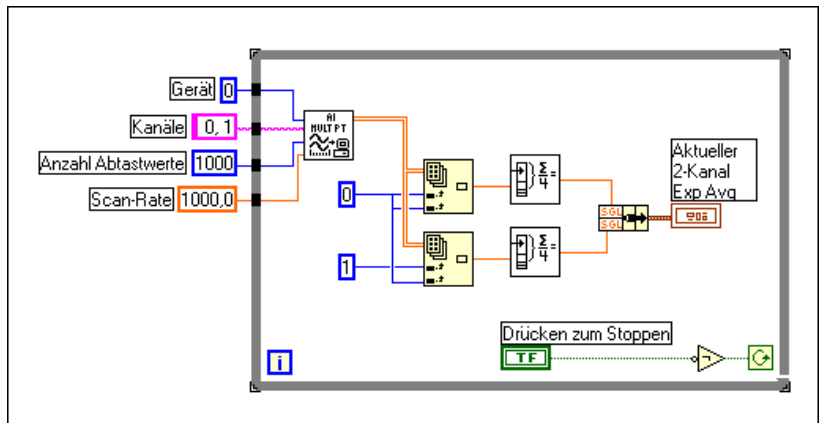
Verwendung eines Speicher-VIs, das nicht auf die Freigabe von Daten ausgelegt ist

Eine andere Situation, die Ablaufinvarianz erfordert, liegt dann vor, wenn Sie mehrere Aufrufe an ein SubVI durchführen, das Daten speichert. Angenommen, Sie erstellen ein SubVI, ExpAvg, das einen fortlaufenden exponentiellen Durchschnitt von vier Datenpunkten berechnet. Um sich die vorherigen Werte zu merken, benutzt ExpAvg ein nicht initialisiertes Schieberegister mit drei linken Terminals. Wenn Sie LabVIEW-Benutzer sind, finden Sie im *LabVIEW Benutzerhandbuch* in Kapitel 3, *Schleifen und Diagramme*, weitere Informationen über nicht initialisierte Schieberegister.



Wenn Sie BridgeVIEW-Benutzer sind, konsultieren Sie Kapitel 10, *Loops and Charts*, im *BridgeVIEW User Manual*.

Nehmen wir als nächstes an, daß ein VI ExpAvg zur Berechnung eines fortlaufenden Durchschnitts zweier Datenerfassungskanäle verwendet wird. Sie überwachen z.B. die Spannungen an zwei Punkten in einem Prozeß und wollen den exponentiellen laufenden Durchschnitt auf einem Streifendiagramm anzeigen. Das Diagramm enthält zwei ExpAvg-Knoten. Die Aufrufe wechseln sich ab, einer für Kanal 0, dann einer für Kanal 1. Nehmen wir an, daß Kanal 0 zuerst ausgeführt wird. Wenn ExpAvg nicht ablaufinvariant ist, verwendet der Aufruf für Kanal 1 den Durchschnitt, der vom Aufruf für Kanal 0 berechnet wurde, und der Aufruf für Kanal 0 verwendet den Durchschnitt, der vom Aufruf für Kanal 1 berechnet wurde. Wenn ExpAvg ablaufinvariant ist, kann jeder Aufruf ohne Risiko, daß die Daten geteilt werden, unabhängig ausgeführt werden.



Abstimmung des Zugriffs auf globale und lokale Variablen und externe Ressourcen

Da das Ausführungssystem mehrere Tasks parallel ausführen kann, müssen Sie sicherstellen, daß auf globale und lokale Variablen und Ressourcen in der passenden Reihenfolge zugegriffen wird.

Wettbewerbsbedingungen

Eine *Wettbewerbsbedingung* tritt ein, wenn mehrere Code-Teile, die parallel ausgeführt werden, den Wert derselben freigegebenen Ressource, gewöhnlich einer globalen oder lokalen Variable, ändern. Die folgenden zwei Diagramme sind ein Beispiel für eine Wettbewerbsbedingung.

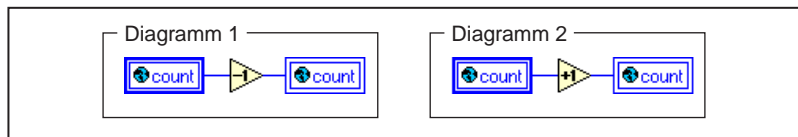


Diagramm 1 vermindert den Wert des Zählers in Inkrementen, und Diagramm 2 erhöht den Wert des Zählers in Inkrementen. Da zwischen den zwei Diagrammen keine Datenabhängigkeit besteht, ist bei der Ausführung der beiden Diagramme der folgende Ablauf möglich (angenommen, daß der Anfangswert des Zählers 4 ist).

Diagramm 1: Zähler lesen (4)

Diagramm 2: Zähler lesen (4)

Diagramm 1: 4 inkrementieren und Zähler schreiben (Zähler ist jetzt 5)

Diagramm 2: 4 dekrementieren und Zähler schreiben (Zähler ist jetzt 3)

Obwohl Sie erwarten könnten, daß bei der Ausführung der beiden Diagramme der Zähler in Inkrementen erhöht und anschließend vermindert wird, wodurch der Zähler effektiv auf den ursprünglichen Wert zurückgesetzt würde, kann die Wettbewerbsbedingung dazu führen, daß nur eine der Operationen wirksam ist.

Wettbewerbsbedingungen lassen sich auf mehrere Arten vermeiden. Die einfachste davon besteht darin, in der ganzen Anwendung nur eine Stelle zuzulassen, an der eine globale Variable geändert werden kann. Im vorherigen Beispiel rufen alle Diagramme, die den Zähler einer globalen Variable in Inkrementen erhöhen oder vermindern, ein allgemeines SubVI, Zähler ändern, auf. Sie übergeben einen booleschen Parameter, um anzugeben, ob die globale Variable in Inkrementen erhöht oder vermindert werden soll.

Da ein nicht ablaufinvariantes VI nur für ein aufrufendes VI auf einmal ausgeführt wird und da “Zähler ändern” das einzige VI in der Anwendung ist, das den globalen Zähler ändert, wird hierdurch die Wettbewerbsbedingung vermieden.



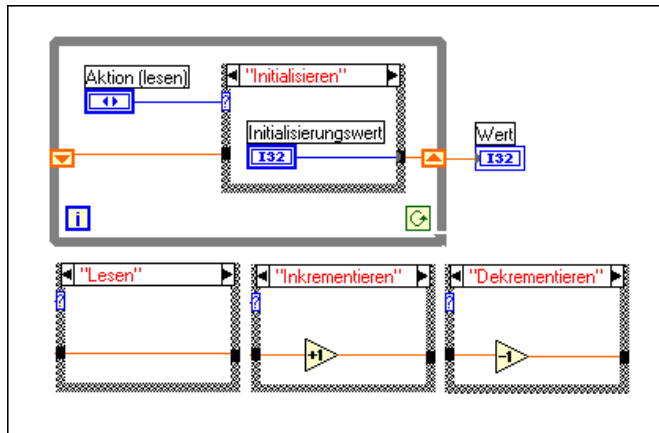
Hinweis

In der Singlethread-Umgebung ist es möglich, ein VI mit Subroutinenpriorität zu verwenden, um eine globale Variable zu lesen/modifizieren/schreiben, ohne einen Wettbewerbszustand auszulösen. Dies liegt daran, daß ein VI mit einer Subroutinenpriorität den Ausführungs-Thread nicht mit anderen VIs teilt. In der Multithread-Umgebung garantiert die Subroutinenpriorität keinen exklusiven Zugriff auf eine globale Variable, da ein anderes VI, das in einem anderen Thread ausgeführt wird, unter Umständen zur gleichen Zeit auf die globale Variable zugreift.

Funktionale globale Variablen

Eine andere Möglichkeit, Wettbewerbsbedingungen in Verbindung mit globalen Variablen zu vermeiden, besteht darin, überhaupt keine globalen Variablen zu verwenden, sondern statt dessen VIs zu verwenden, die zur Aufnahme globaler Daten Schleifen mit nicht initialisierten Schieberegistern verwenden. Solche VIs werden allgemein als funktionale globale Variable oder LabVIEW 2-artige Globals bezeichnet (globale Variable gab es in LabVIEW 2 noch nicht, es wurden ausschließlich funktionale globale Variable verwendet). Eine funktionale globale Variable besitzt gewöhnlich einen Aktionseingangsparameter, der angibt, welche Funktion das VI durchführt. Das VI verwendet ein nicht initialisiertes Schieberegister in einer While-Schleife, um das Ergebnis der

Operation festzuhalten. Die folgende Abbildung zeigt ein Diagramm einer funktionalen globalen Variable, die eine einfache globale Zählervariable durchführt. Die Aktionen in diesem Beispiel sind **initialisieren**, **lesen**, **inkrementieren** und **dekrementieren**.



Jedesmal, wenn das VI aufgerufen wird, wird das Diagramm in der Schleife genau einmal ausgeführt. Je nach dem Parameter **Aktion** wird der Wert des Schieberegisters durch den Case in der Schleife entweder initialisiert, nicht geändert oder er nimmt in Inkrementen zu oder ab.

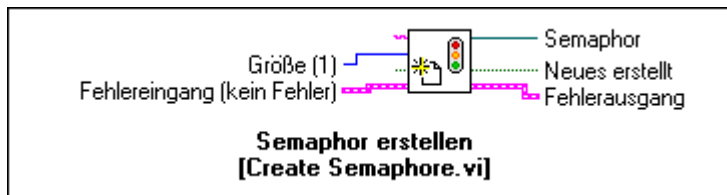
Obleich funktionale globale Variablen zur Durchführung einfacher globaler Variablen verwendet werden können, wie im Beispiel oben zu sehen ist, sind sie insbesondere bei der Implementierung komplexerer Datenstrukturen, wie z.B. einem Stapel- oder Warteschlangenpuffer, nützlich. Funktionale globale Variablen werden auch dazu verwendet, den Zugriff auf globale Ressourcen, wie z.B. Dateien, Instrumente und DAQ-Karten, der nicht mit einer globalen Variable dargestellt werden kann, zu sichern.

Semaphore

Die meisten Abstimmungsprobleme in Verbindung mit globalen Ressourcen lassen sich mit funktionalen globalen Variablen leicht lösen, da das VI der funktionalen globalen Variablen sicherstellt, daß die Daten, die es enthält, nicht jeweils für einen Aufrufer geändert werden. Ein Nachteil von funktionalen globalen Variablen besteht darin, daß das zugehörige VI geändert und eine neue Aktion hinzugefügt werden muß, wenn die Ressource, die sie beinhalten, auf eine neue Art und Weise geändert werden muß. In manchen Anwendungen, in denen sich der Gebrauch der globalen Ressourcen häufig ändert, kann sich dies als unpraktisch erweisen. Gestalten Sie in solchen Fällen Ihre Anwendung so, daß sie einen Semaphor verwendet, um den Zugriff auf die globale Ressource zu sichern.

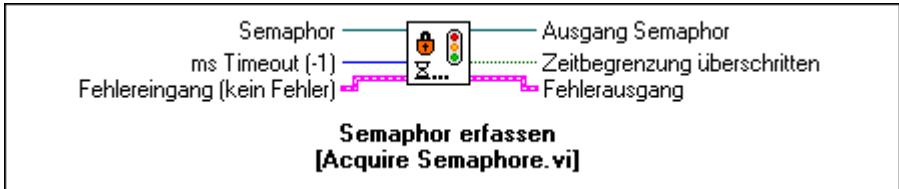
Ein Semaphor, der auch als Mutex bezeichnet wird, ist ein Objekt, das dazu verwendet wird, den Zugriff auf freigegebene Ressourcen zu sichern. Der Code, bei dem auf freigegebene Ressourcen zugegriffen wird, wird oftmals als kritischer Abschnitt bezeichnet. Im allgemeinen sollten Sie darauf achten, daß sich immer nur ein Task im kritischen Abschnitt befindet, der durch einen allgemeinen Semaphor gesichert wird. Es ist durchaus möglich, daß Semaphore mehrere Tasks (bis zu einem vordefinierten Maximum) in einem kritischen Abschnitt zulassen.

Ein neuer Semaphor wird mit dem VI **Semaphore erstellen** erstellt (zu finden in der Palette **Funktionen»Fortgeschritten»Synchronisation»Semaphore**). Das VI nimmt die Anfangsgröße des Semaphors als Eingang. Die Größe bestimmt, wie viele verschiedene Tasks den Semaphor auf einmal benutzen. Jedesmal, wenn ein Task damit beginnt, den Semaphor zu verwenden, wird die Größe des Semaphors in Inkrementen vermindert. Wenn die Größe den Wert Null erreicht, muß jeder Task, der versucht, den Semaphor zu verwenden, solange warten, bis der Semaphor von einem anderen Task freigegeben wird.



Ein Task zeigt mit dem Aufruf des VIs **Semaphore erfassen** an, daß er einen Semaphor verwenden will. Wenn die Größe des Semaphors größer als Null ist, gibt das VI sofort `Timed Out = FALSE` zurück, und der Task wird fortgesetzt. Ist die Größe des Semaphors Null, wartet der Task, bis der Semaphor zur Verfügung steht oder bis die angegebene Wartezeit

abgelaufen ist. Gibt das VI den Wert Timed Out = TRUE zurück, zeigt es an, daß der Semaphor nicht erfaßt wurde, und die Tasks führen den kritischen Abschnitt nicht aus. Wenn das VI den Wert Timed Out = FALSE zurückgibt, bedeutet dies, daß der Semaphor erfolgreich erfaßt wurde.



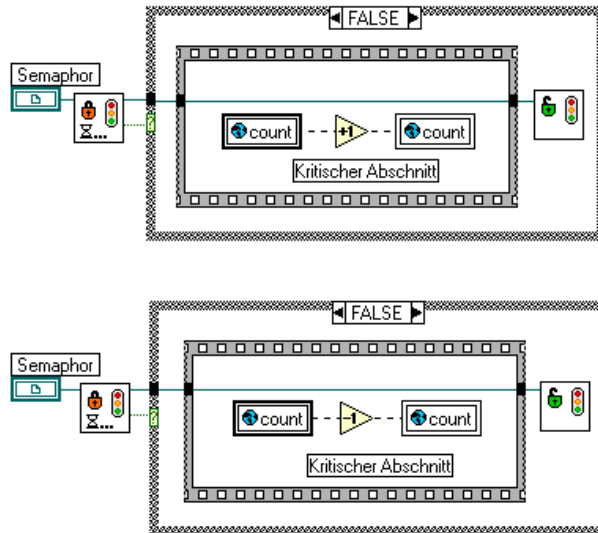
Wenn ein Task einen Semaphor erfolgreich erfaßt und seinen kritischen Abschnitt erfolgreich beendet, wird der Semaphor durch den Aufruf des VIs **Semaphor freigeben** freigegeben. Warten bei der Freigabe des Semaphors irgendwelche Tasks darauf, den Semaphor zu erfassen, so ist der erste Task in der Lage, die Ausführung fortzusetzen. Andernfalls erhöht sich die Größe des Semaphors inkremental.



Wenn ein Semaphor nicht mehr benötigt wird, wird er zerstört, indem das VI "Semaphore zerstören" aufgerufen wird. Gibt es VIs vom Typ "Semaphore erfassen" VIs, die auf den Semaphor warten, geben diese sofort das Signal "Timed Out = TRUE" sowie einen Fehler zurück.



Das folgende Beispiel zeigt die Verwendung eines Semaphors zum Schutz der kritischen Abschnitte (erhöht oder vermindert eine globale Variable in Inkrementen). Der Semaphor wurde dadurch erstellt, daß die Größe 1 an das VI **Semaphore erstellen** übergeben wurde.



Jedes Diagramm, das daran geht, einen kritischen Abschnitt auszuführen, muß zuerst das VI “Semaphore erfassen” aufrufen. Wenn der Semaphore beschäftigt ist (oder die Größe Null aufweist), wartet das VI, bis der Semaphore frei wird. Wenn das VI “Semaphore erfassen” Timed Out = FALSE zurückgibt, womit angegeben wird, daß der Semaphore erfaßt wurde, beginnt das Diagramm mit der Ausführung des Cases FALSE. Wenn das Diagramm seinen kritischen Abschnitt beendet hat (Sequenzrahmen), wird der Semaphore freigegeben, wodurch es einem anderen Diagramm gestattet wird, die Ausführung fortzusetzen.

Andere Abstimmungsfunktionen

Außer den Semaphore-VIs gibt es eine Reihe von zusätzlichen Funktionen und VIs zur Abstimmung paralleler Tasks. All diese Funktionen sind in der Palette **Funktionen»Fortgeschritten»Synchronisation»Semaphore** zu finden.

Occurrence-Funktionen — Occurrence-Funktionen geben Ihnen eine Möglichkeit an die Hand, einen oder mehrere Tasks warten zu lassen, bis sie von einem anderen Task eine Benachrichtigung erhalten. Sie verwenden die Funktion “Occurrence erzeugen” dazu, eine Occurrence zu erstellen, die Sie entweder an eine Funktion “Auf Occurrence warten” oder “Occurrence setzen” übergeben.

Notification-VIs — Occurrences ähnlich, aber mit dem Zusatz, daß Sie, wenn Sie die Benachrichtigung senden, eine Nachricht beifügen. Wenn

mehrere VIs auf dieselbe Benachrichtigung warten, erhalten sie alle diese Nachricht. Im Gegensatz zu Occurrences können Sie eine Benachrichtigung außerdem abbrechen.

Queue-VIs — Wie Benachrichtigungen erlauben Ihnen Warteschlangen, Nachrichten an andere Tasks zu senden. Mehrere Nachrichten können jedoch in eine Warteschlange eingereiht werden, so daß ein Listener, wenn er eine Nachricht abrufen, die älteste zuerst erhält. Außerdem ist die Verwendung von Warteschlangen-VIs in Situationen mit einem einzigen Listener am sinnvollsten. Wenn ein Listener ein Element aus der Warteschlange entfernt, bekommen andere Listener dieses Element nicht zu sehen.

Mit `enqueue.vi` kann einer Warteschlange mittels eines VIs eine Nachricht hinzugefügt werden. Um die älteste Nachricht aus der Warteschlange abzurufen, verwenden Sie `dequeue.vi`. Wenn Sie eine Warteschlange mit "Open Queue Reference" erstellen, geben Sie an, ob sie eine begrenzte Größe hat oder nicht; wenn Sie eine begrenzte Warteschlange erstellen und versuchen, mehr Daten in die Warteschlange einzureihen als diese aufnehmen kann, wartet die Operation `enqueue.vi`, bis ein Listener Daten mit `dequeue.vi` abrufen, bevor das neue Element hinzugefügt wird.

Rendezvous-VIs — Sie können die Rendezvous-VIs in Situationen verwenden, in denen mehrere parallele Tasks an einem gemeinsamen Punkt abgestimmt werden müssen. Jeder Task, der das Rendezvous erreicht, wartet, bis die angegebene Anzahl von Tasks wartet. Zu diesem Zeitpunkt wird die Ausführung aller Tasks fortgesetzt.

Allgemeine Empfehlungen zur Verwendung von Ausführungssystemen und Prioritäten

Im folgenden erhalten Sie eine Zusammenfassung einiger allgemeiner Empfehlungen zur Verwendung der in diesem Kapitel beschriebenen Optionen von Ausführungssystemen.

Während einiges hiervon ziemlich kompliziert erscheint, ist es wichtig zu wissen, daß es in den meisten Anwendungen nicht erforderlich ist, Prioritätsstufen oder alternative Ausführungssysteme zu verwenden. Das Ausführungssystem bewältigt das Multitasking der VIs untereinander automatisch.

Standardmäßig werden alle VIs so eingestellt, daß sie im Standard-Ausführungssystem und mit normaler Priorität ausgeführt werden. Im Multithread-System werden Aktivitäten der Benutzeroberfläche von einem

separaten Thread gehandhabt, so daß Ihre VIs von der Interaktion der Benutzeroberfläche isoliert werden. Selbst wenn Sie ein Singlethread-System verwenden, versucht dieses, die Handhabung der Interaktion mit der Benutzeroberfläche mit der Ausführung der VIs zu verknüpfen, wodurch Sie ähnliche Ergebnisse erzielen.

Im allgemeinen besteht die einfachste und sicherste Art und Weise, die Ausführung zu priorisieren, in der Verwendung von Wartefunktionen, um so Schleifen mit einer niedrigeren Priorität in Ihrer Anwendung zu verzögern. Dies ist insbesondere bei Schleifen für VIs der Benutzeroberfläche nützlich, da Verzögerungen von 100 bis 200 Millisekunden für Benutzer kaum ins Gewicht fallen.

Wenn Sie Prioritäten verwenden, gehen Sie dabei vorsichtig vor. Wenn Sie VIs einer höheren Priorität verfassen, deren Ausführung eine Weile dauert, überlegen Sie sich, ob Sie in weniger zeitkritischen Abschnitten dieser VIs nicht Wartepausen einfügen wollen, damit sie sich die Zeit mit Tasks niedrigerer Priorität teilen.

Seien Sie vorsichtig, wenn Sie globale oder lokale Variablen oder andere externe Ressourcen manipulieren, die möglicherweise von anderen Tasks manipuliert werden könnten. Wenden Sie eines der weiter oben in diesem Kapitel beschriebenen Abstimmungsverfahren an, um den Zugriff auf diese Ressourcen zu schützen.

Die Durchführung der Verwendung von Prioritäten in Singlethread- und Multithread-Systemen führt bei denselben VIs unter beiden Systemen zu einigermaßen vergleichbaren Ergebnissen. Es sind jedoch geringfügige Timing-Unterschiede möglich. Wenn Sie VIs an Kunden verteilen, die ein anderes Betriebssystem ausführen, empfiehlt es sich möglicherweise, Ihre Anwendungen unter derartigen Bedingungen auszuprobieren. Wenn Sie das Multithread-System verwenden, können Sie es so einrichten, daß es sich wie ein Singlethread-System verhält, indem Sie die Voreinstellung **Mit multiplen Threads ausführen** im Abschnitt **Leistung und Speicher** des Dialogfeldes **Voreinstellungen** ausschalten.

Verwalten Ihrer Anwendungen

In diesem Kapitel wird beschrieben, wie Sie Dateien in Ihrer G-Anwendung verwalten können.

Dateianordnung unter Verwendung von VI-Bibliotheken

Mehrere VI-Dateien können zusammengruppiert werden, indem sie als VI-Bibliotheken gespeichert werden (standardmäßig enden die Namen dieser Dateien mit der Erweiterung `.lib`). Oder Sie können die VIs als einzelne Dateien speichern und sie in Verzeichnissen zusammenfassen. Ein Vergleich dieser beiden Methoden wird im Abschnitt *Speichern von VIs* im Kapitel 2, *Bearbeiten von VIs*, gegeben.

Wenn Sie VI-Bibliotheken verwenden, ist es eventuell ratsam, Ihre Anwendung in mehrere VI-Bibliotheken zu unterteilen. Die VIs der oberen Ebene können in einer VI-Bibliothek zusammengefaßt werden, und weitere VI-Bibliotheken können für VIs erstellt werden, die nach Funktionen unterteilt sind. Die Verwaltung der VIs läßt sich leichter bewerkstelligen, wenn Sie Ihre Dateien auf diese Weise organisieren.

Ein weiterer Grund dafür, Ihre VIs in mehrere VI-Bibliotheken zu unterteilen, liegt darin, daß das Speichern von Dateien umso länger dauert, je größer Ihre VI-Bibliothek wird. Wenn Sie ein VI in einer VI-Bibliothek speichern, wird die Bibliothek in das temporäre Verzeichnis kopiert, dann wird das VI komprimiert und in der VI-Bibliothek gespeichert, und die VI-Bibliothek wird wieder zurück in das ursprüngliche Verzeichnis kopiert. Dieses Kopierverfahren ist eine Schutzmaßnahme, um die versehentliche Zerstörung der Original-Datei zu verhindern; es dauert allerdings dann auch länger, VIs zu speichern.

In der Regel gilt, daß man es vermeiden soll, VI-Bibliotheken zu erstellen, die größer sind als 1 MB. Beachten Sie, daß es für die Größe einer Bibliothek eigentlich keine Beschränkung gibt. Die Zeit zum Laden von VIs aus VI-Bibliotheken ändert sich kaum, ganz egal wie groß die VI-Bibliothek ist. Zum Speichern von VIs in VI-Bibliotheken wird jedoch umso mehr Zeit in Anspruch genommen, je größer die Bibliothek wird.

Eine andere Methode, mit der das Speichern von VIs beschleunigt werden kann, und zwar unabhängig davon, ob sie sich in VI-Bibliotheken befinden, besteht darin sicherzustellen, daß sich Ihr temporäres Verzeichnis auf demselben Laufwerk (in UNIX auf derselben Partition) befindet wie Ihre VIs. Die Dateien können schneller kopiert werden, wenn sich Quelle und Ziel auf demselben Laufwerk (bzw. derselben Partition) befinden.

Wenn Sie VI-Bibliotheken verwenden, können Sie Top-Level-VIs hervorheben, indem Sie **Datei»VI-Bibliothek bearbeiten...** aus der Menüoption verwenden, um bestimmte VIs als Top-Level-VIs zu markieren. G listet diese Dateien am Anfang der Dateiliste im Dateidialogfeld auf. Für Verzeichnisse gibt es keine dementsprechende Option, aber Sie können Top-Level-VIs dadurch hervorheben, indem Sie SubVIs in Unterverzeichnisse ablegen.

Sichern von Dateien

Pannen kommen immer wieder vor. Festplatten stürzen ab, und Dateisysteme können beschädigt werden. Es ist ratsam, häufig Sicherungskopien Ihrer VIs anzufertigen, vielleicht einmal pro Tag. Kopieren Sie die VIs auf ein anderes Laufwerk oder auf ein Band.

Hier ein Beispiel für Probleme, die auftreten können: ein Benutzer beschädigte versehentlich die VI-Bibliothek, die in mehreren Arbeitswochen erstellt wurde. Bei der Aufforderung, ein Ziel für eine Datendatei anzugeben, hatte der Benutzer seine VI-Bibliothek spezifiziert und überschrieb anschließend diese VI-Bibliothek mit anderen Daten. Glücklicherweise war er in der Lage, die meisten Dateien seiner VI-Bibliothek wiederherzustellen, aber nur deshalb, weil er seine Dateien zuvor gesichert hatte.

Verteilung von VIs

Wenn Sie Ihre VIs an andere Computer oder Benutzer verteilen möchten, müssen Sie sich überlegen, ob Sie den bearbeitbaren Blockdiagramm-Quellcode verteilen möchten oder ob Sie das Blockdiagramm verbergen oder entfernen wollen. VIs können ohne die dazugehörigen Blockdiagramme gespeichert werden. Dadurch wird die Größe der Datei auf dem Speichermedium reduziert, und niemand kann versehentlich den Quellcode ändern; die Benutzer werden jedoch auch daran gehindert, das VI auf eine andere Plattform zu übertragen oder das VI mit einer zukünftigen höheren Version der Entwicklungsumgebung zu aktualisieren. Wenn Sie diese Funktionen als wichtig erachten, gibt es die Möglichkeit, Ihre Diagramme durch Paßwörter zu schützen. Der Quellcode bleibt weiterhin verfügbar, kann jedoch nur betrachtet und geändert werden, wenn Sie ein Paßwort eingeben.

Mit dem Dialogfeld **Mit Optionen speichern...** können Sie Diagramme aus Ihren VIs entfernen. Wenn Sie Ihre VIs ohne ein Diagramm speichern, kann das VI von anderen Benutzern nicht geändert werden. Wenn Sie Ihre VIs ohne Diagramme speichern, müssen Sie darauf achten, daß die Originalversionen nicht überschrieben werden.



Hinweis

VIs ohne Diagramme können nicht in zukünftige Versionen der Anwendung bzw. in andere Plattformen konvertiert werden. Während der Konvertierung müssen Sie Ihre VIs neu kompilieren. Ohne ein Blockdiagramm kann ein VI nicht neu kompiliert werden.

Darüber hinaus muß die Umgebung berücksichtigt werden, in welcher die Benutzer Ihre VIs ausführen. Möchten Sie Endbenutzern ein Entwicklungssystem oder eine Run-Time-Anwendung an die Hand geben?

Eine Run-Time-Anwendung ist dann angebracht, wenn nicht erwartet wird, daß Benutzer Änderungen an den VIs vornehmen. Eine Run-Time-Anwendung enthält vereinfachte Menüs, und Benutzer können die VIs nicht bearbeiten bzw. keine Blockdiagramme betrachten.

Run-Time-Anwendungen werden mit den LabVIEW Application Builder-Bibliotheken erstellt. Wenn Sie eine Run-Time-Anwendung mit diesen Bibliotheken erstellen, müssen Sie sich folgende drei Fragen stellen.

- Möchte ich eine VI-Bibliothek in die Anwendung einbetten?
- Soll die Anwendung den Menüpunkt **Öffnen...** enthalten?
- Soll die Anwendung den Menüpunkt **Beenden** enthalten?

Wenn Sie eine VI-Bibliothek in die Anwendung einbetten möchten, wird die Bibliothek mit einer Run-Time-Engine in eine Datei zusammengefaßt. Wenn diese Datei gestartet wird, werden automatisch alle Top-Level VIs in der Bibliothek geöffnet. Wenn Sie keine VI-Bibliothek einbetten, kann die Anwendung bei ihrem Start ein beliebiges VI öffnen, falls das VI mit einem Entwicklungssystem für diese Plattform gespeichert wurde.

Wenn der Menüpunkt **Öffnen...** aktiviert wird, können Sie die Anwendung zum Öffnen und Ausführen eines beliebigen VIs im Dateisystem verwenden, und zwar unabhängig davon, ob die Anwendung über eine eingebettete VI-Bibliothek verfügt oder nicht. Durch das Einbetten einer VI-Bibliothek können Sie eine völlig unabhängige Anwendung erstellen, bei der der Benutzer bzw. der Kunde daran gehindert wird, auf die Quell-VIs zuzugreifen, wenn er über das Entwicklungssystem verfügt.

Wenn mehrere VI-Suites an denselben Kunden geschickt werden sollen, ist eine einzelne Run-Time-Anwendung mit dem Menüpunkt **Öffnen...** geeigneter als separate, eingebettete Anwendungen, weil jede eingebettete Anwendung über eine Kopie des Run-Time-Codes verfügt, der ca. 2 bis 3 MB umfaßt.

Zum Erstellen einer Run-Time-Anwendung müssen Sie über die Application Builder-Bibliotheken verfügen, die separat erhältlich sind. Wenn Sie LabVIEW-Benutzer sind, finden Sie Einzelheiten über die Erstellung einer Anwendung in den *LabVIEW Application Builder Release Notes*, die in der Application Builder-Software enthalten sind.



Hinweis

Da das Ausführungssystem für BridgeVIEW aus mehreren separaten Teilen besteht, funktioniert der Application Builder nicht für die Erstellung von Run-Time-Anwendungen in BridgeVIEW.

Paßwortgeschützte VIs

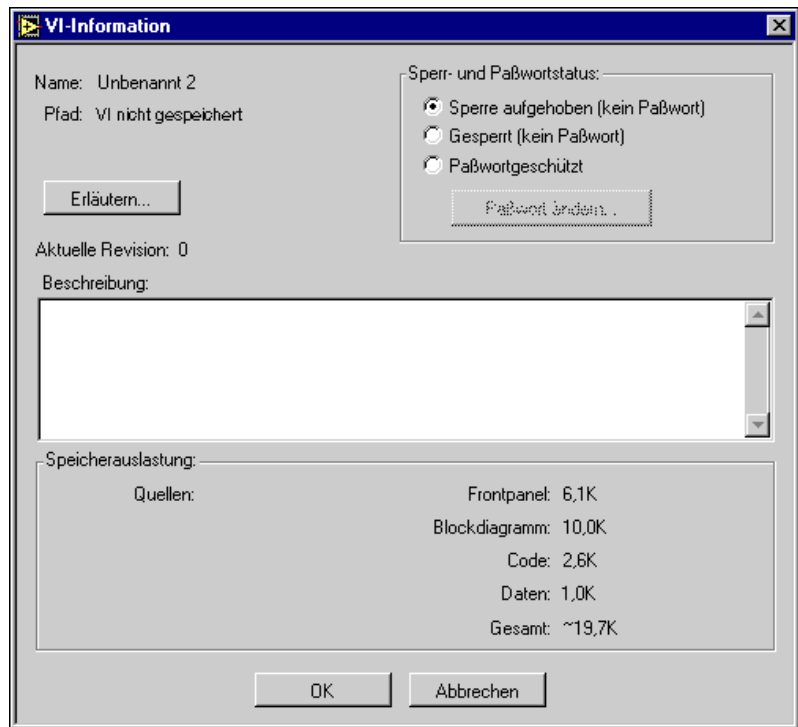
Wenn Sie VIs an andere Personen verteilen, ist es unter Umständen erforderlich, andere daran zu hindern, den im Blockdiagramm erstellten Quellcode einzusehen oder das VI in irgendeiner Weise zu bearbeiten.

Sie könnten dann die VIs ohne Blockdiagramme speichern; allerdings wird dadurch die Software daran gehindert, die VIs neu zu kompilieren, wenn sie in eine andere Plattform oder eine neue Version der Software der G-Sprache importiert werden.

Um andere daran zu hindern, Blockdiagramme einzusehen und VIs zu bearbeiten, können Sie VIs durch Paßwörter schützen. Zur Ansicht des Blockdiagramms eines paßwortgeschützten VIs bzw. zur Bearbeitung dieses VIs in irgendeiner Weise muß der Benutzer das richtige Paßwort für das VI eingeben.

Zum Schutz eines VIs durch ein Paßwort muß **Fenster»VI-Info anzeigen...** oder **Datei»Mit Optionen speichern...** ausgewählt werden.

Mit Hilfe des Dialogfelds **VI-Information** (siehe nachstehende Abbildung) können Sie ein einzelnes VI durch ein Paßwort schützen.



Im folgenden finden Sie Beschreibungen der drei Optionen zum Schutz eines VIs, die im Dialogfeld **VI-Information** zur Verfügung stehen.

Sperre aufgehoben (kein Paßwort) — Gibt an, daß das VI bearbeitet werden kann und nicht paßwortgeschützt ist. Wenn Sie von “Sperre aufgehoben” (kein Paßwort) zu dieser Option umschalten, schaltet das VI sofort in den ungeschützten Zustand um. Wenn der vorherige Zustand “Paßwortgeschützt” war und das Paßwort für dieses VI nicht bereits im Speicher vorhanden ist, werden Sie durch ein Dialogfeld dazu aufgefordert, das Paßwort einzugeben.

Gesperrt (kein Paßwort) — Das VI ist gesperrt und kann somit nicht bearbeitet werden; Sie können es jedoch ausführen und das dazugehörige Diagramm betrachten. Wenn Sie von “Sperre aufgehoben” (kein Paßwort) umschalten, findet der Übergang sofort statt. Wenn jedoch zuvor “Paßwortgeschützt” ausgewählt wurde, müssen Sie das Paßwort eingeben, wenn es nicht gespeichert ist.

Paßwortgeschützt — Dieser Zustand gibt an, daß die VIs paßwortgeschützt sind. Paßwortgeschützte VIs sind mit gesperrten VIs zu vergleichen; Sie können jedoch das Diagramm ohne Eingabe eines Paßwortes nicht aufrufen. Wenn Sie von diesem Zustand umschalten möchten, werden Sie durch ein Dialogfeld zur Eingabe eines Paßwortes aufgefordert, wenn dieses nicht gespeichert ist.

Weitere Informationen über Paßwörter finden Sie im folgenden Abschnitt, *Dialogfeld “Mit Optionen speichern”*.

Dialogfeld “Mit Optionen speichern”

Mit dem Dialogfeld **Mit Optionen speichern** wird es Ihnen einfach gemacht, eine gesamte VI-Hierarchie zur Verteilung zu speichern. Dieses Dialogfeld kann dazu verwendet werden, selektiv eine gesamte Hierarchie ohne die VIs in `vi.lib` zu speichern und alle externen Subroutinen zu speichern, auf die durch die VIs in Ihrer Hierarchie Bezug genommen wird. Das Dialogfeld **Mit Optionen speichern** wird in der nachstehenden Abbildung dargestellt.



Die Optionen links vom Dialogfeld **Mit Optionen speichern** bieten eine Auswahl aus einer Reihe von vordefinierten Speicheroptionen. Wenn Sie Optionen wählen, zeigt der Bereich rechts das Verhalten Ihrer gegenwärtigen Auswahl. Dieses Verhalten kann benutzerspezifisch angepaßt werden, indem bestimmte Optionen rechts vom Dialogfeld ausgewählt werden.

- **Geänderte VIs** — Speichert alle Änderungen des ganz vorne befindlichen VIs und der dazugehörigen SubVIs. Diese Option ist nützlich, da auf diese Weise Änderungen schnell gespeichert werden können.
- **Entwicklungsvertrieb** — Speichert alle VIs, die nicht aus der `vi.lib`-Bibliothek stammen, Bedienelemente und externe Subroutinen an einer einzigen Speicherstelle (entweder ein Verzeichnis oder eine Bibliothek). Diese Option ermöglicht das unkomplizierte Speichern einer Hierarchie, die in ein anderes G-Entwicklungssystem exportiert werden soll, da das andere Entwicklungssystem über seine eigene `vi.lib`-Bibliothek verfügt.
- **Anwendungsverteilung** — Speichert alle VIs, Bedienelemente und externen Subroutinen einschließlich derjenigen in der `vi.lib`-Bibliothek an einer einzigen Speicherstelle und entfernt die Diagramme aus allen VIs. Wenn Sie eine Run-Time-Anwendung mit einer eingebetteten Bibliothek erstellen möchten, können Sie diese Option verwenden. Zur tatsächlichen Erstellung einer Run-Time-Anwendung müssen Sie über die Bibliotheken von Application Builder verfügen, die separat erhältlich sind.

- **Speichern anpassen** — Wählt die spezifischen Optionen aus, die Sie aus dem Bereich **Gewählte Optionen** des Dialogfelds verwenden möchten, wenn keine der vordefinierten Optionen Ihren Anforderungen gerecht wird.
- **Vorlage** — Speichert das VI als VI-Vorlage mit der `.vif`-Dateierweiterung. Wenn eine VI-Vorlage geladen wird, erstellt sie eine Kopie von sich selbst, die unter einem neuen Namen gespeichert werden muß.

Die Taste **Paßwort ändern...** in dem Dialogfeld **Info anzeigen** ist nicht verfügbar, wenn das VI nicht paßwortgeschützt ist. Wenn Sie ein Paßwort ändern möchten, werden Sie in einem Dialogfeld zur Eingabe eines Paßworts aufgefordert, wenn es nicht gespeichert ist.

Mit dem Dialogfeld **Mit Optionen speichern** (siehe vorherige Abbildung) können Sie ein VI oder mehrere VIs im nachhinein durch ein Paßwort schützen. Im folgenden werden der gesperrte Status und der Paßwortstatus im Dialogfeld “Mit Optionen speichern” im einzelnen beschrieben:

- **Keine Paßwortänderung** — Die Auswahl dieser Option hat keine Änderung zur Folge.
- **Paßwörter entfernen** — Bei Auswahl dieser Option werden alle VIs, die noch nicht gespeichert wurden, ohne Paßwortschutz gespeichert. Wenn beim Speichervorgang paßwortgeschützte VIs festgestellt werden, erscheint für jedes Paßwort (nicht unbedingt für jedes VI), das mit einem VI oder einer Gruppe von VIs verknüpft ist, ein Dialogfeld und fordert Sie zur Eingabe des entsprechenden Paßworts auf. Zum Entfernen der Paßwörter müssen Sie das Paßwort für jedes VI bzw. für jede Gruppe von VIs wissen.
- **Neues Paßwort anwenden** — Durch die Auswahl dieser Option werden VIs durch ein Paßwort geschützt. Wenn beim Speichern VIs festgestellt werden, die bereits paßwortgeschützt sind, erscheint für jedes Paßwort (nicht unbedingt für jedes VI), das mit einem VI oder einer Gruppe von VIs verknüpft ist, ein Dialogfeld und fordert Sie zur Eingabe des entsprechenden Paßworts auf. Zum Verwenden der neuen Paßwörter für VIs oder Gruppen von VIs müssen Sie das Paßwort für jedes VI bzw. für jede Gruppe von VIs wissen.
- **Diagramme entfernen** — Mit dieser Option werden Ihre VIs ohne Blockdiagramme gespeichert, so daß die VIs nicht so viel Speicherplatz benötigen. Die VIs erfordern dann zwar weniger Speicherplatz, sie können jedoch nicht in eine andere Version Ihrer Software oder auf andere Plattformen übertragen werden.

Entwerfen von Anwendungen durch mehrere Entwickler

Falls mehrere Entwickler an demselben Projekt arbeiten, sollte etwas Zeit darauf verwendet werden, die Aufgaben der einzelnen Mitarbeiter zu definieren, damit das Projekt reibungslos abläuft. Sie könnten auch die Verwendung des Professional G Developers Toolkits in Erwägung ziehen, das Werkzeuge zur Steuerung des Quellcodes bietet, die bei Umgebungen mit mehreren Entwicklern von Nutzen sind.

Beginnen Sie mit dem Projekt, indem Sie sich das Top-Level-Design Ihrer Anwendung überlegen, und erstellen Sie dann einen Entwurf. Erstellen Sie *VI-Gerüste* für die Hauptkomponenten Ihrer Anwendung. Ein VI-Gerüst ist ein Prototyp eines SubVIs. Es verfügt über Eingänge und Ausgänge, ist jedoch noch unvollständig. Es dient als Platzhalter für zukünftige VI-Entwicklungen, und Sie können die Funktionalität später hinzufügen. Die Erstellung von VI-Gerüsten für LabVIEW-Benutzer wird in Kapitel 28, *Programmdesign* im *LabVIEW Benutzerhandbuch* beschrieben. BridgeVIEW-Benutzer siehe Kapitel 15, *Program Design* Ihres *BridgeVIEW User Manual*.

Wenn sichergestellt ist, daß die VI-Gerüste die erwartete Funktionalität bieten, können Sie festlegen, an welchen Komponenten die einzelnen Entwickler arbeiten sollen. Zur Vereinfachung der Arbeit können Sie die wichtigsten VI-Gerüste in separate Verzeichnisse oder VI-Bibliotheken ablegen, die dann von verschiedenen Entwicklern verwaltet werden.

Aufbewahrung der Originalkopien

Es ist ratsam, die *Originalkopien* Ihrer Projekt-VIs auf einem einzigen Computer aufzubewahren. Sie können dann zur Kontrolle der Änderungen an Ihren VIs allgemeine Richtlinien zum Holen und Ablegen der VIs einführen. Dabei kann ein Entwickler eine Kopie der VIs "ausleihen", ähnlich wie das Ausleihen eines Buchs von einer Bibliothek. Während dieses Zeitraums dürfen andere Entwickler keinen Zugriff auf diese Dateien erhalten. Der Entwickler legt dann die fertigen VIs nach Durchführung der Änderungen wieder ab.

Das Professional G Developers Toolkit enthält Source Code Control-Werkzeuge (SCC), mit denen die gemeinsame Nutzung von Dateien und das Holen und Ablegen von VIs relativ einfach gemacht werden. Außerdem enthält es ein Dienstprogramm, das zum Vergleichen von VIs und zur Betrachtung der Unterschiede verwendet werden kann. Dieses Programm kann sich als außerordentlich nützlich erweisen, wenn Sie überprüfen möchten, welche Änderungen in den verschiedenen Versionen von VIs vorgenommen wurden.

VI-Historiefenster

Zusätzlich zum Frontpanel und Blockdiagramm verfügt ein VI über ein Historiefenster, in dem die Entwicklungshistorie des VIs einschließlich Revisionsnummern angezeigt wird. Jeder Benutzer, der Änderungen an einem VI vornimmt, kann Änderungen in der Historie aufzeichnen. Mit dieser Funktion können Benutzer Änderungen an einem VI überwachen, während diese Änderungen vorgenommen werden. Die VI-Historie bietet jedoch keine Methode zum Vergleichen von zwei VIs, um Unterschiede festzustellen (Informationen über ein Dienstprogramm zur Ausübung dieser Funktionen finden Sie in der Beschreibung des Professional G Developers Toolkits). G speichert die Historie als Teil des VIs. Sie besteht aus einer Reihe von Kommentaren, die von den Benutzern eingegeben wurde, die Änderungen am VI vorgenommen haben.



Hinweis

*Nur wenn Sie die Option **Aufzeichnen der durch LabVIEW erzeugten Kommentare** entweder im Dialogfeld **VI-Einstellungen»Dokumentation oder Bearbeiten»Voreinstellungen»Historie auswählen**, werden die Kommentare in einer VI-Historie automatisch erstellt. Jeder, der Änderungen an dem VI vornimmt, muß die Informationen eingeben und die Historie auf dem neuesten Stand halten.*

Das Fenster "Historie" ist dann verfügbar, wenn Sie das VI bearbeiten, indem Sie **Fenster»Historie anzeigen** auswählen oder die entsprechenden Tasten, <Strg-y> (**Windows**); <Befehlstaste-y> (**Macintosh**); <Metataste-y> (**Sun**); oder <Alt-y> (**HP-UX**) drücken. Nachstehende Abbildung zeigt ein Beispiel eines Historiefensters.

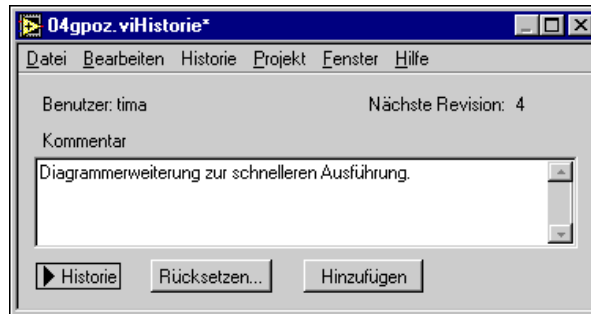


Wenn Sie das VI bearbeiten, können Sie eine Beschreibung wichtiger Änderungen in das Feld “Kommentar” im oberen Bereich des Fensters eingeben. Wenn Sie mit den Änderungen fertig sind, fügen Sie den Kommentar der Historie hinzu, indem Sie auf die Taste **Hinzufügen** klicken. Wenn sich beim Speichern des VIs ein Kommentar im Kommentarfeld befindet, wird dieser der Historie automatisch hinzugefügt. Nachdem Sie den Kommentar hinzugefügt haben, wird er zum permanenten Bestandteil der Historie. Die Historie kann nicht umgeschrieben werden; Sie müssen also vor dem Hinzufügen der Kommentare diese genau überprüfen.

Das Feld im unteren Teil des Fensters zeigt die Historie des VIs an. Das Dialogfeld zeigt für jeden Kommentar in der Historie eine Kopfzeile an, die die Revisionsnummer des VIs, Datum und Zeitpunkt und den Namen der Person angibt, die die Änderung vorgenommen hat. In der oben abgebildeten Historie befinden sich drei Kommentare, wobei jeder mit einer Kopfzeile angezeigt wird. Da es sich bei vorstehender Abbildung lediglich um ein Beispiel handelt, sind die Kommentare kurz gehalten; bei der Bearbeitung Ihrer eigenen VIs können sie jedoch so lang wie gewünscht ausfallen.

Das Historiefenster ist bei angezeigter Historie relativ groß; Sie können die Historie jedoch verbergen, indem Sie auf den Pfeil **Historie** klicken (das kleine schwarze Dreieck links vom Wort “Historie”). Dadurch wird das Fenster so weit verkleinert, daß es Sie bei der Bearbeitung des VIs nicht behindert. Wenn Sie die Historie erneut betrachten möchten, klicken Sie

auf denselben Pfeil, woraufhin die Historie erneut erscheint. Ein Beispiel ist in der nachfolgenden Abbildung dargestellt.



Außerdem können Sie die Größe der Historie und Kommentarfelder ändern, indem Sie die Fenstergröße anpassen.

Revisionsnummern

Die Revisionsnummer ist ebenfalls Teil der Historie des VIs. Mit dieser Nummer kann leicht festgestellt werden, ob das VI geändert wurde (und wie es geändert wurde, wenn Sie einen Kommentar zur Änderung eingegeben haben). Die Revisionsnummer beginnt mit Null und wird jedesmal dann um Eins hochgezählt, wenn Sie das VI speichern. Die aktuelle Revisionsnummer (die sich auf dem Speichermedium befindet) wird im Dialogfeld **Info holen...** angezeigt. Außerdem erscheint sie in der Titelleiste des VIs, wenn die entsprechende Option im Dialogfeld **Bearbeiten»Voreinstellungen...»Historie** mit einem Häkchen markiert ist.

Die im Historiefenster oder VI-Fenster angezeigte Nummer ist die nächste Revisionsnummer, d.h. die Nummer, die auf dem Speichermedium gespeichert wird, wenn Sie das VI speichern. Dabei handelt es sich um die aktuelle Revisionsnummer plus Eins. Wenn Sie der Historie einen Kommentar hinzufügen, wird die nächste Revisionsnummer in die Kopfzeile des Kommentars mit aufgenommen. Wenn Sie in unserem Beispiel auf die Taste **Hinzufügen** klicken, wird Ihr Kommentar mit der Revisionsnummer 4 der Historie hinzugefügt. Wenn Sie das VI speichern, ist die aktuelle Revisionsnummer 4, und der Kommentar, der sich auf die soeben vorgenommenen Änderungen bezieht, wird mit derselben Nummer beschriftet.

**Hinweis**

Die Revisionsnummer wird beim Speichern des VIs nicht um Eins hochgezählt, wenn lediglich Änderungen an der VI-Historie vorgenommen wurden.

In zahlreichen Fällen stimmen die Revisionsnummern nicht mit der Anzahl der Kommentare überein, da das VI ohne Eingabe eines Kommentars gespeichert wurde. Dies ist jedoch kein Problem. Die Revisionsnummer ist sogar sehr nützlich, weil sie unabhängig ist. Angenommen, Sie erhalten eine Kopie eines VIs von einem anderen Benutzer. Später möchten Sie wissen, ob der andere Benutzer das VI aktualisiert hat, seit Sie die Kopie erhalten haben. Dies können Sie leicht anhand der Revisionsnummern feststellen, selbst wenn der andere Benutzer Änderungen vorgenommen hat, ohne Kommentare einzugeben.

Wenn die Revisionsnummer sich nur dann ändern würde, wenn ein Kommentar hinzugefügt wird, würde es sich um eine Kommentarnummer, keine Revisionsnummer handeln. Die Revisionsnummer ist sogar noch wirkungsvoller, wenn sie in Kombination mit der Historie verwendet wird. Wenn Benutzer der Historie Kommentare hinzufügen, können Sie feststellen, welche Änderungen gemacht wurden, seit Sie zuletzt eine Kopie erhalten haben.

Rücksetzen der Historie-Informationen

Unter dem Kommentarfeld befindet sich eine Taste mit der Aufschrift **Rücksetzen...** Durch das Drücken von **Rücksetzen...** wird die Historie gelöscht und die Revisionsnummer wahlweise auf Null zurückgesetzt. Diese Funktion ist dann nützlich, wenn Sie ein VI kopieren und beim neuen VI ganz von vorne anfangen möchten (ohne Historie).

Da es sich bei der Historie um ein reines Entwickler-Werkzeug handelt, wird die Historie automatisch dann entfernt, wenn Sie das Blockdiagramm eines VIs entfernen. Das Historiefenster ist in der Run-Time-Version nicht verfügbar, aber die Revisionsnummer steht im Dialogfeld "VI-Information" zur Verfügung, selbst wenn die Blockdiagramme der VIs entfernt wurden. Die Revisionsnummer kann mit der Taste **Rücksetzen...** entfernt werden, um die Historie Ihres VI zurückzusetzen, bevor Sie das Blockdiagramm entfernen.

Drucken der Historie-Informationen

Wenn Sie ein VI drucken, können Sie die Historien- und Revisionsnummer mit einbeziehen, indem Sie das Dialogfeld **Vollständige Dokumentation** aus dem Dialogfeld “Dokumentation drucken” auswählen. Wenn Sie das Format des Ausdrucks ändern möchten, so daß nur die Historie-Informationen miteinbezogen werden, können Sie die Taste **Druckeinstellungen anpassen** verwenden, um genau anzugeben, was gedruckt werden soll. Außerdem können die Historie-Informationen in eine Datei exportiert werden, indem **Textinfo speichern** aus dem Dialogfeld “Dokumentation drucken” gewählt wird.

Einstellung von verwandten Dialogfeldoptionen für VI-Einstellungen und Voreinstellungen

Die Dialogfelder **VI-Einstellung** und **Voreinstellungen** verfügen beide über Optionen, die dazu verwendet werden können, das Verhalten der VI-Historie einzustellen.

Sie können die Optionen **VI-Einstellung** dazu verwenden, um bei einem gegebenen VI einzustellen, ob und wann Historie-Informationen automatisch aufgezeichnet werden sollen. Informationen über diese Einstellungen finden Sie im Abschnitt *Dokumentationsoptionen* von Kapitel 6, *Einrichten von VIs und SubVIs*.

Mit den Optionen im Dialogfeld **Voreinstellungen** können Sie die standardmäßigen Historie-Einstellungen für neue VIs definieren. Außerdem kann dieses Dialogfeld dazu verwendet werden, einen Benutzernamen anzugeben, der in den Historie-Informationen eingegeben wurde. Informationen über diese Einstellungen finden Sie im Abschnitt *Historie-Voreinstellungen* von Kapitel 7, *Anpassen Ihrer Umgebung*.

Leistung

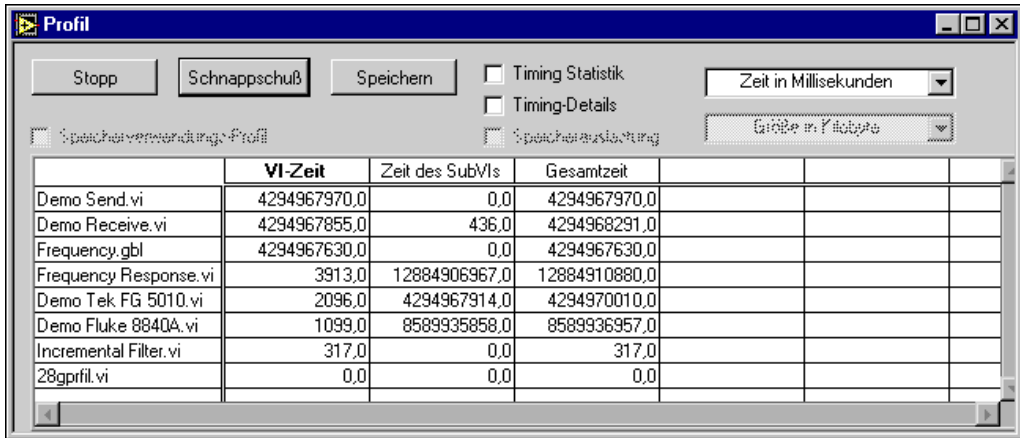
Dieses Kapitel ist in drei Abschnitte unterteilt. Der erste Abschnitt beschreibt den Performance Profiler, eine Funktion, mit der Daten zur Ausführungszeit Ihres VI angezeigt und Singlethread-, Multithread- und Mehrprozessor-Anwendungen überwacht werden können. Im zweiten Abschnitt werden die Faktoren beschrieben, die sich auf die Ausführungszeit auswirken. Im dritten Abschnitt werden Faktoren beschrieben, die sich auf die Speicherauslastung auswirken.

VI-Leistungsprofile

Der VI Performance Profiler ist ein leistungsstarkes Werkzeug, mit dem festgestellt werden kann, wieviel Zeit für Ihre Anwendungen aufgewendet und wie der Speicherplatz ausgenutzt wird. Diese Informationen sind äußerst wichtig, um die Schwachpunkte in Ihrer Anwendung zu bestimmen, so daß VIs, die sehr viel Zeit beanspruchen, optimal eingesetzt werden können. Das Werkzeug verfügt über eine interaktive tabellarische Anzeige, auf der die Verwendung von Zeit und Speicherplatz für jedes VI in Ihrem System angezeigt werden kann. Jede Zeile in der Tabelle enthält Informationen über ein bestimmtes VI. Die für jedes VI aufgewendete Zeit wird in mehrere Kategorien unterteilt und ist in einigen statistischen Darstellungen zusammengefaßt. Das Leistungsprofil-Fenster berechnet die Mindestzeit, Höchstzeit und Durchschnittszeit, die pro Ausführung eines VI aufgewendet wird.

Die interaktive tabellarische Anzeige kann zur Ansicht aller Informationen verwendet werden; außerdem können die Angaben nach unterschiedlichen Kategorien sortiert werden, und die Leistungsdaten von SubVIs beim Aufruf durch ein spezifisches VI können betrachtet werden.

Wählen Sie **Projekt»Profilfenster anzeigen**, um das Profilfenster aufzurufen. Folgende Abbildung ist ein Beispiel eines Fensters, das bereits verwendet wird.



In diesem Fenster sind mehrere Punkte zu beachten. Zunächst einmal ist das Sammeln von Informationen über die Speicherauslastung optional, und zwar deshalb, weil das Sammlungsverfahren Overhead bedeutet und die Ausführungszeit Ihres VI erheblich beeinträchtigen kann. Vor dem Start des Profilers müssen Sie wählen, ob Daten gesammelt werden sollen, indem Sie das Kontrollkästchen **Speicherverwendungsprofil** dementsprechend mit einem Häkchen versehen. Die Markierung dieses Kontrollkästchens kann nicht mehr geändert werden, wenn eine Profilerstellung einmal gestartet wurde. Folgende Tasten sind im Profilfenster verfügbar.

Start

Schnappschuß

Speichern

- **Start** — Ermöglicht die Sammlung von Leistungsdaten. Es ist ratsam, eine Profilerstellung zu starten, wenn Ihre Anwendung nicht ausgeführt wird. Dadurch kann sichergestellt werden, daß nur vollständige VI-Ausführungen und keine Teilausführungen erfaßt werden.
- **Schnappschuß** — Betrachtet die Daten, die momentan zur Verfügung stehen. Hiermit werden Daten von allen VIs im Speicher gesammelt und in der tabellarischen Anzeige angezeigt.
- **Speichern** — Speichert die aktuell angezeigten Daten als Text-Tabellendatei mit Tabulator-Trennzeichen auf ein Speichermedium. Diese Daten können dann in einem Tabellenprogramm oder durch VIs betrachtet werden.

Anzeigen der Ergebnisse

Sie können nach Belieben nur Teilinformationen in der Tabelle anzeigen lassen. Einige grundlegende Daten bleiben stets sichtbar, aber Statistik, Details und (falls aktiviert) Speicherauslastung können nach Belieben angezeigt werden, indem die entsprechenden Kontrollkästchen im Profilenster aktiviert oder deaktiviert werden.

Leistungsinformationen werden auch für Global-VIs angezeigt. Diese Informationen erfordern jedoch manchmal eine geringfügig abweichende Interpretation, wie in den Abschnitten zu den einzelnen Kategorien beschrieben ist.

Leistungsdaten für SubVIs, die von einem spezifischen VI aufgerufen werden, können angezeigt werden, indem Sie auf den Namen dieses VIs in der tabellarischen Anzeige doppelklicken. Daraufhin erscheinen neue Zeilen direkt unter dem Namen des VIs, in denen Leistungsdaten für jedes SubVI aufgeführt sind. Wenn Sie auf den Namen eines Global-VIs doppelklicken, erscheinen für jedes einzelne Bedienelement neue Zeilen auf dem entsprechenden Frontpanel.

Die Datenzeilen in der tabellarischen Anzeige können sortiert werden, indem Sie in der Kopfzeile der gewünschten Spalte klicken. Die aktuelle Sortierspalte wird durch einen fettgedruckten Kopfzeilentitel markiert.

Die Zeitwerte für VIs entsprechen nicht unbedingt der Zeit, die bis zur vollständigen Ausführung eines VIs verstrichen ist. Dies liegt daran, daß ein Multithread-Ausführungssystem die Ausführung von zwei oder mehr VIs übernehmen kann. Außerdem gibt es ein gewisses Overhead, das keinem VI direkt zugewiesen wird, wie z.B. die Zeit, die ein Benutzer benötigt, auf ein Dialogfeld zu reagieren, oder der Zeitaufwand für die Wartefunktion in einem Blockdiagramm oder die für Mausclicks erforderliche Zeit.

Die grundlegenden Informationen, die stets in den ersten drei Spalten der Leistungsdaten sichtbar sind, umfassen die folgenden drei Bereiche:

- **VI-Zeit** — Die Gesamtzeit, die tatsächlich zur Ausführung des Codes dieses VIs und zur Anzeige der entsprechenden Daten aufgewendet wird sowie die Zeit, die vom Benutzer auf die Interaktion mit den Bedienelementen des Frontpanels (falls vorhanden) aufgewendet wird. Diese Zusammenfassung wird wie nachstehend beschrieben im Feld **Timing-Details** in Unterkategorien unterteilt. Für Global-VIs ist diese Zeit die Gesamtzeit, die auf das Kopieren von Daten in alle oder von allen dazugehörigen Bedienelementen aufgewendet wird. Zur Anzeige der Timing-Informationen für einzelne Bedienelemente in einem Global-VI können Sie auf den Namen des Global-VIs doppelklicken.
- **SubVI-Zeit** — Gesamtzeit, die von allen SubVIs dieses VIs aufgewendet wurde. Dabei handelt es sich um die Summe der VI-Zeit (siehe oben) für alle Aufgerufenen dieses VIs sowie ihrer Aufgerufenen etc.
- **Gesamtzeit** — Summe der beiden oben genannten Kategorien, wobei die Gesamtzeit berechnet wird.

Timing-Informationen

Wenn das Kontrollkästchen **Timing Statistik** aktiviert ist, werden die folgenden Spalten in der tabellarischen Anzeige sichtbar:

- **# Ausführungen** — Anzahl der vollständigen Ausführungen dieses VIs. Für Global-VIs ist diese Zeit die Gesamtanzahl der Zugriffe auf alle dazugehörigen Bedienelemente.
- **Durchschnitt** — Durchschnittszeit, die von diesem VI pro Ausführung aufgewendet wurde. Dabei handelt es sich ganz einfach um die VI-Zeit geteilt durch die Anzahl der Ausführungen.
- **Kürzeste** — Mindestzeit, die das VI auf eine Ausführung aufwendet.
- **Längste** — Höchstzeit, die das VI auf eine Ausführung aufwendet.

Wenn das Kontrollkästchen **Timing-Details** aktiviert ist, können Sie mehrere Timing-Kategorien einzeln anzeigen lassen, in denen die Zeit, die von einem VI aufgewendet wurde, addiert wird. Bei VIs, die über eine relativ aufwendige Benutzeroberfläche verfügen, können Ihnen diese Kategorien dabei helfen festzustellen, welche Vorgänge die meiste Zeit beanspruchen.

- **Diagramm** — Zeit, die nur auf die Ausführung des Codes aufgewendet wird, der für das Diagramm dieses VIs erzeugt wurde.
- **Anzeige** — Zeit, die zur Aktualisierung der Frontpanel-Bedienelemente dieses VIs mit neuen Werten vom Diagramm aufgewendet wurde.
- **Zeichnen** — Zeit, die zum Zeichnen des Frontpanels dieses VIs aufgewendet wurde. Die Zeichenzeit umfaßt folgendes:
 - Die Zeit, die zum Zeichnen eines Frontpanels benötigt wird, wenn sein Fenster gerade geöffnet wurde oder wenn es wieder angezeigt wird, nachdem es durch ein anderes Fenster verborgen wurde.
 - Die Zeit, die konzeptuell als Anzeigezeit erfaßt wird und durch neue Werte vom Diagramm verursacht wird, wobei diese Werte auftreten, weil das Bedienelement durchsichtig und/oder überlappend ist. Diese Bedienelemente müssen den Bildschirmbereich ungültig machen, wenn sie neue Daten aus dem Diagramm empfangen, so daß alles in diesem Bereich in der richtigen Reihenfolge neu gezeichnet werden kann. Andere Bedienelemente können auf dem Frontpanel sofort neu zeichnen, wenn sie neue Daten aus dem Diagramm erhalten. Ein Overhead ist aufgrund der Ungültigmachung und Neuzeichnung erforderlich—ein Großteil dieser zusätzlichen Zeit (jedoch nicht alles) fließt in die Werte für das Zeichen-Timing mit ein.
- **Tracking** — Die Zeit, die zum Tracking der Maus aufgewendet wurde, während der Benutzer mit dem Frontpanel dieses VIs interagierte. Dies kann für bestimmte Arten von Operationen von Bedeutung sein, wie z.B. beim Vergrößern und Verkleinern eines Graphen, bei der Auswahl von Menüpunkten aus einem Popup-Menü oder der Auswahl und Eingabe von Text in einem Bedienelement.
- **Lokale Variable** — Die Zeit, die zum Kopieren von Daten in oder von lokalen Variablen auf dem Diagramm aufgewendet wurde. Erfahrungen mit Benutzern haben gezeigt, daß dieser Zeitaufwand manchmal ganz erheblich sein kann, besonders, wenn es sich um umfangreiche, komplexe Daten handelt.

Speicherinformationen

Wenn das Kontrollkästchen **Speicherauslastung** aktiviert ist (denken Sie daran, daß diese Option nur dann zur Verfügung steht, wenn das Kontrollkästchen **Speicherverwendungsprofil** aktiviert wurde, bevor Sie mit der Profilerstellung begannen), können Sie Informationen über die Speicherauslastung Ihrer VIs anzeigen lassen. Diese Werte sind Maßangaben zur Speicherauslastung durch den Datenplatz für das VI und umfassen nicht die unterstützenden Datenstrukturen, die für alle VIs erforderlich sind. Der Datenplatz für das VI enthält nicht nur die Daten, die explizit von Frontpanel-Bedienelementen verwendet werden, sondern auch temporäre Puffer, die implizit vom Compiler erstellt werden.

Die Speichergößen werden nach Beendigung der Ausführung eines VIs gemessen und spiegeln unter Umständen nicht die genaue Gesamtauslastung wider. Wenn z.B. ein VI während seiner Ausführung große Arrays erstellt, deren Größe jedoch vor Beendigung des VIs reduziert, spiegeln die angezeigten Größen nicht die größeren Zwischenwerte wider.

In diesem Abschnitt werden zwei Datensätze angezeigt — Daten, die sich auf die Anzahl der verwendeten Byte beziehen und Daten, die sich auf die Anzahl der verwendeten Blöcke beziehen. Ein Block ist ein kontinuierliches Speichersegment, das zum Speichern einer einzelnen Dateninformation verwendet wird. Ein Array aus Integer-Werten kann z.B. mehrere Byte lang sein, nimmt jedoch nur einen Block ein. Das Ausführungssystem verwendet unabhängige Speicherblöcke für Arrays, Strings, Pfade und Bilder (vom Werkzeugsatz zur Bildsteuerung). Umfangreiche Blöcke im Heap-Speicher Ihrer Anwendung können insgesamt eine Verschlechterung der Leistung (nicht nur der Ausführung) mit sich bringen.

Die Kategorien der Speicherauslastung sind nachfolgend aufgeführt.

- **Avg Byte** — Durchschnittliche Anzahl von Bytes, die vom Datenplatz dieses VIs pro Ausführung verwendet werden.
- **Min Byte** — Mindestanzahl von Bytes, die vom Datenplatz dieses VIs für eine Einzelausführung verwendet werden.
- **Max Byte** — Höchstanzahl von Bytes, die vom Datenplatz dieses VIs für eine Einzelausführung verwendet werden.
- **Avg Blöcke** — Durchschnittliche Anzahl von Blöcken, die vom Datenplatz dieses VIs pro Ausführung verwendet werden.

- **Min Blöcke** — Mindestanzahl von Blöcken, die vom Datenplatz dieses VIs für eine Einzelausführung verwendet werden.
- **Max Blöcke** — Höchstanzahl von Blöcken, die vom Datenplatz dieses VIs für eine Einzelausführung verwendet werden.

VI-Ausführungsgeschwindigkeit

Obwohl der Compiler einen Code erzeugt, der im Regelfall sehr schnell ausgeführt wird, ist es ratsam, beim Erstellen von zeitkritischen Anwendungen die Leistung Ihres VI so optimal wie möglich zu gestalten. In diesem Abschnitt werden Faktoren behandelt, die sich auf die Ausführungsgeschwindigkeit auswirken; außerdem werden einige Programmiertechniken erläutert, mit denen eine optimale Leistung erzielt werden kann.

Prüfen Sie folgende Punkte, um die Ursachen einer zu langsamen Leistung zu bestimmen.

- Eingang/Ausgang (Dateien, GPIB, Datenerfassung, Vernetzung)
- Bildschirmanzeige (große Bedienelemente, überlappende Bedienelemente, zu viele Anzeigen)
- Speicherverwaltung (ineffiziente Ausnutzung von Arrays und Strings, ineffiziente Datenstrukturen)

Andere Faktoren, wie z.B. das Ausführungs-Overhead und Overhead für SubVI-Aufrufe, können sich ebenfalls negativ auf die Leistung auswirken; dies ist jedoch normalerweise auf ein geringes Ausmaß beschränkt und stellt nicht die Hauptursache für eine langsame Ausführung dar.

Eingang/Ausgang

Eingang/Ausgang-Aufrufe verursachen in der Regel einen erheblichen Verwaltungsaufwand. Sie nehmen häufig mehr Zeit in Anspruch, als Berechnungsoperationen. Eine einfache Leseoperation am seriellen Ausgang kann z.B. zu einem Verwaltungsaufwand von mehreren Millisekunden führen. Dieser Verwaltungsaufwand besteht für alle Anwendungen, bei denen serielle Anschlüsse verwendet werden. Der Grund für diesen Verwaltungsaufwand liegt darin, daß bei einem I/O-Aufruf Informationen durch mehrere Schichten eines Betriebssystems übertragen werden müssen.

Die beste Methode zum Vermeiden von übermäßigem Overhead ist es, die Anzahl der I/O-Aufrufe auf ein Mindestmaß zu beschränken. Die Leistung wird verbessert, wenn Sie Ihre Anwendung so strukturieren können, daß

bei jedem Aufruf große Datenmengen übertragen werden, anstatt mehrere I/O-Aufrufe mit kleineren Datenmengen vorzunehmen.

Wenn Sie z.B. ein Datenerfassungs-VI (NI-DAQ) erstellen, stehen Ihnen zum Lesen von Daten mehrere Optionen zur Verfügung. Sie können eine Einzelpunkt-Datenübertragungsfunktion wählen wie etwa das VI "AI Kanal abtasten", oder Sie können eine Mehrpunkt-Datenübertragungsfunktion verwenden, wie etwa das VI "AI Signalverlauf erfassen". Wenn Sie 100 Punkte erfassen müssen, verwenden Sie das VI "AI Kanal abtasten" in einer Schleife mit einer Wartefunktion, um das Timing festzustellen. Oder Sie können das VI "AI Signalverlauf erfassen" mit einem Eingang verwenden und angeben, daß Sie 100 Punkte erfassen möchten.

Durch die Verwendung des VIs "AI Signalverlauf erfassen" können wesentlich höhere und genauere Datenabtastraten erzielt werden, da zur Verwaltung der Datenabtwerte Hardware-Timer eingesetzt werden. Außerdem entspricht das Overhead für das VI "AI Signalverlauf erfassen" in etwa dem Overhead für einen einzigen Aufruf zum VI "AI Kanal abtasten", selbst wenn dabei wesentlich größere Datenmengen übertragen werden.

Bildschirmanzeige

Die Aktualisierung von Bedienelementen auf einem Frontpanel kann häufig zu einer der teuersten Operationen in einer Anwendung werden. Dies gilt vor allem dann, wenn einige der komplizierten Anzeigen wie Graphen und Diagramme verwendet werden.

Dieser Verwaltungsaufwand wird bis zu einem gewissen Grad minimiert, weil die meisten Bedienelemente intelligent sind. Wenn sie neue Daten erhalten, die mit den alten Daten übereinstimmen, werden sie nicht neu gezeichnet. Graphen und Diagramme bilden eine Ausnahme zu dieser Regel.

Wenn die Neuzeichnungsrate zu einem Problem wird, ist es am besten, die Anzahl der verwendeten Bedienelemente zu reduzieren und die Anzeigen so einfach wie möglich zu halten. Im Falle von Graphen und Diagrammen können die Autom. Skalierung, Achsenmarker und Gitter deaktiviert werden, um die Anzeigen zu beschleunigen.

Wenn sich Bedienelemente mit anderen Objekten überlappen, wird die Anzeigegeschwindigkeit ganz erheblich reduziert. Der Grund dafür liegt darin, daß bei teilweiser Verdeckung eines Bedienelements ein Mehraufwand erforderlich ist, um diesen Bildschirmbereich neu zu zeichnen. Wenn die Voreinstellung **Graphik glätten** nicht aktiviert ist,

flackert der Bildschirm bei einer Überlappung von Bedienelementen wesentlich mehr.

Wie bei anderen I/O-Typen gibt es bei der Anzeige eines Bedienelements ein gewisses festes Overhead. Mit einigen Bedienelementen wie z.B. Diagrammen können mehrere Punkte an ein Anzeigeelement übergeben werden. Die Anzahl der Diagrammaktualisierungen kann minimiert werden, indem jedesmal mehr Daten an das Diagramm übergeben werden. Sie können wesentlich höhere Datenanzeigeraten erzielen, wenn Sie Ihre Diagrammdaten in Arrays erfassen, um mehrere Punkte gleichzeitig anzuzeigen, anstatt jeden Punkt dann anzuzeigen, wenn er empfangen wird.

Wenn Sie SubVIs entwerfen, deren Frontpanels während der Ausführung geschlossen sind, stellt das Overhead für die Anzeige kein Problem dar. Wenn das Frontpanel geschlossen ist, fällt für Bedienelemente kein Overhead für das Zeichnen an, so daß Graphen nicht kostspieliger sind als Arrays.

Bedien- und Anzeigeelemente verfügen über eine synchrone Popup-Anzeige, die bei Multithread-Systemen steuert, ob Aktualisierungen verzögert werden können. Bei einer Singlethread-Ausführung hat dieser Punkt keine Auswirkungen. Wenn dieser Punkt jedoch in der Singlethread-Version innerhalb eines VIs ein- oder ausgeschaltet wird, wirken sich diese Änderungen auf das Aktualisierungsverhalten aus, falls diese VIs in ein Multithread-System geladen werden.

Standardmäßig ist dieser Punkt deaktiviert, d.h. daß bei der Übergabe von Daten an Bedien- und Anzeigeelemente des Frontpanels durch das Ausführungssystem die Daten übergeben werden können und die Ausführung unmittelbar danach wieder aufgenommen werden kann. Zu einem späteren Zeitpunkt stellt dann das Benutzerschnittstellensystem fest, daß das Bedien- oder Anzeigeelement aktualisiert werden muß; daraufhin wird es neu gezeichnet, um die neuen Daten anzuzeigen. Wenn das Ausführungssystem versucht, das Bedienelement mehrmals in rascher Folge zu aktualisieren, können Sie eventuell die Zwischenaktualisierungen nicht sehen.

In zahlreichen Anwendungen kann dies eine Ausführung erheblich beschleunigen, ohne daß dadurch die Anzeige für den Benutzer beeinträchtigt wird. Selbst wenn z.B. dieser Punkt deaktiviert ist, können Sie einen Booleschen Wert mehrere Hundertmale in einer Sekunde aktualisieren, d.h. es erfolgen mehr Aktualisierungen, als für das bloße Auge erkennbar sind. Asynchrone Anzeigen ermöglichen es dem Ausführungssystem, mehr Zeit auf die Ausführung von VIs aufzuwenden,

wobei Aktualisierungen vom Benutzerschnittstellen-Thread automatisch auf eine langsamere Rate reduziert wird.

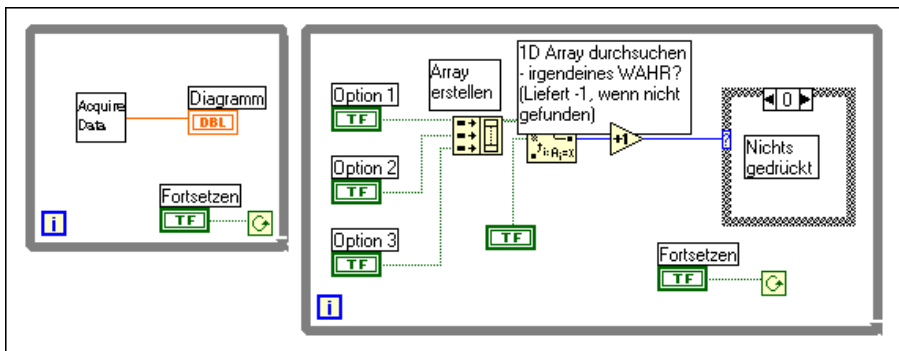
Wenn Sie diese Art von Anzeige vermeiden möchten, können Sie die synchrone Anzeige aktivieren.

Sonstige Themen

Parallele Diagramme

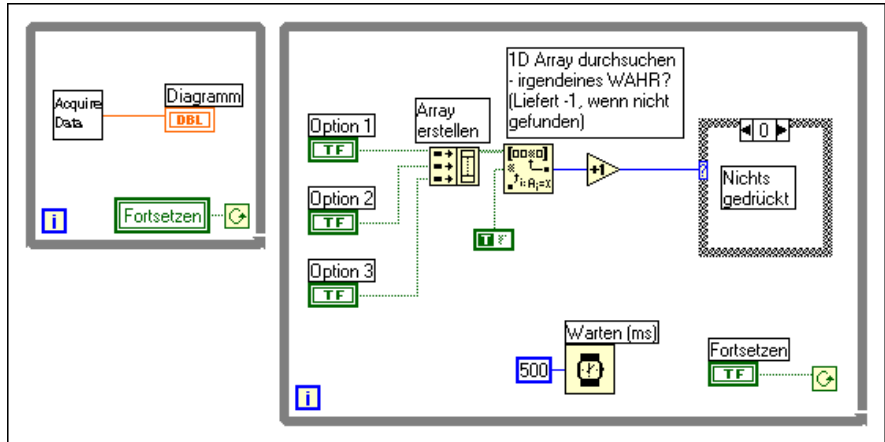
Wenn mehrere Diagramme parallel ablaufen, schaltet das Ausführungssystem in regelmäßigen Abständen zwischen ihnen hin und her. Wenn einige dieser Schleifen weniger wichtig sind, kann mit der Wartefunktion sichergestellt werden, daß diese Schleifen weniger Zeit in Anspruch nehmen.

Betrachten wir z.B. einmal folgendes Diagramm.



Hier handelt es sich um zwei parallele Schleifen. Eine der Schleifen erfaßt Daten und muß so häufig wie möglich ausgeführt werden. Die andere Schleife überwacht die Eingabe durch den Benutzer. Den Schleifen wird aufgrund der Programmstruktur die gleiche Zeit zugewiesen. Die Schleife, mit der die Aktionen des Benutzers überwacht werden, kann somit mehrere Male pro Sekunde ausgeführt werden.

In der Praxis ist es jedoch in der Regel ausreichend, wenn die Schleife zur Überwachung der Taste nur einmal alle halbe Sekunde oder sogar noch seltener ausgeführt wird. Durch das Aufrufen der Wartefunktion (ms) in der Benutzerschnittstellenschleife können Sie der anderen Schleife erheblich mehr Zeit zuweisen.



Overhead für SubVIs

Wenn ein SubVI aufgerufen wird, hat dieser Aufruf ein gewisses Overhead zur Folge. Dieses Overhead ist relativ gering (in der Größenordnung vom Mehrfachen von zehn Mikrosekunden), besonders im Vergleich zum I/O-Overhead und Anzeige-Overhead, das von Millisekunden bis zum Mehrfachen von zehn Millisekunden reichen kann.

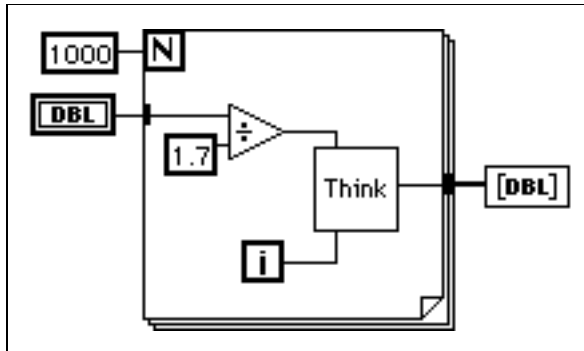
Dieses Overhead kann sich jedoch in einigen Fällen rasch summieren. Wenn Sie z.B. ein SubVI 10.000-mal in einer Schleife aufrufen, kann dieses Overhead unter Umständen einen erheblichen Zeitaufwand bedeuten. In diesem Fall sollten Sie sich überlegen, ob die Schleife in das SubVI eingebettet werden kann.

Eine weitere Option, die Sie verwenden könnten, wäre die Möglichkeit, gewisse SubVIs in Subroutinen umzuwandeln (unter Verwendung des Menüpunktes **Priorität** unter "VI-Einstellungen"). Wenn ein VI als Subroutine markiert wird, minimiert das Ausführungssystem das Overhead zum Aufruf eines SubVIs. Dadurch ergeben sich jedoch einige Nachteile. Subroutinen können keine Frontpaneldaten anzeigen (G kopiert keine Daten von oder in Frontpanel-Bedienelemente von Subroutinen); sie können keine Timing- oder Dialogfeldfunktionen enthalten, und mit ihnen kann kein Multitasking mit anderen VIs durchgeführt werden. Subroutinen sind kurze, häufig ausgeführte Aufgaben und sind im allgemeinen dann angebracht, wenn Sie zusammen mit VIs verwendet werden, bei denen keine Interaktion mit dem Benutzer erforderlich ist. Weitere Informationen finden Sie im Kapitel 26, [Überblick über das G-Ausführungssystem](#).

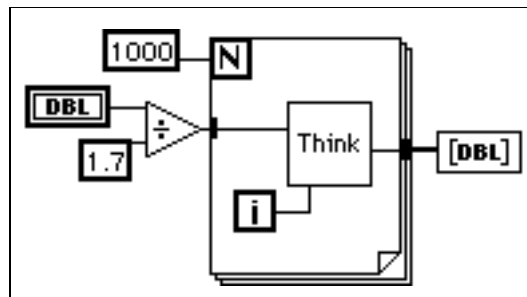
Unnötige Berechnungen in Schleifen

Vermeiden Sie es, eine Berechnung in Schleifen einzubinden, wenn die Berechnung denselben Wert für jede Iteration erzeugt. Verlegen Sie statt dessen die Berechnung aus der Schleife heraus und übergeben Sie das Resultat in die Schleife.

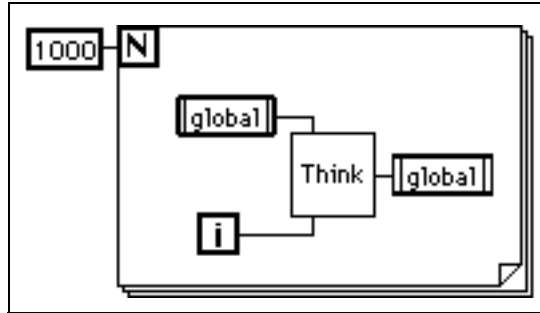
Betrachten wir z.B. folgendes Diagramm.



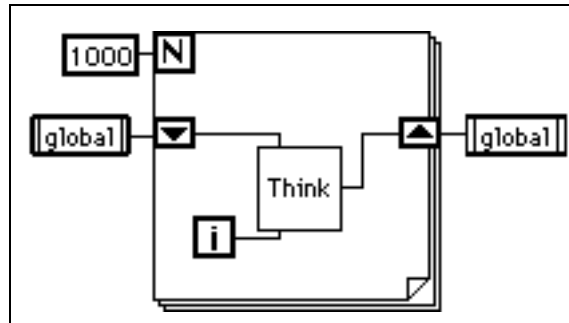
Das Ergebnis der Division ist in der Schleife jedesmal gleich; Sie können somit die Leistung verbessern, indem Sie die Division aus der Schleife heraus legen (siehe folgende Abbildung).



Sehen Sie sich jetzt das Diagramm in folgender Abbildung an.

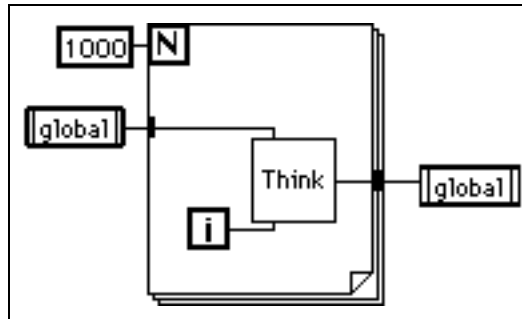


Wenn Sie wissen, daß der Wert der globalen Variable nicht durch ein anderes gleichzeitig ablaufendes Diagramm oder VI in der Schleife geändert wird, verschwendet dieses Diagramm Zeit, da es jedesmal beim Ablauf der Schleife von der globalen Variable abliest und zur globalen Variable schreibt. Wenn während dieser Schleife die globale Variable nicht abgelesen oder in anderes Diagramm geschrieben werden muß, könnte statt dessen folgendes Diagramm verwendet werden.



Beachten Sie, daß die Schieberegister den neuen Wert vom SubVI zur nächsten Iteration der Schleife übergeben müssen. Folgendes Diagramm enthält einen Fehler, der häufig von anfänglichen Benutzern gemacht wird.

Da kein Schieberegister vorhanden ist, werden die Ergebnisse des SubVIs nicht als neuer Eingabewert an das SubVI zurückgegeben.



VI-Speicherauslastung

G berücksichtigt zahlreiche Details, mit denen Sie sich bei einer konventionellen Programmiersprache selbst befassen müssen. Eine der Hauptschwierigkeiten einer konventionellen Sprache ist die Speicherauslastung. Bei einer konventionellen Sprache müssen Sie, der Programmierer, den Speicherplatz vor Ausnutzung entsprechend zuweisen und bei Beendigung wieder freigeben. Außerdem müssen Sie besonders darauf achten, nicht versehentlich über die zuvor zugewiesene Speicherkapazität hinaus zu schreiben. Das Fehlen einer Speicherzuweisung bzw. eine ungenügende Speicherzuweisung ist einer der größten Fehler, die von Programmierern bei konventionellen, textbasierten Sprachen gemacht wird. Darüber hinaus ist ein Debugging von unzureichender Speicherzuweisung schwierig zu bewerkstelligen.

Das Datenfluß-Paradigma für G löst die meisten Probleme, die bei der Verwaltung von Speicherplatz entstehen. In G werden Variablen weder zugewiesen, noch werden ihnen bzw. von ihnen Werte zugewiesen. Statt dessen erstellen Sie ein Diagramm mit Verbindungen, die den Datenübergang darstellen.

Funktionen, die Daten erzeugen, weisen auch gleichzeitig den Speicherplatz für diese Daten zu. Wenn die Daten nicht mehr verwendet werden, wird der mit ihnen verknüpfte Speicherplatz wieder freigegeben. Wenn Sie einem Array oder String neue Informationen hinzufügen, wird automatisch genügend Speicherplatz zum Verwalten der neuen Informationen zugewiesen.

Diese automatische Speicherplatzverwaltung ist einer der Hauptvorteile von G. Da diese Funktionalität jedoch automatisch abläuft, verfügen Sie selbst über weniger Kontrolle darüber, wann diese Funktionen ausgeführt werden. Wenn Ihr Programm mit umfangreichen Datensätzen arbeitet, ist es wichtig zu wissen, wann die Speicherzuweisung stattfindet. Wenn Sie die zugrundeliegenden Prinzipien besser verstehen, sind Sie in der Lage, Programme zu erstellen, die wesentlich geringere Speicherausforderungen haben. Außerdem können Sie, wenn Sie wissen, wie man die Speicherauslastung minimiert, die VI-Ausführungsgeschwindigkeit erhöhen, da die Zuweisung von Speicherplatz und das Kopieren von Daten einen erheblichen Zeitaufwand bedeuten kann.

Virtueller Speicher

Wenn Ihr Computer nur eine beschränkte Speicherkapazität hat, können Sie eventuell virtuellen Speicherplatz verwenden, um den für Anwendungen verfügbaren Speicherplatz zu erhöhen. Der virtuelle Speicher ist eine Fähigkeit Ihres Betriebssystems, bei dem der Festplattenspeicher als Arbeitsspeicher eingesetzt wird. Wenn Sie einen umfassenden virtuellen Speicher zuweisen, wird dies von Anwendungen als Speicher aufgefaßt, der allgemein zum Speichern von Daten zur Verfügung steht.

Bei Anwendungen spielt es keine Rolle, ob es sich bei dem Speicher um einen realen Arbeitsspeicher oder virtuellen Speicher handelt. Das Betriebssystem verdeckt die Tatsache, daß es sich um virtuellen Speicher handelt. Der Hauptunterschied besteht in der Geschwindigkeit. Bei der Verwendung von virtuellem Speicher können Sie gelegentlich eine verlangsamte Leistung beobachten, wenn der Speicher durch das Betriebssystem zur und von der Festplatte ausgelagert wird. Der virtuelle Speicher kann zur Ausführung von größeren Anwendungen nützlich sein, ist jedoch wahrscheinlich nicht für Anwendungen geeignet, die über kritische Zeitbeschränkungen verfügen.

Macintosh-Speicher

Wenn Sie eine Anwendung auf dem Macintosh starten, weist ihr das System einen einzelnen Speicherblock zu, von dem aus der gesamte Speicherplatz zugewiesen wird. Wenn Sie VIs laden, werden deren Komponenten in diesen Speicher geladen. Wenn Sie ein VI ausführen, wird gleichermaßen der gesamte eingesetzte Speicher von diesem einzelnen Speicherblock zugewiesen.

Die Größe des Speichers, der vom System zum Zeitpunkt des Starts zugewiesen wird, können Sie mit dem Befehl **Datei»Info holen** im “Finder” konfigurieren. Denken Sie daran, daß Ihre Anwendung die Größe dieses Speicherpools nicht erhöhen kann, wenn die Speicherkapazität für die Anwendung erschöpft ist. Daher sollte dieser Parameter so eingestellt werden, daß der Speicher so groß ausfällt wie es praktikabel ist. Wenn Sie einen Computer mit einem Arbeitsspeicher von 16 MB besitzen, sollten Sie sich überlegen, welche Anwendungen parallel ausgeführt werden sollen. Wenn Sie nicht planen, andere Anwendungen auszuführen, kann die Voreinstellung für den Speicher so groß wie möglich ausfallen.

Speicherverwaltung der VI-Komponenten

Ein VI hat vier Hauptkomponenten:

- Frontpanel
- Blockdiagramm
- Code (in Maschinencode kompiliertes Diagramm)
- Daten (Werte für Bedien- und Anzeigeelemente, Standarddaten, Diagrammkonstantendaten usw.)

Wenn ein VI geladen wird, werden das Frontpanel, der Code (wenn er mit der Plattform übereinstimmt) und die Daten für das VI in den Speicher geladen. Wenn das VI aufgrund einer Plattformänderung oder Änderung der Schnittstelle eines SubVIs neu kompiliert werden muß, wird auch das Diagramm in den Speicher geladen.

Das VI lädt außerdem den Code und Datenplatz seiner SubVIs in den Speicher. Unter gewissen Umständen kann das Frontpanel einiger SubVIs ebenfalls in den Speicher geladen werden. Dies kann z.B. dann geschehen, wenn das SubVI Attributknoten verwendet, da Attributknoten Zustandsinformationen für Frontpanel-Bedienelemente verarbeitet. Weitere Informationen über SubVIs und ihre Frontpanels werden weiter hinten in diesem Kapitel abgehandelt.

Ein weiterer wichtiger Faktor bei der Organisation von VI-Komponenten ist die Tatsache, daß Sie im allgemeinen nicht viel mehr Speicher verwenden, wenn Sie einen Abschnitt Ihres VI in ein SubVI konvertieren. Wenn Sie ein einziges, riesiges VI ohne SubVIs erstellen, werden das Frontpanel, der Code und die Daten für dieses VIs der höchsten Ebene gespeichert. Wenn Sie das VI jedoch in SubVIs unterteilen, ist der Code für das VI der höchsten Ebene kleiner, und der Code und die Daten des SubVIs sind speicherresident. In einigen Fällen wird auch eine niedrigere Run-Time-Speicherauslastung erzielt. Dieses Konzept wird weiter hinten in diesem Kapitel im Abschnitt *Überwachung der Speicherauslastung* behandelt.

Sie werden auch feststellen, daß umfangreiche VIs mehr Zeit für die Bearbeitung erfordern. Dieses Problem ist nicht so ausgeprägt, wenn Sie Ihr VI in SubVIs unterteilen, da der Editor kleinere VIs effizienter verarbeiten kann. Außerdem ist ein stärker hierarchisch gestaltetes VI im allgemeinen leichter zu pflegen und zu lesen.

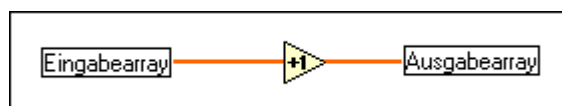
**Hinweis**

Wenn das Frontpanel oder Blockdiagramm eines bestimmten VIs sehr viel größer als der Bildschirm ist, wäre es ratsam, das VI in SubVIs aufzuteilen, damit es leichter zugänglich wird.

Datenfluß-Programmierung und Datenpuffer

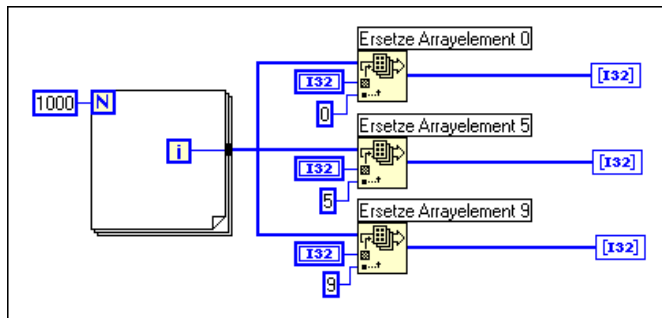
Bei der Datenfluß-Programmierung werden im allgemeinen keine Variablen verwendet. Datenfluß-Modelle beschreiben in der Regel Knoten als verbrauchende Dateneingänge und erzeugende Datenausgänge. Eine exakte Umsetzung dieses Modells erzeugt Anwendungen, die sehr viel Speicherkapazität verwenden und eine sehr langsame Ausführung zur Folge haben. Jede Funktion erzeugt für jedes Ziel, an das eine Ausgabe übergeben wird, eine Kopie der Daten. Der G-Compiler steigert die Leistung der Anwendung, indem er versucht zu bestimmen, wann Speicherkapazität neu eingesetzt werden kann, und die Ziele einer Ausgabe betrachtet, um festzustellen, ob für jedes einzelne Terminal Kopien erstellt werden müssen.

In einem traditionelleren Ansatz für den Compiler z.B. werden im folgenden Diagramm zwei Blöcke mit Datenspeicher verwendet, einer für den Eingang und einer für den Ausgang.



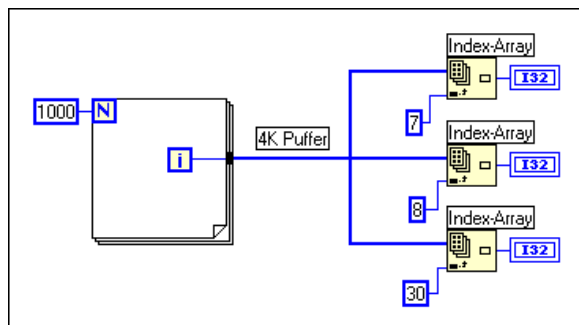
Das Eingangs-Array und das Ausgangs-Array enthalten dieselbe Anzahl von Elementen, und der Datentyp ist bei beiden Arrays gleich. Stellen Sie sich das eingehende Array als einen Puffer mit Daten vor. Anstatt einen neuen Puffer für den Ausgang zu erstellen, verwendet der Compiler den Eingangspuffer neu. Dadurch wird Speicherplatz eingespart, und die Ausführung wird beschleunigt, da bei der Ausführung keine Speicherzuweisung stattfinden muß.

Der Compiler kann Speicherpuffer jedoch nicht in allen Fällen wieder verwenden (siehe nachstehende Abbildung).



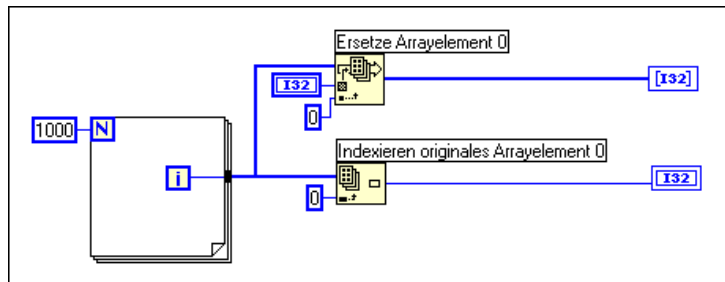
Eine einzelne Datenquelle wird an mehrere Ziele übergeben. Die Ziele müssen die Daten abändern, um die entsprechenden Arrays zu erzeugen. In diesem Fall erstellt der Compiler neue Datenpuffer für zwei der Funktionen und kopiert die Arraydaten in die Puffer. Somit wird durch eine der Funktionen das Eingangs-Array erneut verwendet, während dies bei den anderen Funktionen nicht der Fall ist. Dieses Diagramm verwendet ca. 12 KB (4 KB für das ursprüngliche Array und je 4 KB für die beiden zusätzlichen Datenpuffer).

Betrachten wir einmal folgendes Diagramm.



Wie zuvor sind die Eingangszweige zu drei Funktionen gleich. In diesem Fall müssen jedoch bei keiner Funktion die Daten geändert werden. Wenn die Daten an mehrere Stellen übergeben werden, wobei überall die Daten ohne Änderung gelesen werden, erstellt G keine Kopie der Daten. Bei diesem Diagramm werden ca. 4 KB für Daten verwendet.

Und betrachten wir zum Schluß folgendes Diagramm.



In diesem Fall zweigt der Eingang zu zwei Funktionen ab, wobei eine Funktion die Daten ändern muß. Zwischen den beiden Funktionen besteht keine Abhängigkeit. Daher können Sie vorhersagen, daß mindestens eine Kopie erstellt werden muß, so daß die Funktion “Arrayelement ersetzen” die Daten sicher ändern kann. In diesem Fall plant der Compiler die Ausführung der Funktionen jedoch in der Weise, daß die Funktion, bei der die Daten gelesen werden, zuerst ausgeführt wird, und die Funktion, bei der die Daten geändert werden, zuletzt ausgeführt wird. Somit wird bei der Funktion “Arrayelement ersetzen” der eingehende Array-Puffer wieder verwendet, ohne daß eine Kopie des Arrays erstellt werden muß. Wenn die Anordnung der Knoten von Bedeutung ist, kann die Anordnung explizit erfolgen, indem entweder eine Sequenz oder ein Ausgang eines Knotens als Eingang für einen anderen Knoten verwendet wird.

In der Praxis ist die Analyse von Diagrammen durch den Compiler nicht perfekt. In einigen Fällen ist der Compiler eventuell nicht in der Lage, die beste Methode zur Wiederverwendung von Diagrammspeicher zu bestimmen.

Überwachung der Speicherauslastung

Zur Bestimmung der Speicherauslastung gibt es eine Reihe von Methoden.

Wenn Sie **Hilfe»Über LabVIEW...** wählen (falls Sie ein LabVIEW-Benutzer sind) oder **Help»About BridgeVIEW...** wählen (falls Sie ein BridgeVIEW-Benutzer sind), wird eine Statistik angezeigt, in der der Gesamt Speicherplatz zusammengefaßt wird, der von Ihrer Anwendung in Anspruch genommen wird. Dieser Speicherplatz umfaßt Speicher für VIs sowie den Arbeitsspeicher für die Anwendung. Dieser Wert kann vor und nach der Ausführung eines Satzes mit VIs geprüft werden, um eine ungefähre Vorstellung zu erhalten, wieviel Speicherplatz die VIs beanspruchen.

Der Performance Profiler bietet eine Ansicht der dynamischen Speicherauslastung durch Ihre VIs. Damit werden statistische Daten über die Mindest-, Höchst- und Durchschnittszahl von Byte und Blöcken erstellt, die von jedem VI pro Ausführung verwendet werden. Nähere Einzelheiten hierzu finden Sie im Abschnitt [VI-Leistungsprofile](#) zu Beginn dieses Kapitels.

Wie in folgender Abbildung dargestellt, können Sie **Windows»VI-Info anzeigen...** verwenden, um für ein bestimmtes VI die Speicherauslastung genau aufzuschlüsseln zu lassen. In der linken Spalte wird die Festplattenauslastung aufgeführt, und in der rechten Spalte wird angegeben, wieviel RAM momentan für verschiedene Komponenten des VIs verwendet werden. Beachten Sie, daß die Speicherauslastung von SubVIs in dieser Statistik nicht enthalten ist.



Eine vierte Methode zur Bestimmung der Speicherauslastung ist die Verwendung eines VI, das als Memory Monitor-VI bezeichnet wird (in der Datei `memmon.llb` im Verzeichnis `examples`). Dieses VI verwendet VI-Server-Funktionen, um die Speicherauslastung für alle VIs im Speicher zu bestimmen.

Regeln zur besseren Speicherauslastung

Der Schwerpunkt der vorangegangenen Abhandlung lag darin, daß der Compiler versucht, den Speicher intelligent wiederzuverwenden. Die Regeln dafür, wann der Compiler Speicherkapazität wieder verwenden kann und wann nicht sind relativ kompliziert und werden am Ende dieses Kapitels behandelt. In der Praxis können Ihnen folgende Regeln dabei helfen, VIs zu erstellen, die den Speicher effizient auslasten.

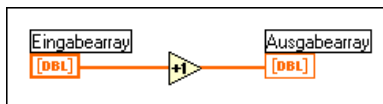
- Die Unterteilung eines VIs in SubVIs wirkt sich im allgemeinen nicht negativ auf die Speicherauslastung aus. In zahlreichen Fällen wird die Speicherauslastung sogar verbessert, da das Ausführungssystem den Datenspeicher von SubVIs wieder freigeben kann, wenn das SubVI nicht ausgeführt wird.
- Kopien von Skalarwerten stellen kein Problem dar; nur eine sehr große Anzahl von Skalarwerten wirkt sich negativ auf die Speicherauslastung aus.
- Beim Arbeiten mit Arrays und Strings sollten nicht zu viele globale und lokale Variablen verwendet werden; das Lesen einer globalen oder lokalen Variable führt zur Erstellung einer Kopie der Daten der Variable.
- Bei geöffneten Frontpanels sollten nur dann große Arrays und Strings angezeigt werden, wenn es notwendig ist. Anzeigeelemente auf geöffneten Frontpanels behalten eine Kopie der angezeigten Daten bei.
- Wenn Frontpanels eines SubVIs nicht angezeigt werden, sollten keine ungenutzten Attributknoten auf dem SubVI verbleiben. Attributknoten führen dazu, daß das Frontpanel eines SubVIs im Speicher bleibt, was eine unnötige Speicherauslastung verursachen kann.
- Die Funktion “Datenbereichsprüfung unterdrücken” sollte bei zeit-/speicherkritischen VIs nicht verwendet werden. Das Frontpanel für das SubVI muß geladen sein, um die Bereichsprüfung zu ermöglichen, und es werden zusätzliche Kopien der Daten für Bedien- und Anzeigeelemente des SubVIs angefertigt.
- Beim Entwerfen von Diagrammen muß auf Stellen geachtet werden, bei denen die Größe eines Eingangs von der Größe eines Ausgangs abweicht. Wenn Sie z.B. Stellen sehen, an denen Arrays oder Strings mit den Funktionen “Arrays erstellen” oder “Strings verknüpfen” häufig vergrößert werden, erzeugen Sie Datenkopien.

- Für Arrays sollten konsistente Datentypen verwendet werden, und beim Übergeben von Daten an SubVIs und Funktionen sollte auf Formatumwandelungspunkte geachtet werden; wenn Datentypen geändert werden, fertigt das Ausführungssystem eine Kopie der Daten an.
- Es sollten keine komplizierten, hierarchischen Datentypen verwendet werden (z.B. Arrays aus Clustern mit großen Arrays oder Strings oder Cluster mit großen Arrays oder Strings). Dadurch wird unter Umständen mehr Speicher verwendet, und die Leistung wird beeinträchtigt. Nähere Einzelheiten und Vorschläge zur Taktik beim Entwerfen von Datentypen finden Sie im Abschnitt *Entwicklung effizienter Datenstrukturen* weiter hinten in diesem Kapitel.

Speicher bei Frontpanels

Wenn ein Frontpanel geöffnet ist, behalten Bedien- und Anzeigeelemente ihre eigenen Kopien der Daten, die von ihnen angezeigt werden.

In folgender Abbildung wird die Inkrementfunktion unter Hinzufügung von Bedien- und Anzeigeelementen auf dem Frontpanel dargestellt.



Wenn Sie das VI ausführen, werden die Daten des Frontpanel-Bedienelements an das Diagramm übergeben. Die Inkrementfunktion verwendet den Eingangspuffer erneut. Das Anzeigeelement erstellt dann eine Kopie der Daten für Anzeigezwecke. Somit sind drei Kopien des Puffers vorhanden.

Diese Datenschutzmaßnahme des Frontpanel-Bedienelements verhindert, daß z.B. der Fall eintritt, daß Daten in ein Bedienelement eingegeben werden, das dazugehörige VI ausgeführt wird und die Daten im Bedienelement sich ändern, während sie an gleicher Stelle an darauffolgende Knoten übergeben werden. Gleichmaßen werden Daten von Anzeigeelementen geschützt, so daß sie die vorherigen Werte zuverlässig anzeigen, bis sie neue Daten erhalten.

Bei SubVIs können Sie Bedien- und Anzeigeelemente als Eingänge und Ausgänge verwenden. Unter folgenden Umständen erstellt das Ausführungssystem eine Kopie der Daten des Bedien- und Anzeigeelements des SubVIs.

- Das Frontpanel befindet sich im Speicher — dies kann aus den folgenden Gründen der Fall sein.
 - Das Frontpanel ist geöffnet.
 - Das VI wurde geändert, jedoch nicht gespeichert (alle Komponenten des VIs verbleiben im Speicher, bis das VI gespeichert wird).
 - Das Panel verwendet den Datendruck.
 - Das Diagramm verwendet Attributknoten.
- Das VI verwendet lokale Variablen.
- Das Panel verwendet Datenprotokollierung.
- Ein Bedienelement verwendet die Funktion “Datenbereichsprüfung unterdrücken”.

Einige dieser Gründe sind nicht intuitiv und bedürfen einer ausführlicheren Erklärung.

Ein Grund hängt mit Eigenschaften wie Diagramm-Historie zusammen. Wenn ein Attributknoten in der Lage sein soll, die Diagramm-Historie in SubVIs bei geschlossenen Panels zu lesen, muß das Bedien- oder Anzeigeelement die an es übergebenen Daten anzeigen. Da es zahlreiche andere Eigenschaften dieser Art gibt, behält das Ausführungssystem SubVI-Panels im Speicher, wenn das SubVI Attributknoten verwendet.

Wenn ein Frontpanel Datenprotokollierung oder Datendruck verwendet, behalten Bedien- und Anzeigeelemente Kopien ihrer Daten. Außerdem werden Panels zum Datendruck im Speicher beibehalten, damit das Panel ausgedruckt werden kann.

Wenn ein VI die Funktion “Datenbereichsprüfung unterdrücken” verwendet, werden Daten in Bedien- und Anzeigeelemente des Frontpanels und daraus kopiert. Die Frontpanelwerte werden beibehalten, so daß das Frontpanel angezeigt werden kann, wenn irgendwelche Daten außerhalb des Bereichs liegen. Die Funktion “Datenbereichsprüfung unterdrücken” darf nicht verwendet werden, wenn der Speicherplatz und die Geschwindigkeit ein großes Problem darstellen.

Beachten Sie vor allem folgendes: Wird ein SubVI mit VI-Einstellungen oder SubVI-Einstellungen so eingestellt, daß sein Frontpanel beim Aufrufen angezeigt wird, wird das Panel beim Aufruf des SubVIs in den Speicher geladen. Wenn Sie den Menüpunkt **Schließen, wenn ursprünglich geschlossen** einstellen, wird das Panel aus dem Speicher gelöscht, nachdem das SubVI die Ausführung beendet hat.

SubVIs können den Datenspeicher wieder verwenden

Im allgemeinen kann ein SubVI Datenpuffer vom Aufrufenden ebenso leicht verwenden, als ob die Diagramme des SubVIs auf der obersten Ebene dupliziert wären. In den meisten Fällen wird nicht mehr Speicher verwendet, wenn ein Abschnitt Ihres Diagramms in ein SubVI umgewandelt wird. Bei VIs mit speziellen Anzeigeanforderungen (siehe vorheriger Abschnitt) gibt es für Frontpanels und Bedienelemente eventuell eine zusätzliche Speicherverwendung.

Lokale Variablen können den Datenspeicher nicht wieder verwenden

Wenn Sie SubVIs erstellen, wird ein Anschlußfeld erstellt, das beschreibt, wie Daten an das SubVI und vom SubVI übergeben werden. Datenpuffer, die von mit einem Anschlußfeld verbundenen Terminals stammen, können Datenpuffer von aufrufenden VIs wieder verwenden. Lokale Variablen sind dazu nicht in der Lage. Statt dessen wird, wenn von einer lokalen Variable gelesen wird, ein neuer Puffer für die Daten vom verbundenen Bedienelement erstellt.

Wenn Sie lokale Variablen verwenden, um große Datenmengen von einer Stelle des Diagramms an eine andere Stelle zu übertragen, verwenden Sie in der Regel mehr Speicherplatz, und es kommt in der Folge zu einer langsameren Ausführungsgeschwindigkeit als bei der Datenübertragung über eine Verbindung.

Globale Variablen behalten stets Kopien ihrer Daten

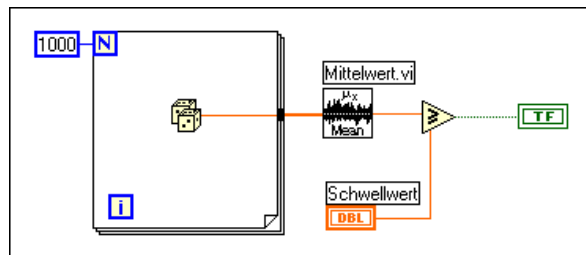
Wenn Sie von einer globalen Variable lesen, erstellen Sie eine Kopie der Daten, die in dieser globalen Variable gespeichert sind.

Beim Manipulieren von großen Arrays und Strings kann der Zeit- und Speicheraufwand zum Manipulieren von globalen Variablen sehr umfangreich sein. Dies erweist sich besonders bei Arrays als ineffizient, wenn Sie nur ein einzelnes Arrayelement ändern möchten und das gesamte Array anschließend speichern. Wenn Sie an mehreren Stellen in Ihrem Diagramm die globale Variable ablesen, erstellen Sie dabei unter Umständen mehrere Speicherpuffer.

Eine Technik zur Optimierung der Speicherauslastung besteht in diesem Fall darin, daß ein nicht initialisiertes Schieberegister in einem SubVI zum Speichern von Daten verwendet wird. Mit dieser Technik kann die Kompaktheit einer globalen Variable mit der Effizienz eines Schieberegisters kombiniert werden. Weitere Informationen hierzu finden Sie unter der Überschrift *Fallstudie 2: Globale Tabelle gemischter Datentypen* weiter hinten in diesem Kapitel.

Wann Speicher freigegeben wird

Betrachten Sie folgendes Diagramm.



Nach der Ausführung des Mean-VIs wird das Array mit Daten nicht mehr benötigt. Da die Feststellung, wann Daten nicht länger benötigt werden, in größeren Diagrammen sehr kompliziert sein kann, werden die Datenpuffer eines bestimmten VIs während seiner Ausführung durch diese nicht freigegeben.

Wenn für das Ausführungssystem auf dem Macintosh nicht genügend Speicher vorhanden ist, werden Datenpuffer beliebiger VIs, die momentan nicht ausgeführt werden, freigegeben. Das Ausführungssystem gibt keinen Speicher frei, der von Bedien- und Anzeigeelementen des Frontpanels, globalen Variablen oder nicht initialisierten Schieberegistern verwendet wird.

Nehmen wir einmal an, daß das zuvor beschriebene VI als SubVI eines größeren VIs eingesetzt wird. Das Array mit Daten wird nur im SubVI erstellt und verwendet. Wenn das SubVI auf dem Macintosh nicht ausgeführt wird und nicht genügend Speicher im System vorhanden ist, können unter Umständen die Daten im SubVI freigegeben werden. Hier handelt es sich um einen Fall, bei dem die Speicherauslastung durch die Verwendung von SubVIs verbessert werden kann.

Auf Windows- und Unix-Plattformen werden Datenpuffer normalerweise erst dann freigegeben, wenn ein VI geschlossen und aus dem Speicher entfernt wurde. Der Speicher wird vom Betriebssystem nach Bedarf zugewiesen, und der virtuelle Speicher kann auf diesen Plattformen wirksam eingesetzt werden. Aufgrund der Fragmentierung scheint die Anwendung unter Umständen mehr Speicher zu verwenden als dies tatsächlich der Fall ist. Wenn der Speicher zugewiesen und freigegeben wird, versucht die Anwendung, die Speicherauslastung so zu konsolidieren, daß unbenutzte Blöcke an das Betriebssystem zurückgegeben werden.

Auf allen Plattformen ist es möglich, die Voreinstellung “Speicher schnellstmöglich freigeben” wahlweise einzustellen. Wenn dieses Element bei Beendigung der SubVI-Ausführung aktiviert ist, wird der Speicher sofort freigegeben. Dies kann sich positiv auf die Speicherauslastung auswirken, hat jedoch eine erhebliche Verlangsamung der Systemleistung zur Folge.

Wann Ausgänge Eingangspuffer wieder verwenden können

Wenn ein Ausgang und ein Eingang über dieselbe Größe und denselben Datentyp verfügen und der Eingang an keiner anderen Stelle benötigt wird, kann der Ausgang den Eingangspuffer wieder verwenden. Wie schon zuvor erwähnt, können der Compiler und das Ausführungssystem in einigen Fällen, selbst wenn ein Eingang an anderer Stelle verwendet wird, die Code-Ausführung so anfordern, daß der Eingang für einen Ausgangspuffer wieder verwendet wird. Die Regeln hierfür sind jedoch komplex und nicht in allen Fällen absolut zuverlässig.

Konsistente Datentypen

Wenn ein Eingang einen anderen Datentyp von einem Ausgang erhält, kann der Ausgang diesen Eingang nicht wieder verwenden. Wenn Sie z.B. einem 16-Bit-Integer-Wert einen 32-Bit-Integer-Wert hinzufügen, sehen Sie einen Formatumwandlungspunkt, der anzeigt, daß der 16-Bit-Integer-Wert in einen 32-Bit-Integer-Wert umgewandelt wird. Der Eingang mit dem 32-Bit-Integer-Wert kann für den Ausgangspuffer verwendungsfähig sein, wenn man davon ausgeht, daß er allen anderen Anforderungen entspricht (z.B. daß die 32-Bit-Integer-Wert nicht an anderer Stelle wieder verwendet wird).

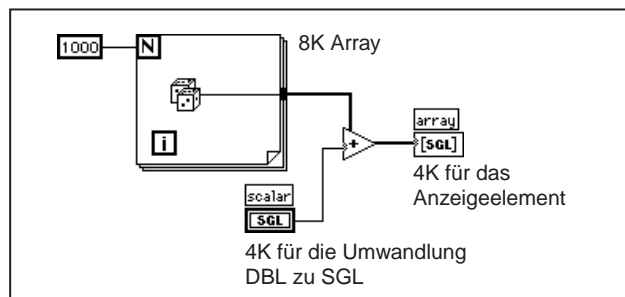
Darüber hinaus implizieren Formatumwandlungspunkte für SubVIs und zahlreiche Funktionen eine Umwandlung von Datentypen. Im allgemeinen erstellt der Compiler einen neuen Puffer für die umgewandelten Daten.

Zur Optimierung der Speicherauslastung sollten so weit wie möglich konsistente Datentypen verwendet werden. Dadurch werden aufgrund der Überführung der Daten nach Größe weniger Datenkopien erzeugt. Durch die Verwendung konsistenter Datentypen wird der Compiler flexibler bei der Bestimmung, wann Datenpuffer wieder verwendet werden können.

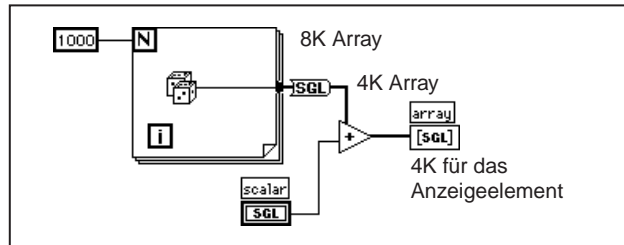
Bei einigen Anwendungen können Sie in Erwägung ziehen, ob Sie kleinere Datentypen verwenden möchten. Sie könnten z.B. Single-Zahlen mit vier Byte anstelle von Double-Zahlen mit acht Byte verwenden. Überlegen Sie sich jedoch, welche Datentypen von aufzurufenden SubVIs erwartet werden, damit unnötige Umwandlungen vermieden werden können.

Wie Daten des richtigen Typs erzeugt werden

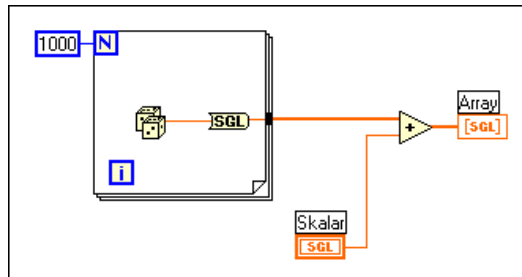
Siehe folgendes Beispiel, in dem ein Array von 1.000 Zufallswerten erstellt und einem Skalar hinzugefügt werden. Der Formatumwandlungspunkt bei der Funktion "Hinzufügen" tritt auf, da es sich bei den Zufallsdaten um Double-Zahlen handelt, während der Skalar eine Single-Zahl ist. Der Skalar wird vor der Hinzufügung zu einer Double-Zahl hochgestuft. Die daraus resultierenden Daten werden dann an das Anzeigeelement übergeben. Dieses Diagramm verwendet bis zu 16 KB Speicher.



In folgender Abbildung wird auf inkorrekte Weise versucht, dieses Problem zu lösen, indem das Array mit Double-Zufallszahlen in ein Array mit Single-Zufallszahlen umgewandelt wird. Dabei wird derselbe Speicherplatz wie im vorherigen Beispiel verwendet.



Die beste Lösung ist die Umwandlung der Zufallszahl in eine Single-Zahl während ihrer Erstellung (siehe folgende Abbildung), und zwar bevor Sie ein Array erstellen. Dadurch wird die Umwandlung eines großen Datenpuffers von einem Datentyp in einen anderen vermieden.

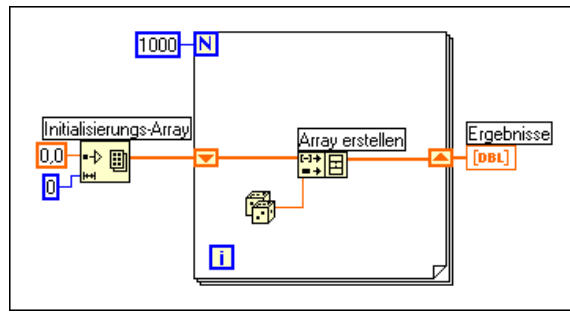


Vermeiden Sie eine ständige Größenänderung von Daten

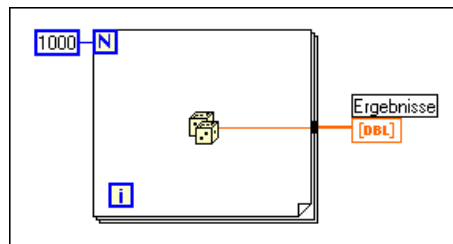
Wenn die Größe eines Ausgangs sich von der Größe eines Eingangs unterscheidet, verwendet der Ausgang die Eingangsdatenpuffer nicht wieder. Dies trifft auf Funktionen wie “Array erstellen”, “String verknüpfen” und “Array-Subset” zu, mit denen ein Array oder String vergrößert oder verkleinert wird. Beim Arbeiten mit Arrays und Strings sollten Sie die ständige Verwendung dieser Funktionen vermeiden, da Ihr Programm mehr Speicherplatz benötigt und langsamer läuft, da es ständig Daten kopieren muß.

Beispiel 1: Erstellen von Arrays

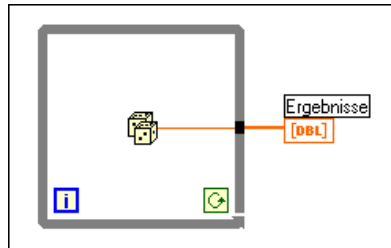
Als Beispiel betrachten wir einmal folgendes Diagramm, das zur Erstellung eines Arrays von Daten verwendet wird. Dieses Diagramm erstellt ein Array in einer Schleife, indem zur Verknüpfung eines neuen Elements ständig die Funktion “Array erstellen” aufgerufen wird. Das Eingangs-Array wird durch “Array erstellen” nicht wieder verwendet. Statt dessen ändert das VI ständig die Größe des Ausgangspuffers, um Platz für das neue Array zu schaffen, und kopiert Daten vom alten Array in das neue Array. Die daraus resultierende Ausführungsgeschwindigkeit ist sehr langsam, besonders dann, wenn die Schleife immer wieder ausgeführt wird.



Wenn Sie dem Array mit jeder Iteration der Schleife einen Wert hinzufügen möchten, wird die beste Leistung dann erzielt, wenn die Auto-Indizierung am Rand einer Schleife verwendet wird. Bei For-Schleifen kann das VI die Größe des Arrays vorausbestimmen (basierend auf dem Wert, der mit N verbunden ist), und die Größe des Puffers wird nur einmal angepaßt.



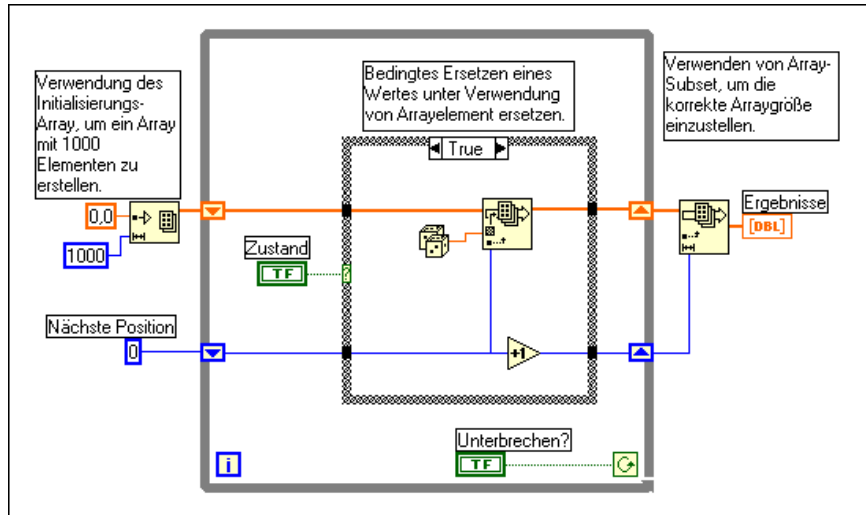
Bei While-Schleifen ist die Auto-Indizierung nicht ganz so wirksam, da die endgültige Größe des Arrays nicht bekannt ist. Die Auto-Indizierung der While-Schleife verhindert jedoch die Größenänderung des Ausgangs-Arrays bei jeder Iteration, indem das Ausgangs-Array in großen Inkrementen vergrößert wird. Wenn die Schleife abgeschlossen ist, wird das Ausgangs-Array an die richtige Größe angepaßt. Die Leistung der Auto-Indizierung der While-Schleife ist nahezu identisch mit der Leistung der Auto-Indizierung der For-Schleife.



Bei der Auto-Indizierung wird davon ausgegangen, daß mit jeder Iteration der Schleife dem resultierenden Array ein Wert hinzugefügt wird. Wenn Sie einem Array bedingt Werte hinzufügen müssen und der Array-Größe bereits eine obere Grenze setzen können, ist es unter Umständen ratsam, das Array im voraus zuzuweisen und dann die Funktion "Arrayelement ersetzen" zu verwenden, um das Array mit Werten zu füllen.

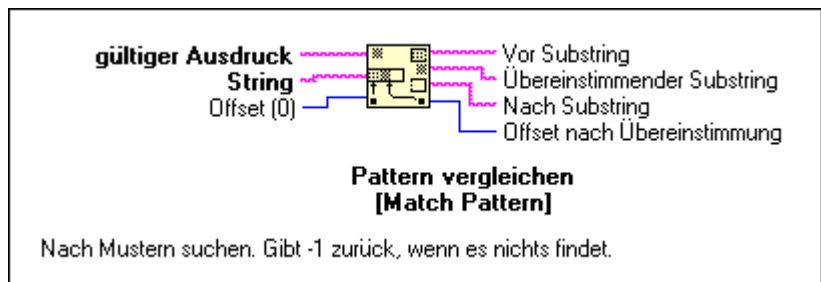
Wenn die Auffüllung der Array-Werte abgeschlossen ist, kann das Array an die korrekte Größe angepaßt werden. Das Array wird nur einmal erstellt, und die Funktion "Arrayelement ersetzen" kann den Eingangspuffer für den Ausgangspuffer wieder verwenden. Die Leistung hierfür entspricht größtenteils der Leistung von Schleifen unter Verwendung der Funktion "Auto-Indizierung". Wenn Sie diese Technik anwenden, muß darauf geachtet werden, daß das Array, in dem Werte ersetzt werden, groß genug für die resultierenden Daten ist, da mit "Arrayelement ersetzen" die Größe der Arrays nicht angepaßt wird.

Ein Beispiel für dieses Verfahren wird in folgender Abbildung dargestellt.



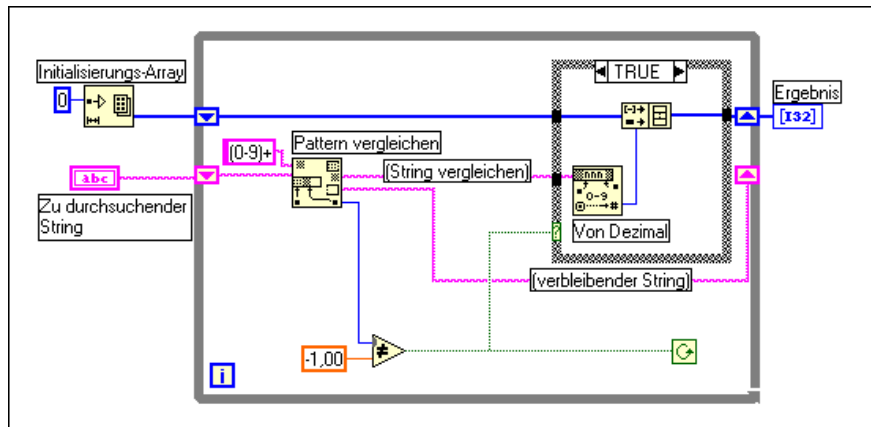
Beispiel 2: Durchsuchen von Strings

Zum Durchsuchen eines Strings nach einem Pattern können Sie die Funktion "Pattern vergleichen" verwenden. Je nachdem, wie diese Funktion eingesetzt wird, kann die Leistung verlangsamt werden, indem unnötigerweise Datenpuffer für Strings erstellt werden. In der folgenden Abbildung wird das Hilfenfenster für die Funktion "Pattern vergleichen" dargestellt.

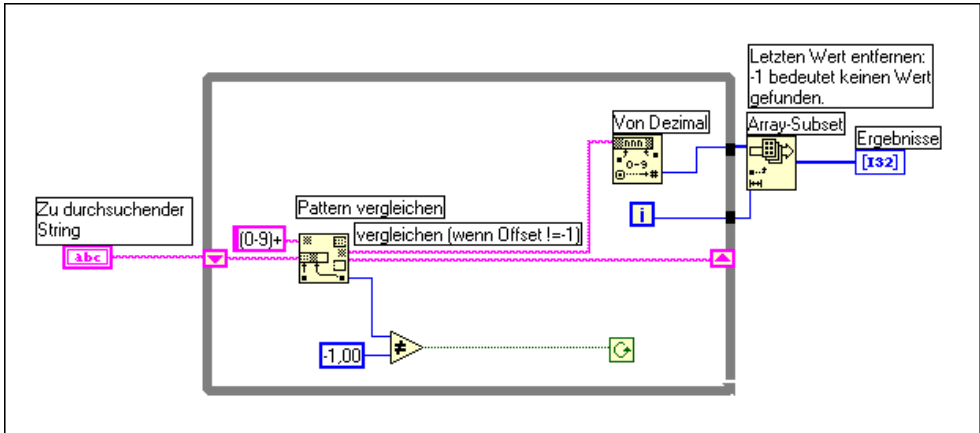


Angenommen, Sie möchten einen Integer-Wert in einem String vergleichen. Sie können dann `[0-9]+` als gültige Ausdruckeingabe für diese Funktion verwenden. Zum Erstellen eines Arrays aller Integer-Werte in einem String verwenden Sie eine Schleife und rufen die Funktion "Pattern vergleichen" wiederholt auf, bis der zurückgegebene Offset-Wert `-1` ist.

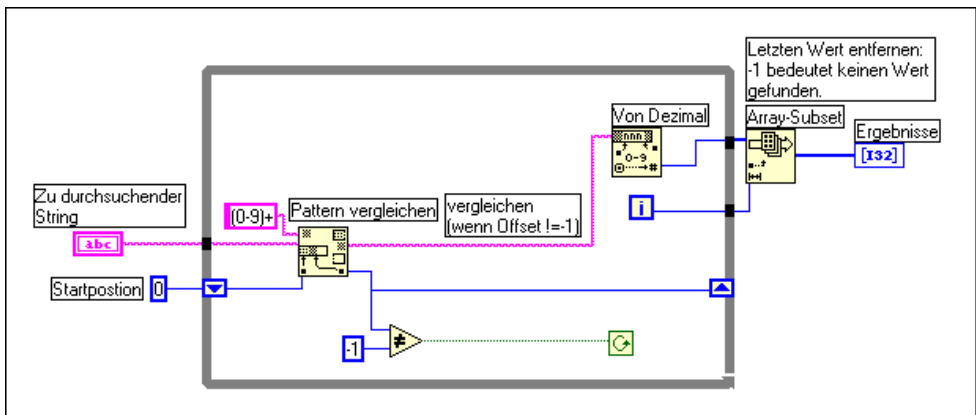
Folgendes Diagramm beschreibt eine Methode zum Suchen aller Integer-Werte in einem String. Zunächst wird ein leeres Array erstellt, und anschließend wird der restliche String bei jedem Ablauf der Schleife auf das numerische Pattern abgesucht. Wenn das Pattern gefunden wird (Offset ist nicht -1), verwendet dieses Diagramm die Funktion "Array erstellen", um dem resultierenden Array aus Zahlen die Zahl hinzuzufügen. Wenn im String keine Werte mehr übrig sind, gibt "Pattern vergleichen" den Wert -1 wieder, und dieses Diagramm schließt die Ausführung ab.



Ein Problem bei diesem Diagramm besteht darin, daß es die Funktion "Array erstellen" in der Schleife verwendet, um den neuen Wert mit dem vorherigen Wert zu verknüpfen. Statt dessen können Sie die Auto-Indizierung verwenden, um Werte am Rand der Schleife zu sammeln. Beachten Sie, daß Sie zum Schluß einen zusätzlichen, unerwünschten Wert im Array antreffen, der aus der letzten Iteration der Schleife stammt, als die Funktion "Pattern vergleichen" keine Übereinstimmung gefunden hat. Eine Lösung für dieses Problem besteht darin, ein Array-Subset zu verwenden, um den zusätzlichen, unerwünschten Wert zu entfernen. Dies wird in folgender Abbildung dargestellt.



Ein weiteres Problem in diesem Diagramm liegt darin, daß Sie bei jedem Ablauf der Schleife eine unnötige Kopie des verbleibenden Strings erstellen. Die Funktion “Pattern vergleichen” verfügt über einen Eingang, der dazu verwendet werden kann, um anzuzeigen, wo mit der Suche begonnen werden soll. Wenn Sie sich an den Offset-Wert in der vorherigen Iteration erinnern können, kann diese Zahl zur Angabe verwendet werden, wo bei der nächsten Iteration mit der Suche begonnen werden soll. Diese Technik wird in folgender Abbildung dargestellt.

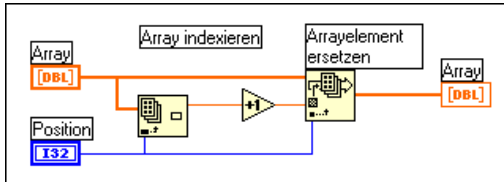


Entwicklung effizienter Datenstrukturen

Es wurde im vorangegangenen Abschnitt unter anderem darauf hingewiesen, daß hierarchische Datenstrukturen wie Arrays aus Clustern mit großen Arrays oder Strings oder Cluster mit großen Arrays oder Strings nicht effizient manipuliert werden können. In diesem Abschnitt wird der Grund dafür erläutert, und es werden Strategien zur Auswahl effizienterer Datentypen beschrieben.

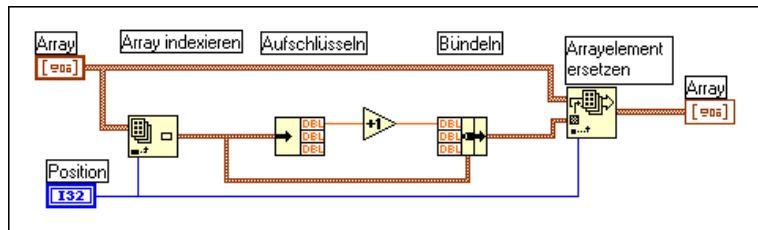
Das Problem bei komplizierten Datenstrukturen besteht darin, daß es schwierig ist, Elemente innerhalb einer Datenstruktur aufzurufen und zu ändern, ohne daß Kopien der aufgerufenen Elemente erzeugt werden. Wenn diese Elemente groß ausfallen, wie es der Fall ist, wenn es sich bei dem Element selbst um ein Array oder einen String handelt, bedeuten diese zusätzlichen Kopien mehr Speicher- und Zeitaufwand zum Kopieren des Speichers.

Im allgemeinen können Skalar-Datentypen sehr effizient manipuliert werden. Gleichmaßen können Sie kleine Strings und Arrays effizient manipulieren, wenn es sich bei dem Element um einen Skalar handelt. Im Fall eines Arrays aus Skalaren zeigt der folgende Code, wie ein Wert in einem Array um eins erhöht wird.



Diese Methode ist sehr wirkungsvoll, da es nicht notwendig ist, zusätzliche Kopien des Gesamtarrays zu erzeugen. Außerdem handelt es sich bei dem durch die Funktion "Array indexieren" erzeugten Element um einen Skalar, der effizient erstellt und manipuliert werden kann.

Dasselbe gilt für ein Array aus Clustern, wenn man davon ausgeht, daß der Cluster nur Skalare enthält. Im folgenden Diagramm wird die Manipulation von Elementen etwas komplizierter, da Sie nun die Funktionen “Bündeln” und “Aufschlüsseln” verwenden müssen. Da der Cluster jedoch wahrscheinlich klein ausfällt (Skalare verwenden sehr wenig Speicher), ist der Zugriff auf Cluster-Elemente und das Wiedereinfügen der Elemente in den ursprünglichen Cluster nicht mit sehr viel Overhead verbunden.



Wenn Sie über ein Array aus Clustern verfügen, bei dem jeder Cluster große Subarrays oder Strings hat, kann die Indizierung und das Ändern der Elementwerte im Cluster mehr Speicher- und Zeitaufwand bedeuten.

Wenn Sie ein Element im Gesamtarray indizieren, wird eine Kopie dieses Elements erstellt. Somit wird auch eine Kopie des Clusters und des dazugehörigen großen Subarrays oder Strings erstellt. Da Strings und Arrays verschieden groß sind, kann das Kopierverfahren zu Aufrufen zur Speicherzuweisung führen, um einen String oder ein Subarray der entsprechenden Größe zu erstellen, und zwar zusätzlich zum Overhead, das durch das tatsächliche Kopieren der Daten eines Strings oder Subarrays entsteht. Diese Operation ist nicht erheblich, wenn sie planmäßig nur einige wenige Male ausgeführt wird. Wenn in Ihrer Anwendung diese Datenstruktur sehr häufig aufgerufen wird, kann sich das Overhead für Speicher und Ausführung allerdings schnell summieren.

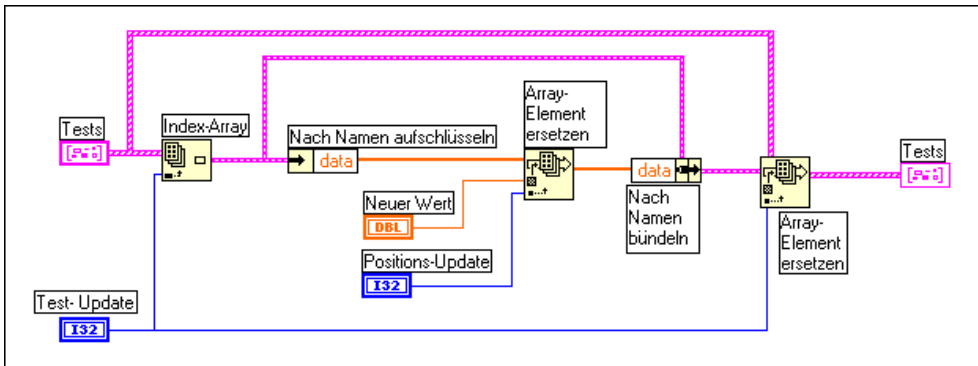
Die Lösung besteht darin, nach alternativen Darstellungsweisen für Ihre Daten zu suchen. Die folgenden drei Fallstudien erläutern drei unterschiedliche Anwendungen sowie Empfehlungen für die besten Datenstrukturen in jedem Einzelfall.

Fallstudie 1: Vermeiden komplizierter Datentypen

Betrachten wir einmal eine Anwendung, in der Sie die Ergebnisse verschiedener Tests aufzeichnen möchten. In den Ergebnissen soll ein String enthalten sein, in dem der Test beschrieben wird, und ein Array der Testergebnisse. Ein möglicher Datentyp zum Speichern dieser Informationen wird in folgender Abbildung beschrieben.



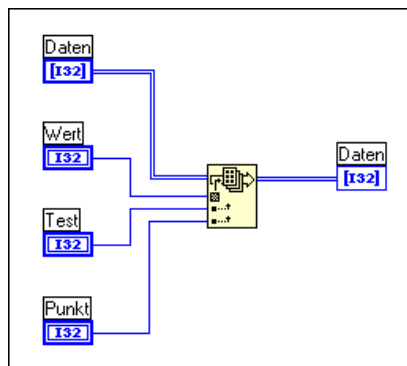
Zum Ändern eines Elements im Array müssen Sie ein Element des Gesamtarrays indizieren. Für diesen Cluster müssen Sie dann die Elemente aufschlüsseln, um das Array zu erreichen. Anschließend ersetzen Sie ein Element des Arrays und speichern das resultierende Array im Cluster. Schließlich speichern Sie den resultierenden Cluster im Original-Array. Ein Beispiel hierfür wird in folgender Abbildung dargestellt.



Jede Ebene der Aufschlüsselung/Indizierung könnte dazu führen, daß eine Kopie der erzeugten Daten erstellt wird. Beachten Sie jedoch, daß eine Kopie nicht unbedingt erstellt wird. Das Kopieren von Daten bedeutet einen erheblichen Zeit- und Speicheraufwand. Die Lösung besteht darin, daß Sie versuchen müssen, die Datenstrukturen so flach wie möglich zu gestalten. In diesem Fall müßte die Datenstruktur z.B. in zwei Arrays unterteilt werden. Bei dem ersten Array handelt es sich um das Array mit Strings. Bei dem zweiten Array handelt es sich um ein 2D-Array, wo jede Zeile die Ergebnisse eines gegebenen Tests darstellt. Dieses Ergebnis wird in folgender Abbildung dargestellt.



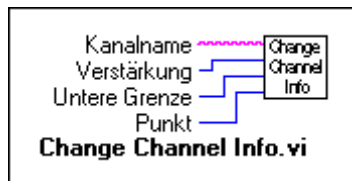
Wenn man von dieser Datenstruktur ausgeht, können Sie ein Arrayelement direkt mit der Funktion "Arrayelement ersetzen" ersetzen (siehe nachstehende Abbildung).



Fallstudie 2: Globale Tabelle gemischter Datentypen

Hier ist eine andere Anwendung, die eine Tabelle mit Informationen enthalten soll. In dieser Anwendung sollen die Daten global zugänglich sein. Diese Tabelle kann Einstellungen für ein Instrument enthalten, einschließlich Verstärkung, untere und obere Spannungsgrenzen und einen Namen, mit dem auf den Kanal Bezug genommen werden kann.

Um die Daten in der gesamten Anwendung zugänglich zu machen, könnte es ratsam sein, einen Satz mit SubVIs zu erstellen, um auf die Daten in der Tabelle zugreifen zu können. Dabei könnten z.B. folgende SubVIs eingesetzt werden: das VI “Kanal-Info ändern” und das VI “Kanal entfernen”.



Im folgenden Abschnitt werden drei unterschiedliche Implementierungen für diese VIs beschrieben.

Offensichtliche Implementierung

Mit diesem Satz von Funktionen müssen für die zugrundeliegende Tabelle mehrere Datenstrukturen berücksichtigt werden. Sie könnten zunächst eine globale Variable mit einem Array aus Clustern verwenden, wobei jeder Cluster die Verstärkung, untere Grenze, obere Grenze und den Kanalnamen enthält.

Wie im vorangegangenen Abschnitt erläutert, ist es schwierig, diese Datenstruktur effizient zu manipulieren, da Sie im allgemeinen mehrere Ebenen der Indizierung und Aufschlüsselung durchlaufen müssen, um auf die Daten zuzugreifen. Und da es sich bei der Datenstruktur um eine Anhäufung von verschiedenen Informationen handelt, können Sie nicht die Funktion “1D Array durchsuchen” zur Suche nach einem Kanal verwenden. Die Funktion “1D Array durchsuchen” kann zur Suche nach

einem spezifischen Cluster in einem Array aus Clustern verwendet werden, aber Sie können sie nicht zur Suche nach Elementen verwenden, die mit einem einzelnen Cluster-Element übereinstimmen.

Alternative Implementierung 1

Wie beim vorangegangenen Beispiel müssen auch hier die Daten in zwei separaten Arrays gespeichert werden. Das eine Array enthält die Kanalnamen. Das andere Array enthält die Kanaldaten. Der Index eines gegebenen Kanalnamens im Array aus Namen wird dazu verwendet, die entsprechenden Kanaldaten im anderen Array zu finden.

Beachten Sie, daß Sie die Funktion “1D Array durchsuchen” zur Suche nach einem Kanal verwenden können, da das Array aus Strings von den Daten getrennt ist.

In der Praxis ist bei der Erstellung eines Arrays aus 1.000 Kanälen unter Verwendung des VIs “Kanal-Info ändern” diese Implementierung ca. doppelt so schnell wie die vorherige Version. Diese Änderung ist allerdings nicht sehr erheblich, da es mehr Overhead gibt, das sich auf die Leistung auswirkt.

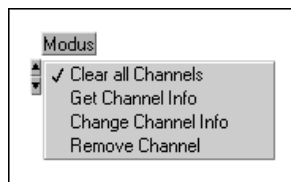
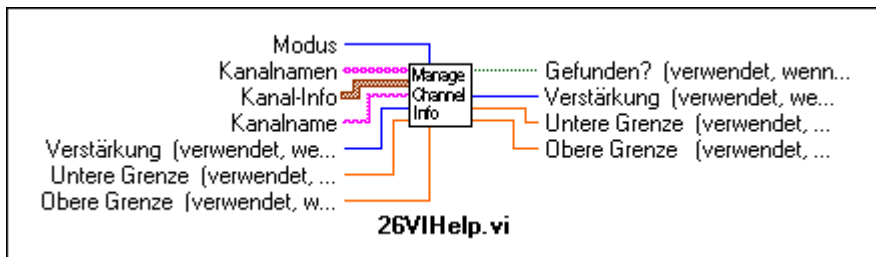
Ein Hinweis im Abschnitt *VI-Speicherauslastung* in diesem Kapitel warnt vor einer übermäßigen Verwendung lokaler und globaler Variablen. Wenn Sie von einer globalen Variable lesen, wird eine Kopie der Daten der globalen Variable erzeugt. Daher wird jedesmal, wenn Sie auf ein Element zugreifen, eine vollständige Kopie der Daten des Arrays erstellt. Der nächste Abschnitt erläutert eine noch effizientere Methode, mit der dieses Overhead vermieden wird.

Alternative Implementierung 2

Zum Speichern globaler Daten gibt es eine alternative Methode: die Verwendung eines nicht initialisierten Schieberegisters. Wenn Sie einen Initialisierungswert nicht verbinden, merkt sich ein Schieberegister von Aufruf zu Aufruf seinen Wert. Wenn Sie LabVIEW-Benutzer sind und mit nicht initialisierten Schieberegistern nicht vertraut sind, siehe Kapitel 3, *Schleifen und Diagramme des LabVIEW Benutzerhandbuchs*, bevor Sie mit dieser Abhandlung fortfahren. Wenn Sie BridgeVIEW-Benutzer sind, finden Sie ähnliche Informationen in Kapitel 10, *Loops and Diagrams in Ihrem BridgeVIEW User Manual*.

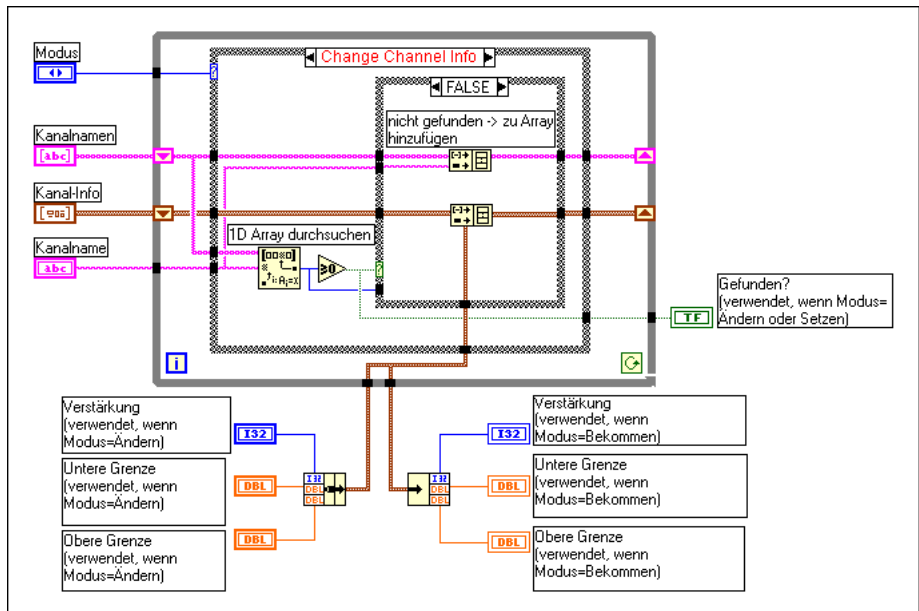
Der G-Compiler behandelt den Zugriff auf Schieberegister auf effiziente Weise. Durch das Ablesen des Wertes eines Schieberegisters wird nicht unbedingt eine Kopie der Daten erzeugt. Sie können ein Array, das in einem Schieberegister gespeichert ist, sogar indizieren und seinen Wert ändern und aktualisieren, ohne daß zusätzliche Kopien des Gesamtarrays erstellt werden. Das Problem beim Schieberegister liegt nur darin, daß das VI, welches das Schieberegister enthält, auf die Schieberegisterdaten zugreifen kann. Andererseits verfügt das Schieberegister über den Vorteil der Modularität.

Um Abhilfe zu schaffen, können Sie ein einzelnes SubVI mit einem Modus-Eingang erstellen, der angibt, ob Sie einen Kanal lesen, ändern oder entfernen möchten oder ob Sie die Daten für alle Kanäle auf Null stellen möchten (siehe nachstehende Abbildung).



Das SubVI enthält eine While-Schleife mit zwei Schieberegistern — eins für die Kanaldaten und eins für die Kanalnamen. Keines der beiden Schieberegister ist initialisiert. Dann plazieren Sie eine mit dem Modus-Eingang verbundene Case-Struktur in die While-Schleife. Je nach dem Wert des Modus' können Sie die Daten im Schieberegister eventuell lesen und ändern.

Nachstehend befindet sich der Umriß eines SubVIs mit einer Schnittstelle, die diese drei verschiedenen Modi verwalten kann. Es wird nur der Code für “Kanal-Info ändern” angezeigt.

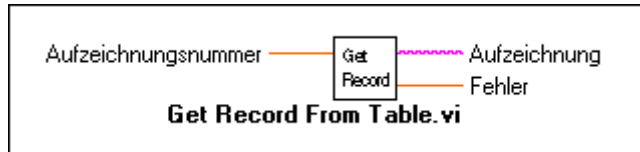
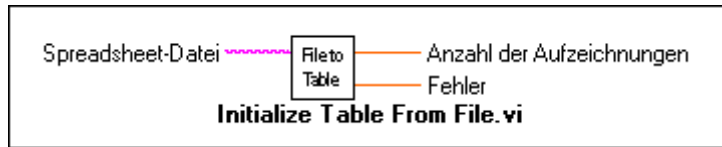


Für 1.000 Elemente ist diese Implementierung doppelt so schnell wie die vorherige Implementierung und viermal so schnell wie die ursprüngliche Implementierung.

Fallstudie 3: Eine statische globale Tabelle mit Strings

Im vorangegangenen Beispiel wurde eine Anwendung behandelt, in der die Tabelle gemischte Datentypen enthielt, wobei sich die Tabelle relativ häufig ändern könnte. In zahlreichen Anwendungen gibt es eine Tabelle mit Informationen, die nach ihrer Erstellung relativ statisch bleiben. Die Tabelle kann dann unter Umständen von einer Tabellendatei gelesen werden. Wenn sie einmal in den Speicher gelesen wurde, kann sie hauptsächlich zum Nachlesen von Informationen verwendet werden.

In diesem Fall könnte Ihre Implementierung aus den folgenden zwei Funktionen bestehen: “Tabelle von Datei initialisieren” und “Datensatz von Tabelle holen”.



Eine Methode zur Implementierung der Tabelle besteht in der Verwendung eines zweidimensionalen Arrays aus Strings. Beachten Sie, daß der Compiler jeden String in einem Array aus Strings in einem separaten Speicherblock speichert. Wenn es eine große Anzahl von Strings gibt (z.B. mehr als 5.000 Strings), könnten Sie die Strings unter Umständen in den Speichermanager laden. Dadurch kann es zu einem erheblichen Leistungsverlust kommen, da sich die Anzahl der einzelnen Objekte erhöht.

Eine alternative Methode zum Speichern einer großen Tabelle ist das Lesen der Tabelle als einzelnen String. Anschließend wird ein separates Array mit den Offsets jedes Datensatzes im String erstellt. Dadurch wird die Organisation geändert, so daß Sie anstelle von Tausenden von relativ kleinen Speicherblöcken einen großen Speicherblock (den String) und einen separaten kleineren Speicherblock (das Array aus Offsets) haben.

Diese Methode ist eventuell etwas schwieriger zu implementieren, kann jedoch bei großen Tabellen erheblich schneller sein.

Portierbarkeit und Lokalisierungsthemen

In diesem Kapitel werden Themen im Zusammenhang mit der Übertragung von VIs zwischen Plattformen und die Lokalisierung behandelt.

Portierbare und nicht-portierbare VIs

VIs sind zwischen allen Plattformen portierbar, auf denen Ihre Anwendung ausgeführt wird, solange die Versionen der Anwendung übereinstimmen. VIs mit CINs oder plattformspezifischen Funktionen wie DDE sind nicht portierbar. In einem solchen Fall wird das VI übertragen, ist jedoch danach defekt.

G verwendet auf allen Plattformen dasselbe Dateiformat. Sie können VIs von einem System in ein anderes übertragen, entweder mit einem Speichermedium oder über ein Netzwerk.

Nachdem die Datei in das neue System übertragen wurde, können Sie das VI öffnen. Ihre Anwendung stellt fest, daß das VI von einer anderen Plattform stammt und kompiliert das VI neu, damit es die richtigen Anweisungen für den aktuellen Prozessor befolgt. Wenn VIs auf ein Speichermedium übertragen werden, das für eine andere Plattform formatiert wurde, ist zum Lesen des Speichermediums eventuell ein Dienstprogramm erforderlich, wie beispielsweise Apple File Exchange auf dem Macintosh.

Folgende VIs können nicht portiert werden.

- VIs, die in der Bibliothek `vi.lib` verteilt werden. Jede Verteilung enthält ihre eigene `vi.lib`; es dürfen daher keine VIs in `vi.lib` zwischen Plattformen übertragen werden.
- VIs, die CINs enthalten. Der Fehler `object code not found` erscheint, wenn das CIN von einer anderen Plattform stammt. Wenn Sie den CIN-Quellcode in einer plattformunabhängigen Weise schreiben, können Sie ihn auf einer anderen Plattform neu kompilieren und wieder mit dem portierten VI verknüpfen.

- VIs, die den Aufruf ext. Bibliotheken enthalten. Das VI kann portiert werden, ist jedoch defekt, wenn keine Bibliothek mit demselben Namen gefunden wird.
- Plattformspezifische Kommunikations-VIs wie AppleEvents auf dem Macintosh und DDE auf Windows.
- In Port- und Out Port Utility-VIs für Windows und die Peek and Poke Utility-VIs für Macintosh.

Portieren zwischen Plattformen

Die Portierbarkeit zwischen Plattformen kann auf mehrere Arten erleichtert werden. Portabilitätsthemen umfassen Unterschiede bei Dateinamen, Trennzeichen, Auflösungen und Schriftarten, mögliche Überlappung von Labels und Unterschiede bei Bildformaten.

Ein Punkt, der in Erwägung gezogen werden muß, ist der Dateiname. Dateinamen sind auf acht Zeichen plus eine wahlweise Erweiterung mit drei Zeichen beschränkt, normalerweise `.vi` in DOS/Windows 3.x und in FAT-Datenträgern in Windows NT. Macintosh-Dateinamen können 31 Zeichen lang sein. Windows 95/NT und UNIX-Dateinamen können 255 Zeichen lang sein, einschließlich der `.vi`-Erweiterung.

Zum Vermeiden von Komplikationen sollten die VIs entweder mit kurzen Namen oder in einer VI-Bibliothek gespeichert werden. Eine VI-Bibliothek ist eine einzige Datei, die mehrere VIs enthalten kann. Ein Bibliotheksname muß den Plattformbeschränkungen entsprechen, aber VIs in Bibliotheken können unabhängig von der Plattform über Namen mit bis zu 255 Zeichen verfügen. Daher sind VI-Bibliotheken das bequemste Format zum Übertragen von VIs, da Bibliotheken die meisten Abhängigkeiten vom Dateisystem ausschalten. Weitere Informationen über die Erstellung von VI-Bibliotheken finden Sie im Abschnitt *Speichern von VIs* im Kapitel 2, *Bearbeiten von VIs*.

Ein anderes Thema in bezug auf Dateinamen ist die Tatsache, daß auf UNIX die Groß/Kleinschreibung beachtet werden muß, wohingegen dies auf anderen Plattformen nicht notwendig ist. Wenn z.B. auf ein VI durch den Namen Bezug genommen wird, muß sichergestellt werden, daß überall dort, wo der Name verwendet wird, die Groß- und Kleinschreibung richtig beachtet wird.

Trennzeichenunterschiede

Wenn Sie ein VI zur Verwendung auf mehreren Plattformen entwickeln, dürfen keine plattformspezifischen Pfadtrennzeichen [\ , / und :] in Ihren Dateinamen verwendet werden. Die Dateinamen sollten keine Sonderzeichen enthalten, da diese von unterschiedlichen Dateisystemen unterschiedlich behandelt werden können. Versteckte Dateien in UNIX beginnen z.B. mit einem Punkt.

Auflösungs- und Schriftartunterschiede

Ein weiteres Portabilitätsthema betrifft Unterschiede der Bildschirmauflösung und Schriftarten. Schriftarten können sich von Plattform zu Plattform unterscheiden, so daß es nach dem Portieren eines VIs eventuell notwendig wird, neue Schriftarten zu wählen, damit die Anzeige attraktiv gestaltet wird. Beim Entwickeln von VIs sollte berücksichtigt werden, daß es drei Schriftarten gibt, die sich am besten zwischen Plattformen übertragen lassen — die Anwendungsschriftart, die Systemschriftart und die Dialogschriftart.

Diese vordefinierten Schriftarten und die tatsächlichen Schriftarten, die abgebildet werden, sind folgende:

- Die Anwendungsschriftart ist die Standardschriftart. Sie wird in der Palette **Elemente**, der Palette **Funktionen** und bei neuen Bedienelementen verwendet.
 - **(Windows)** Die US-Version von Windows verwendet üblicherweise die Schriftart Arial. Die Größe hängt von den Einstellungen des Videotreibers ab, da häufig Videotreiber mit höherer Auflösung zur Verwendung großer Schriftarten oder kleiner Schriftarten eingestellt werden können. In der japanischen Version von Windows verwendet die Anwendung die Schriftart, die von Windows für Dateinamen im Programm-Manager eingesetzt wird.
 - **(Macintosh)** Die Anwendung verwendet dieselbe Schriftart als Anwendungsschriftart, die im Finder für Dateinamen verwendet wird. Auf dem US-Macintosh-System z.B. verwendet die Anwendung Geneva; in der japanischen Macintosh-Version wird jedoch von der Anwendung Osaka eingesetzt.
 - **(UNIX)** Die Anwendung verwendet standardmäßig Helvetica.

- Die Systemschriftart ist die Schriftart, die in Menüs verwendet wird.
 - **(Windows)** Die Anwendung verwendet Helvetica, wobei die Größe vom Videotreiber abhängt.
 - **(Macintosh)** Die Anwendung verwendet normalerweise Chicago in der US-Systemsoftware, Osaka in der japanischen Systemsoftware und so weiter.
 - **(UNIX)** Die Anwendung verwendet normalerweise Helvetica für diese Schriftart.
- Die Dialogschriftart ist die Schriftart, die von G für Text in Dialogfeldern verwendet wird.
 - **(Windows)** In der US-Version von Windows ist diese Schriftart eine Fettdruckversion der Anwendungsschriftart. In der japanischen Version von Windows stimmt diese Schriftart mit der Systemschriftart überein.
 - **(Macintosh)** Die Anwendung verwendet dieselbe Schriftart, die als Systemschriftart eingesetzt wird.
 - **(UNIX)** Die Anwendung verwendet normalerweise Helvetica für diese Schriftart.

Wenn ein VI mit einer dieser Schriftarten auf eine andere Plattform übertragen wird, stellt Ihre Anwendung sicher, daß die Schriftart auf dieser Plattform in einer ähnlichen Schriftart abgebildet wird.

Wenn Sie die vordefinierten Schriftarten nicht verwenden, sondern statt dessen eine spezifische Schriftart wie Geneva oder New York auswählen, kann die Schriftart auf der neuen Plattform in einer anderen Größe erscheinen, da es Unterschiede in den verfügbaren Schriftarten und Unterschiede in der Auflösung der Anzeige gibt. Wenn Sie Geneva oder New York auf dem Macintosh auswählen, findet die Anwendung auf dem Sun- oder HP-UX-System keine Entsprechung, und es wird eine Schriftart mit der Bezeichnung *fixed* eingesetzt. Wenn Sie ein VI mit einer unbekanntenen Schriftart in Windows portieren, wird eventuell eine ungünstige neue Schriftart zugewiesen.

Wenn Sie eine vordefinierte Schriftart für einen Textabschnitt einsetzen, wird von National Instruments empfohlen, die Größe dieses Texts nicht zu ändern. Wenn Sie nämlich die Größe der Schriftart in eine andere Größe ändern, die nicht der Standard ist, und dann das VI auf eine andere Plattform portieren, versucht die Anwendung, die Schriftart auf die neue Größe anzupassen, die unter Umständen aufgrund der Auflösung des Bildschirms unpassend sein kann. Zum Beispiel sieht eine Anwendung mit der Größe 10 (ein Pixel größer als der Standard) auf dem Macintosh gut aus

(Geneva 10), auf Windows jedoch erscheint diese Schriftart mit einem Hochauflösungs-Videotreiber winzig, da sie drei Pixel kleiner als der Standard ausfällt.

Überlappen von Labels

Wenn Sie ein VI auf eine neue Plattform übertragen, wird eventuell die Größe von Bedienelementen und Labels geändert, je nachdem, ob die Schriftarten größer oder kleiner sind. G versucht zu vermeiden, daß sich Labels mit ihren dazugehörigen Bedienelementen überlappen, indem sie von diesen weg plaziert werden. Außerdem hat jedes Label und jede Konstante eine Standardeigenschaft, die als **Größenanpassung an Text** bezeichnet wird. Wenn Sie ein Label oder eine Konstante zum erstenmal erstellen, wird diese Eigenschaft so eingestellt, daß die Größe der Objektgrenzen gegebenenfalls angepaßt wird, damit der gesamte Text, der darin enthalten ist, angezeigt wird.

Wenn Sie die Größenanpassung des Objekts einmal manuell vornehmen, schaltet die Anwendung diese Eigenschaft aus (der Menüpunkt im Popup-Menü ist dann nicht mehr aktiviert). Bei ausgeschalteter Funktion **Größenanpassung an Text** bleiben die Objektgrenzen konstant, und die Anwendung schneidet den eingeschlossenen Text entsprechend zu (oder ab). Wenn nicht gewünscht wird, daß die Anwendung beim Übertragen zwischen Systemen oder Plattformen Text zuschneidet, muß diese Eigenschaft für Labels und Konstanten eingeschaltet bleiben.

Die meisten Sun- und HP-Monitore sind wesentlich größer und haben eine höhere Auflösung als PC- und Macintosh-Monitore. Wenn Sie Sun- oder HP-UX-Benutzer sind, sollten Sie Ihre Frontpanels nicht sehr groß gestalten, wenn sie gut portierbar bleiben sollen.

Zum Erzielen der besten Ergebnisse sollte die Überlappung von Bedienelementen vermieden und etwas Spielraum gelassen werden. Wenn sich ein Label nur teilweise mit einem anderen Objekt überlappt und die Schriftart vergrößert ist, wird dadurch unter Umständen das Bedienelement überlappt.

Bildunterschiede

Der grundlegendste Bildtyp enthält eine Bitmap, eine Reihe von Werten, durch die die Farbe jedes Pixels im Bild bestimmt wird. Komplexere Bilder können eine beliebige Anzahl von Befehlen enthalten, die immer dann ausgeführt werden, wenn das Bild gezeichnet wird, Bilder mit Zeichenbefehlen werden durch Zeichenprogramme oder in der Zeichenschicht einer Grafikanwendung erstellt. Bitmap-basierte Bilder

werden durch Paint-Programme oder in der Paint-Schicht einer Grafikanwendung erstellt.

Bitmaps sind häufig verwendete Speicherformate für Bilder auf allen Plattformen. Wenn Sie Bilder verwenden, die auf Ihren Frontpanels Bitmaps enthalten, sehen die Bilder in der Regel gleich aus, wenn Sie Ihre VIs auf eine andere Plattform übertragen. Bilder mit Zeichenbefehlen enthalten jedoch eventuell einige Befehle, die auf anderen Plattformen nicht unterstützt werden, wie z.B. Abschneiden und Musterfüllung. Diese Bilder sehen dann unter Umständen auf anderen Plattformen eigenartig aus. Überprüfen Sie das Aussehen Ihrer VIs auf einer anderen Plattform, wenn sie dort verwendet werden sollen.

Sie können trotzdem ein Zeichenprogramm oder die Zeichenschicht einer Grafikanwendung einsetzen, um Ihre Bilder zu erstellen. Fügen Sie jedoch das endgültige Bild in ein Paint-Programm oder in die Paint-Schicht einer Grafikanwendung ein, bevor das Bild importiert wird, damit es portabler wird.

(Windows 95/NT und Macintosh) Bei einigen Anwendungen auf Windows 95/NT und vielen Grafikanwendungen auf Macintosh können Sie ein Bild, das nicht rechteckig geformt ist, ausschneiden oder kopieren. In Macintosh, können Sie z.B. ein Lasso-Werkzeug verwenden (siehe Abbildung links), um die Kontur eines Kreises, Dreiecks oder einer komplizierteren Form wie eine Musiknote auszuwählen. Auf anderen Plattformen werden diese unregelmäßigen Formen eventuell auf einem rechteckigen weißen Hintergrund gezeichnet. In Windows 95/NT können Anwendungen eingesetzt werden, die erweiterte Metadateien unterstützen, wenn Bilder mit unregelmäßigen Formen und gut skalierbare Bilder gewünscht werden.



VI-Lokalisierung

Die Strings auf dem Frontpanel können lokalisiert werden, indem sie in eine Tagged-Textdatei exportiert werden, diese Textdatei übersetzt wird und diese Datei wieder in Ihr LabVIEW-Panel importiert werden. Weitere Informationen finden Sie in Kapitel 5, *Drucken und Dokumentieren von VIs*. Wenn Sie diese Textdatei übersetzen, können Sie den VI-Fenstertitel übersetzen (siehe nachstehende Abbildung) oder Sie können den Fenstertitel interaktiv über “VI-Einstellungen” ändern. Weitere Informationen finden Sie im Abschnitt *Bearbeiten von VI-*

Fenstertiteln weiter hinten in diesem Kapitel und dem Abschnitt *Fensteroptionen* in Kapitel 6, *Einrichten von VIs und SubVIs*.



Abbildung 29-1. VI-Lokalisierungsbeispiel

Alle Frontpanel-Objekte können Untertitellabels enthalten. Namen-Labels können nicht übersetzt werden, da das Programm der Sprache G dieses Label zum Kennzeichnen des Objektes verwendet. Namen-Labels können nicht geändert werden, ohne daß dies Auswirkungen auf das Diagramm hat, aber die Untertitel-Label können übersetzt und das Namen-Label anschließend verborgen werden. Weitere Informationen finden Sie im Abschnitt *G-Umgebung* in Kapitel 2, *Bearbeiten von VIs*.

Sie können nicht nur Strings auf dem Frontpanel übersetzen, sondern auch lokalisierte Dezimaltrennzeichen verwenden, wenn Sie Zahlen in Strings konvertieren. Weitere Informationen finden Sie im Abschnitt *Die Dezimaltrennzeichen Punkt und Komma* weiter hinten in diesem Kapitel. Die Funktion "Formatiere Datum-/Zeit-String" zeigt das Datum und die Uhrzeit gemäß Ihrer Eingaben an.

Importieren und Exportieren von VI-Strings

Das Werkzeug zum Exportieren/Importieren von VI-Strings schreibt alle lokalisierbaren Strings im Frontpanel eines VIs in eine Tagged-Textdatei, die als VI-Stringdatei bezeichnet wird. Dazu wählen Sie **Projekt»VI-Strings exportieren** oder **Projekt»VI-Strings importieren** und anschließend innerhalb des Dialogfeldes "Datei" eine Textdatei, in die die VI-Strings exportiert oder importiert werden. Die Software importiert eine VI-Stringdatei nicht, wenn die Datei unbekannte Tags enthält oder wenn ein Tag fehlt.

Folgende Strings können lokalisiert werden.

- VI-Fenstertitel und Beschreibungen
- Untertitel-Labels und Beschreibungen von Objekten
- Freie Labels
- Standarddaten (String, Tabelle, Pfad und Array-Standarddaten)
- Private Daten (Objektnamen in Listenfeldern, Kopfzeilen für Tabellenzeilen und -spalten, Namen von Graphenplots und Namen von graphischen Cursors).

Durch das Importieren und Exportieren von Strings wird außerdem eine Protokolldatei erstellt, in der eine Liste mit Fehlern aufgeführt wird, die während der Operation aufgetreten sind.

Syntax der VI-Stringdatei

Das Format der VI-Stringdatei ähnelt einer HTML-Datei. Jedes Element ist durch ein Start-Tag und ein End-Tag markiert. Ein Start-Tag beginnt mit `<` und endet mit `>`, und ein End-Tag beginnt mit `</` und endet mit `>`. Nicht darstellbare Zeichen werden ignoriert, wenn sie sich nicht im Text befinden. Da das Zeichen `<` den Beginn eines Tags anzeigt, wird `<<` für das Zeichen "kleiner als" im Text verwendet; dementsprechend wird `>>` für das Zeichen "größer als" verwendet. Die doppelten Anführungszeichen werden durch `" "` ersetzt. Außerdem werden Zeilenendzeichen als `<CR>`, `<CRLF>` oder `<LF>` wiedergegeben, die jeweils als Wagenrücklauf, Wagenrücklauf gefolgt von einem Zeilenvorschub und Zeilenvorschub behandelt werden. Dieses Format kann eventuell als schwer lesbar empfunden werden, es ist jedoch in dieser Form maschinenlesbar. Wenn Tags sich ändern oder gelöscht werden, werden beim Import der Datei in die Software Fehlermeldungen ausgegeben.

Tabelle 29-1 listet die VI-Tagtypen und die entsprechende VI-Tag-Syntax auf.

Tabelle 29-1. Beschreibungen der VI-Tags

Typ des VI-Tags	Syntax des VI-Tags
[VI-Stringdatei]	<code><VI [VI-Eigenschaft] > [VI-Info]</code> <code></VI></code>
[VI-Eigenschaft]	<code>syntaxVersion=1 lvVersion=nnn</code> <code>Revision=nnn Name="Text"</code>
[VI-Info]	[VI-Titel] [Beschreibung] [Inhalt]
[VI-Titel]	<code><TITLE>Text</TITLE></code> <code><TITLE><NO_TITLE></TITLE></code>
[Beschreibung]	<code><DESC>Text</DESC></code>
[Inhalt]	<code><CONTENT>[Objekte]</CONTENT></code>

VI-Eigenschaften werden durch ein Leerzeichen voneinander getrennt, und zwischen dem Eigenschaftsnamen und dem darauffolgenden Gleichheitszeichen und zwischen dem Gleichheitszeichen und dem Eigenschaftswert darf kein Leerzeichen vorhanden sein.

Zum Beispiel:

```
<VI syntaxVersion=1 LVversion=4502007 Revision=10
name="AO Signalverlauf erzeugen.vi">

<TITLE>AO Signalverlauf erzeugen.vi</TITLE>

<DESC>Dieses VI erzeugt einen einfach gepufferten
Signalverlauf mit Zeitbegrenzung für den gegebenen
Ausgangskanal zur spezifizierten Updaterate.</DESC>

<CONTENT>

.....

</CONTENT>

</VI>
```

Tabelle 29-2 listet die Tags auf, die den Inhalt des Frontpanels beschreiben, d.h. die freien Labels und zu Objekten zugehörigen Labels, Untertitel-Labels und Eigenschaften.

Tabelle 29-2. Inhalte des Frontpanels

Typ des Content-Tags	Syntax des Content-Tags
[Inhalt]	<CONTENT>[Objekte]</CONTENT>
[Objekte]	([Bedienelement] [Label]) *
[Bedienelement]	<CONTROL [Bedienelement-Eigenschaften]> [Bedienelement-Info] </CONTROL>
[Label]	<LABEL>[Stiltext] </LABEL>
[Stiltext]	<STEXT>Text mit Schriftartinformationen </STEXT>

Zwischen <STEXT> und </STEXT> können Sie Spezifikationen für die Schriftart eingeben. Schriftartinformationen werden mit Hilfe des folgenden Formates kodiert: . Die Schriftarteigenschaften können in beliebiger Reihenfolge aufgelistet werden. Schriftartspezifikationen unterscheiden sich von anderen Elementen, da sie über keine End-Tags verfügen. Ein Untertitel mit dem Text "**Fettgedrucktes Label**" kann z.B. wie folgt beschrieben werden.

```
<LABEL><STEXT><FONT Name="Times New Roman" Größe=12
Stil='F'>Fett <FONT Stil='I'>Label</STEXT></LABEL>
```

Schriftarten können als "predef" definiert werden, um eine der vordefinierten Schriftarten zu spezifizieren (Anwendungsschriftart, Dialogschriftart oder Systemschriftart).

Tabelle 29-3 listet die Tags auf, welche das [Bedienelement] beschreiben.

Tabelle 29-3. Tags für das [Bedienelement]

Typ des Content-Tags	Syntax des Content-Tags
[Bedienelement]	<CONTROL [Bedienelementeigenschaften]> [Bedienelement-Info] </CONTROL>
[Bedienelement- eigenschaften]	ID=xxx Type="Boolesch" Name="Schalter"
[Bedienelement- Info]	[Beschreibung] [Element] [Abschnitt privDaten] [Abschnitt defDaten] [Inhalt]
[Element]	<PARTS> [Element]*</PARTS>
[Element]	<PART [Elementeigenschaften]> [Element-Info] </PART>
[Element- eigenschaften]	partID=nnn partOrder=nnn
[Element-Info]	[Bedienelement] [Label] [MultiLabel]

Nachstehend finden Sie ein Beispiel einer Beschreibung eines Ring-Bedienelements mit dem Untertitel "Ring" und den Optionen "Laden", "Entfernen", "Öffnen" und "Schließen".

```
<CONTROL ID=87 type="Ring" Name="RING-Bedienelement">
<DESC>Ring-Bedienelement</DESC>
<PARTS>
<PART ID=12 Anordnung=0
type="Ringtext"><MLABEL><STRINGS><STRING>Laden</
STRING><STRING>Entfernen</STRING><STRING>Öffnen</
STRING><STRING>Schließen</STRING></STRINGS></MLABEL></
PART>
<PART ID=82 Anordnung=0
type="Untertitel"><LABEL><STEXT><FONT Farbe=FF0033
Größe=12>RING</STEXT></LABEL></PART>
</PARTS>
</CONTROL>
```

Der Tag `MLABEL` (Multilabel - siehe oben) wird dazu verwendet, den Optionsstring auf einem Ring-Bedienelement oder die Strings auf Booleschen Knöpfen festzulegen, und zwar einen String für jeden der vier Zustände. Nachstehend finden Sie eine allgemeine Beschreibung der Tag-Syntax für `MLABEL`.

```
[MultiLabel]<MLABEL> [Mlabel-Info] </MLABEL>
[Mlabel-Info][Schriftart][Strings]
```

Tabelle 29-4 listet die Tags auf, die die Standarddaten für Strings, Tabellen, Arrays und Pfade beschreiben.

Tabelle 29-4. Standarddaten für Strings

Typ des Content-Tags	Syntax des Content-Tags
[defData section]	<DEFAULT> [defData] </DEFAULT>
[defData]	[str def] [table def] [arr data] [path data]
[str def]	[string] <SAME_AS_TEXT>
[table def]	[strings]
[arr data]	<ARRAY nElems=n> [arr element data] </ARRAY>
[arr element data]	clust data [str data] [non-str data]
[str data]	[string]
[non-str data]	<NON_STRING>
[clust data]	<CLUSTER nElems=n> [clust element data] </CLUSTER>
[clust element data]	[clust data] [str data] [non-str data] [arr data] [path data]
[path data]	<PATH type = "absolute"> a<SEP> SYSTEM </PATH>

Für [arr data] muß n [arr element data] dem <ARRAY>Tag folgen. In gleicher Weise muß für [clust data] n [clust element data] vorhanden sein.

Für Standarddaten des String-Bedienelements verwenden Sie einen besonderen Tag, <SAME_AS_TEXT>, der angibt, daß die Standarddaten des Strings mit denen des Labels kStrTextID auf der Elementliste des Strings übereinstimmen. Wenn Sie diesen Tag verwenden, braucht derselbe Text für den Label kStrTextID und die Standarddaten des Strings nicht wiederholt zu werden.

Für die Standarddaten des Pfad-Bedienelements kann das Start-Tag <PATH> über eine Eigenschaft verfügen, den den Pfadtyp spezifiziert. Die möglichen Eigenschaftswerte sind "absolute", "relative", "not-a-path" und "unc". Die Pfadsegmente, die zwischen den Tags <PATH> und

</PATH> auftreten, werden durch das Tag <SEP> begrenzt. Auf der Windows-Plattform wird ein absoluter Pfad c:\windows\temp\temp.txt z.B. wie folgt geschrieben.

```
<PATH type="absolute">c<SEP>windows<SEP>temp<SEP>temp.txt</PATH>
```

Tabelle 29-5 listet die Tags auf, welche private Daten wie beispielsweise Objektnamen von Listenfeldern, Kopfzeilen für Tabellenzeilen und -spalten, die für Tabellenzellen verwendete Schriftart und Graphplotnamen und Cursornamen beschreiben.

Tabelle 29-5. Tag-Beschreibungen für Tabellenzellen, Graphplotnamen und Cursornamen

Typ des Inhalt-Tags	Syntax des Content-Tags
[privData section]	<PRIV> [privData] </PRIV>
[privData]	([items] [col header] [row header] [cell fonts] [plots] [cursors])
[items]	<ITEMS> [string]* </ITEMS>
[col header]	<COL_HEADER> [string]* </COL_HEADER>
[row header]	<ROW_HEADER> [string]* </ROW_HEADER>
[cell fonts]	<CELL_FONTS> [cell font]* </CELL_FONTS>
[plots]	<PLOTS> [string]* </PLOTS>
[cursors]	<CURSORS> [string]* </CURSORS>
[cell font]	[row# col#][font]
[font]	

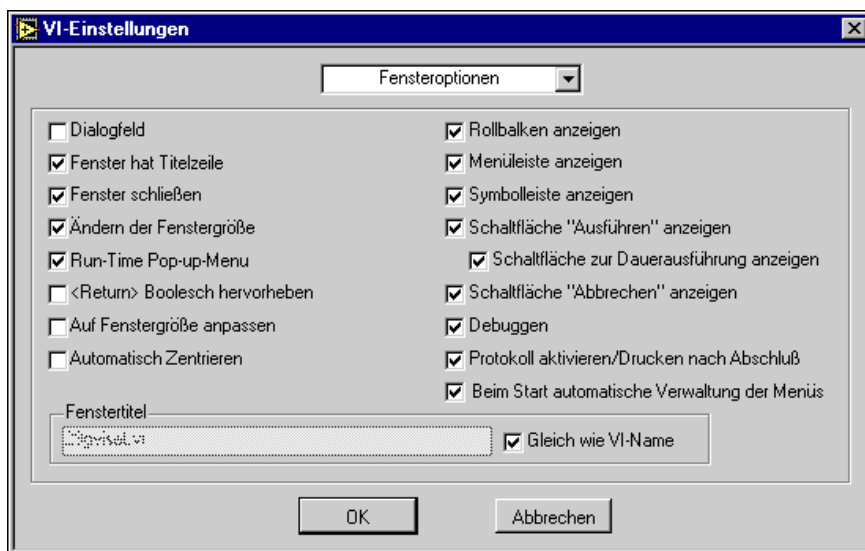
[strings] und [string] haben das folgende Format:

```
[strings]<STRINGS> [string]* </STRINGS>
```

```
[string]<STRING> text </STRING>
```

Bearbeiten von VI-Fenstertiteln

Sie können den VI-Fenstertitel so anpassen, daß er deskriptiver als der VI-Dateiname ist. Diese Funktion ist wichtig für lokalisierte VIs. Der VI-Fenstertitel kann in die Landessprache übersetzt werden, unterliegt keinerlei Benennungsbeschränkungen durch Dateisysteme und wird dennoch durch die VIs erkannt, die als Aufrufende auftreten. Zum Ändern des VI-Fenstertitels während der Bearbeitung des VIs wählen Sie **VI-Einstellungen** und **Fensteroptionen** vom oberen Ring (siehe nachstehende Abbildung).



Zum Ändern von **Fenstertitel** machen Sie die Auswahl von **Gleich wie VI-Name** rückgängig und geben Sie den gewünschten VI-Fenstertitel ein. Zum programmatischen Ändern des VI-Fenstertitels siehe Kapitel 21, [VI-Server](#).

Die Dezimaltrennzeichen Punkt und Komma

Durch die Auswahl von **Frontpanel** aus dem Dropdown-Menü im Dialogfeld **Bearbeiten»Voreinstellungen** können Sie die Verwendung des Dezimaltrennzeichens des Systems oder die Verwendung eines Punkts einstellen. Als Alternative dazu können Sie mit den folgenden Funktionen einen Punkt als Dezimaltrennzeichen zwingend vorschreiben, wenn Zahlen in Strings umgewandelt werden oder umgekehrt.

- In technisches Format
- In Dezimalbruch
- In exponentielles Format
- Von Exp./Wiss./Tech.

Formatiere Datum-/Zeit-String

Sie können das Anzeigeformat für Datum und Zeit einstellen, indem Sie die Funktion “Formatiere Datum-/Zeit-String” verwenden. Informationen über diese Funktion finden Sie in der *Online-Referenz*, die online über **Hilfe»Online-Referenz** verfügbar ist.

Portieren zwischen Anwendungen in der Sprache G

Alle LabVIEW-VIs, die noch das dazugehörige Blockdiagramm enthalten, können in BridgeVIEW konvertiert werden, wenn die Version der Sprache G in LabVIEW und BridgeVIEW kompatibel ist. Die Konvertierung von VIs von BridgeVIEW in LabVIEW ist auch möglich, jedoch lassen nicht alle VIs konvertieren.

LabVIEW in BridgeVIEW

Der wichtigste Faktor, der beim Konvertieren von VIs von LabVIEW in BridgeVIEW berücksichtigt werden muß, ist die Version der Programmiersprache G, die zur Erstellung des VIs verwendet wurde. Beziehen Sie sich auf die nachstehende Tabelle. Alle LabVIEW VIs, die mit LabVIEW 3.x bis LabVIEW 4.x erstellt wurden, können in eine beliebige Version von BridgeVIEW geladen werden, vorausgesetzt die VIs verfügen noch über die dazugehörigen Diagramme. VIs, die mit LabVIEW 5.0.x erstellt wurden, können nur von BridgeVIEW 2.0 oder höher geladen werden.

LabVIEW-Version	G-Version	BridgeVIEW-Version	G-Version
beliebige Version bis zu 4.0.1	beliebige Version bis zu 4.0.1	1.0	4.1
4.1	4.0.2	1.0.1	4.1.1
4.1.1	4.0.3	1.1	4.1.2
5.0	5.0	2.0	5.0

BridgeVIEW in LabVIEW

Es gibt zwei Faktoren, die bestimmen, ob ein BridgeVIEW-VI mit LabVIEW geladen und ausgeführt werden kann. Zunächst einmal müssen die Versionen der Sprache G kompatibel sein. Daher müssen BridgeVIEW-VIs zu diesem Zeitpunkt durch LabVIEW 5.0 oder höher geladen werden, da G 4.0.x keine VIs laden kann, die durch G 4.1.x oder höher erstellt wurden. Der zweite Faktor, der bestimmt, ob ein VI von BridgeVIEW in LabVIEW konvertiert werden kann, ist die Frage, ob das VI andere VIs enthält, die für BridgeVIEW spezifisch sind. Alle BridgeVIEW-VIs, die Funktionen verwenden, die für BridgeVIEW spezifisch sind, können in LabVIEW eventuell nicht geladen oder richtig bedient werden. Zum Beispiel ist der Tag-Datentyp für BridgeVIEW spezifisch. Alle VIs, die den Tag-Datentyp verwenden oder das BridgeVIEW-Programm aufrufen, sind in LabVIEW nicht einsetzbar.

Wenn Sie vorhaben, zur Entwicklung von VIs BridgeVIEW zu verwenden und diese dann in LabVIEW zu konvertieren, wählen Sie den Palettensatz Basic G oder LabVIEW. Dieser Palettensatz zeigt dieselben Funktionen an wie der Palettensatz LabVIEW. VIs, die nur mit den Funktionen in diesen Paletten erstellt wurden, sind zwischen BridgeVIEW und LabVIEW portierbar.

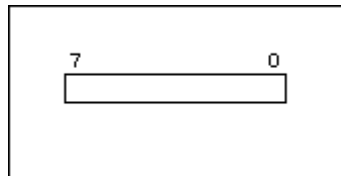
Datenspeicherformate

Dieser Anhang beschreibt die Formate, in denen Sie Daten speichern können. Diese Informationen sind besonders für fortgeschrittene Benutzer nützlich, d.h. für diejenigen, die Code Interface Nodes (CINs) verwenden und von Dateien lesen und zu Dateien schreiben, die von den Datei I/O-Funktionen verwendet werden. In diesem Anhang wird erklärt, wie Daten gespeichert werden; es wird die Beziehung von Typ-Deskriptoren zur Datenspeicherung erläutert sowie die Methode behandelt, mit der Daten zur Speicherung auf einem Speichermedium gewandelt werden.

Datenformate für Frontpanel-Bedien- und Anzeigeelemente

Boolesche Ausdrücke

Boolesche Ausdrücke werden als 8-Bit-Werte gespeichert. Wenn der Wert Null ist, ist der Boolesche Ausdruck FALSE. Alle Werte, die nicht Null sind, stellen TRUE dar.

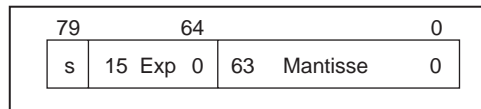


Numerische Elemente

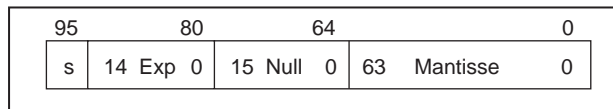
Extended

Wenn Extended-Zahlen auf einem Speichermedium gespeichert werden, werden sie in einem plattformunabhängigen 128-Bit-Format gespeichert, was dem UNIX-Speicherformat entspricht. Im Speicher hängen Größe und Präzision von der Plattform ab.

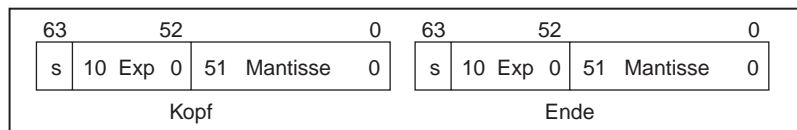
(Windows) Extended-Fließkommazahlen haben ein 80-Bit-Format (80287 Extended-Format).



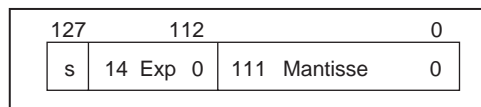
(68K Macintosh) Extended-Fließkommazahlen haben ein 96-Bit-Format (MC68881-MC68882 Extended-Format).



(Power Macintosh) Extended-Fließkommazahlen werden als zwei kombinierte Double-Fließkommazahlen dargestellt, d.h. im Double-Double-Format von Apple.



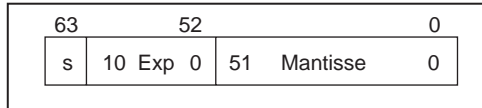
(Sun) Extended-Fließkommazahlen haben ein 128-Bit-Format.



(HP-UX) Extended-Fließkommazahlen werden als Double-Fließkommazahlen dargestellt (siehe nachstehende Abbildung).

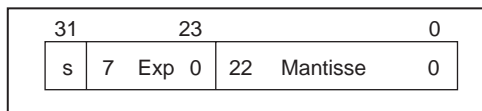
Double

Double-Fließkommazahlen haben ein 64-Bit IEEE-Double-Format (Formatstandard).



Single

Single-Fließkommazahlen haben ein 32-Bit IEEE-Single-Format.



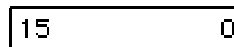
Long-Integer-Wert

Long-Integer-Zahlen haben ein 32-Bit-Format und sind vorzeichenbehaftet oder vorzeichenlos.



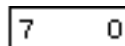
Word-Integer-Wert

Word-Integer-Zahlen haben ein 16-Bit-Format und sind vorzeichenbehaftet oder vorzeichenlos.



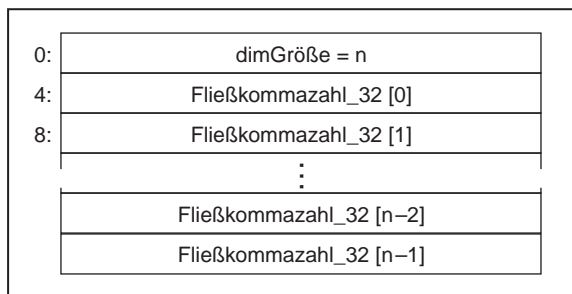
Byte-Integer-Wert

Byte-Integer-Zahlen haben ein 8-Bit-Format und sind vorzeichenbehaftet oder vorzeichenlos.

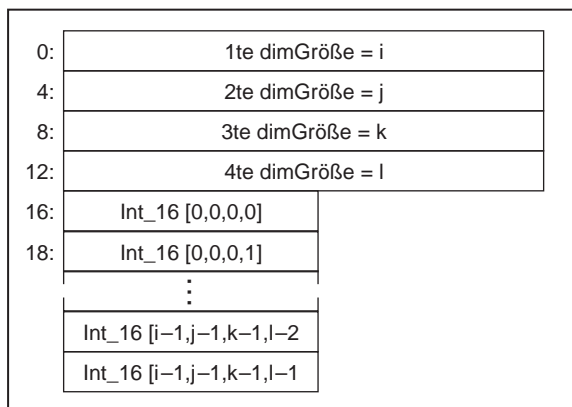


Arrays

Arrays werden so gespeichert, daß die Größe jeder Dimension des Arrays in Long-Integer-Werten, gefolgt von den Daten ausgedrückt wird. Aufgrund von Ausrichtungsbeschränkungen bei bestimmten Plattformen können auf die Dimensionsgröße einige Byte zur Auffüllung folgen, so daß das erste Element der Daten richtig ausgerichtet ist. Wenn Sie ein LabVIEW-Benutzer sind, finden Sie weitere Informationen hierzu im Abschnitt *Alignment Considerations* von Kapitel 2, *CIN Parameter Passing* des *LabVIEW Code Interface Reference Manuals*. Dieses Dokument ist auf Ihren Programmdisketten oder der Programm-CD nur als portables Dokumentformat (PDF) erhältlich. Das folgende Beispiel zeigt ein eindimensionales Array aus Single-Fließkommazahlen. Die Dezimalzahlen links stellen die Byte-Offsets der Speicherstellen dar, von denen das Array beginnt.

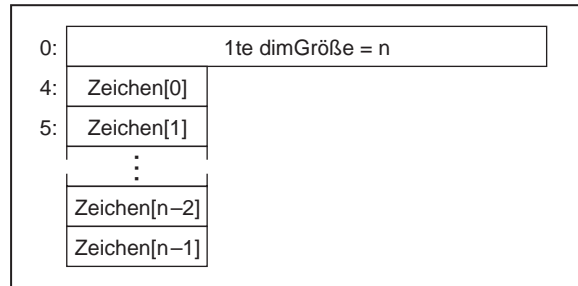


Diese Abbildung zeigt ein vierdimensionales Array aus Word-Integer-Werten.



Strings

Strings werden so gespeichert, als seien sie eindimensionale Arrays aus Byte-Integer-Werten (8-Bit-Zeichen).



Pfade

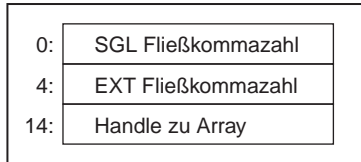
Bei Pfaden werden der Pfadtyp und die Anzahl der Pfadkomponenten in Word Integer-Werten gespeichert, auf die die Pfad-Komponenten unmittelbar folgen. Der Pfadtyp 0 steht für einen absoluten Pfad und 1 für einen relativen Pfad. Alle anderen Werte für den Pfadtyp geben an, daß der Pfad ungültig ist. Jede Pfad-Komponente ist ein Pascal String (P-String), in dem das erste Byte die Länge des P-String in Byte darstellt (ohne das Byte für Länge).

Cluster

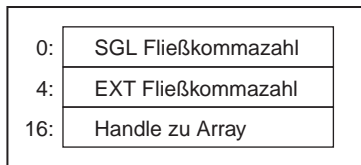
Ein Cluster speichert Elemente mit unterschiedlichen Datentypen gemäß der *Cluster-Anordnung*. Skalardaten werden direkt im Cluster gespeichert. Arrays, Strings, Handles und Pfade werden indirekt gespeichert. Der Cluster speichert ein Handle, das zum Speicherbereich weist, in welchem die Daten tatsächlich gespeichert sind. Aufgrund von Ausrichtungsbeschränkungen gewisser Plattformen folgen auf die Dimensionsgröße eventuell einige Byte zur Auffüllung, damit das erste Element der Daten richtig ausgerichtet ist. Wenn Sie ein LabVIEW-Benutzer sind, finden Sie weitere Informationen im Abschnitt *Alignment Considerations* von Kapitel 2, *CIN Parameter Passing*, des *LabVIEW Code Interface Reference Manuals*. Dieses Dokument ist auf Ihren Programm-Disketten oder der Programm-CD nur im PDF-Format (portablen Dokumentformat) erhältlich.

Folgende Abbildungen zeigen ein Cluster mit einer Single-Fließkommazahl, einer Extended-Fließkommazahl und einem Handle zu einem eindimensionalen Array aus vorzeichenlosen Word-Integer-Werten, und zwar in dieser Reihenfolge. Weitere Informationen finden Sie im Kapitel 14, *Bedien- und Anzeigeelemente vom Typ Array und Cluster*.

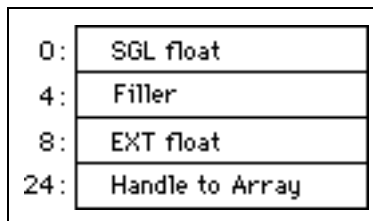
- **(Windows)**



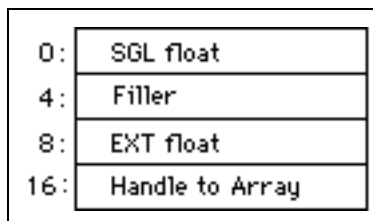
- **(Macintosh)**



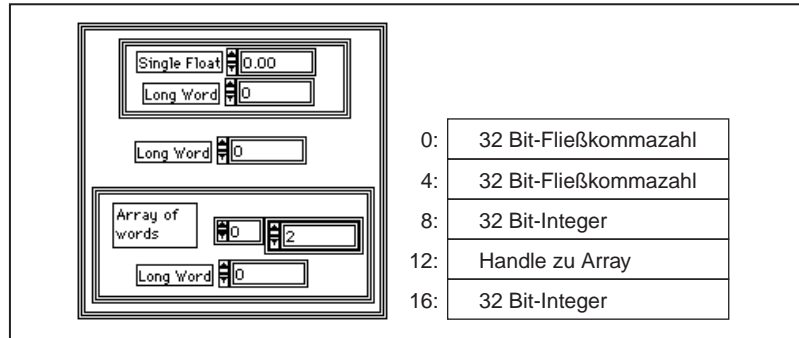
- **(Sun)**



- **(HP-UX)**



Im nächsten Beispiel werden die eingebetteten Cluster nicht indirekt gespeichert. Statt dessen werden die Daten direkt in den eingebetteten Clustern gespeichert, als ob die Daten nicht im Subcluster eingebettet wären. Nur Arrays, Strings und Handles werden indirekt gespeichert.



Typ-Deskriptoren

Jede Verbindung und jedes Terminal im Blockdiagramm ist mit einem Datentyp verknüpft. Sie können diesen Typ mit einer Struktur im Speicher überwachen, die als *Typ-Deskriptor bezeichnet wird*. Dieser Deskriptor ist ein String aus Word-Integer-Werten, der jeden Datentyp in G beschreiben kann. Numerische Werte werden in hexadezimaler Form geschrieben, wenn es nicht anders angegeben ist.

Das allgemein gültige Format eines Typ-Deskriptors ist: `<length>`
`<type code>`.

Einige Typ-Deskriptoren haben zusätzliche Informationen, die auf den Bezeichner folgen. Arrays und Cluster sind strukturierte Datentypen oder Datentypen mit mehreren Elementen, da sie andere Typen enthalten. Der Clustertyp enthält z.B. zusätzliche Informationen über den Typ der einzelnen dazugehörigen Elemente.

Das erste Wort (16 Bit) in einem beliebigen Typ-Deskriptor ist die Länge dieses Typ-Deskriptors in Byte (einschließlich des Längenwortes). Das zweite Wort (16 Bit) ist der Bezeichner. Das Byte der höheren Ordnung des Bezeichners sind für den internen Gebrauch reserviert. Beim Vergleich von zwei Typ-Deskriptoren ist es ratsam, dieses Byte zu ignorieren; zwei Deskriptoren sind gleich, selbst wenn die Byte der höheren Ordnung von Bezeichnern es nicht sind.

Der Bezeichner kodiert die tatsächlichen Typeninformationen wie Single- oder Extended-Fließkommazahl (siehe nachstehende Tabelle). Diese Bezeichnerwerte können sich in zukünftigen Versionen ändern. Die Buchstaben *xx* im Typ-Deskriptor stellen reservierte Werte dar. Sie können ignoriert werden.

Datentypen

In folgenden Tabellen werden numerische und nicht numerische Skalar-Datentypen, Bezeichner und Typ- Deskriptoren aufgeführt.

Tabelle A-1. Numerische Skalar-Datentypen

Datentyp	Bezeichner (Zahlen in Hexadezimal- form)	Typ-Deskriptor (Zahlen in Hexadezimalform)
Byte-Integer-Wert	01	0004 <i>xx</i> 01
Word-Integer-Wert	02	0004 <i>xx</i> 02
Long-Integer-Wert	03	0004 <i>xx</i> 03
Vorzeichenloser Byte-Integer-Wert	05	0004 <i>xx</i> 05
Vorzeichenloser Word-Integer-Wert	06	0004 <i>xx</i> 06
Vorzeichenloser Long-Integer-Wert	07	0004 <i>xx</i> 07
Single-Fließkommazahl	09	0004 <i>xx</i> 09
Double-Fließkommazahl	0A	0004 <i>xx</i> 0A
Extended-Fließkommazahl	0B	0004 <i>xx</i> 0B
Komplexe Single-Fließkommazahl	0C	0004 <i>xx</i> 0C
Komplexe Double-Fließkommazahl	0D	0004 <i>xx</i> 0D
Komplexe Extended-Fließkommazahl	0E	0004 <i>xx</i> 0E
Enumerated Byte-Integer-Wert	15	<nn> <i>xx</i> 15 <k> <k pstrs>
Enumerated Word-Integer-Wert	16	<nn> <i>xx</i> 16 <k> <k pstrs>
Enumerated Long-Integer-Wert	17	<nn> <i>xx</i> 17 <k> <k pstrs>

Tabelle A-1. Numerische Skalar-Datentypen (Fortsetzung)

Datentyp	Bezeichner (Zahlen in Hexadezimal- form)	Typ-Deskriptor (Zahlen in Hexadezimalform)
Single Physikalische Menge	19	<nn> xx19 <k> <k base-exp>
Double Physikalische Menge	1A	<nn> xx1A <k> <k base-exp>
Extended Physikalische Menge	1B	<nn> xx1B <k> <k base-exp>
Komplexe Single Physikalische Menge	1C	<nn> xx1C <k> <k base-exp>
Komplexe Double Physikalische Menge	1D	<nn> xx1D <k> <k base-exp>
Komplexe Extended Physikalische Menge	1E	<nn> xx1E <k> <k base-exp>

Tabelle A-2. Nicht-Numerische Datentypen

Datentyp	Bezeichner (Zahlen in Hexadezimal- form)	Typ-Deskriptor (Zahlen in Hexadezimalform)
Boolescher Wert	21	0004 xx21
String	30	0008 xx30 <len>
Handle	31	0006 xx31 <kind>
Pfad	32	0008 xx32 <len>
Bild	33	0008 xx33 <len>
Array	40	<nn> xx40 <k> <k dims> <element type descriptor>
Cluster	x50	<nn> xx50 <k> <k element type descriptors>

Der Mindestwert im Größenfeld eines Typ-Deskriptors ist 4 (siehe Tabelle A-1). Es kann jedoch bei allen Typ-Deskriptoren ein Name (ein Pascal-String) angehängt werden, wobei die Größe um die Länge des Namens größer ist (aufgerundet auf ein Mehrfaches von 2).

Die Datentypen für String, Pfad und Bild haben eine Länge von 32 Bit (ähnlich wie die Array-Dim-Größe), auch wenn der einzige Wert, der momentan kodiert ist, FFFFFFFF (-1) ist, was auf variable Größen hinweist. Die tatsächliche Länge wird mit den Daten gespeichert.

Beachten Sie, daß die Datentypen für Array und Cluster jeweils über ihre eigenen Bezeichner verfügen. Außerdem enthalten sie zusätzliche Informationen über ihre Dimensionalität (für Arrays) oder die Anzahl der Elemente (für Cluster) sowie Informationen über die Datentypen ihrer Elemente.

Im folgenden Beispiel eines Enumerated Byte-Integer-Wertes für die Objekte `am`, `fm`, `fm stereo` stellt jede Gruppe mit Zeichen ein Wort mit 16 Bit dar. Der in Anführungszeichen aufgeführte Leerschritt (" ") stellt einen ASCII-Leerschritt dar.

```
0016 0015 0003 02a m02 fm 09f m" " st er eo
```

0016 gibt einen Gesamtwert von 22 Byte an. 0015 gibt einen Enumerated Byte-Integer-Wert an. 0003 gibt an, daß drei Objekte vorhanden sind.

Im folgenden Beispiel einer Double Physikalischen Menge mit den Einheiten `m/s` stellt jede Gruppe ein Wort mit 16 Bit dar.

```
000E 001A 0002 0002 FFFF 0003 0001
```

000E gibt einen Gesamtwert von 14 Byte an. 0x1A gibt an, daß es sich um eine Double-Zahl mit Einheiten handelt. 0002 gibt zwei Basis-Exponentenpaare an. 0002 bezeichnet den Sekundenbasis-Index. FFFF (-1) ist der Exponent der Sekunden. 0003 bezeichnet den Meterbasis-Index. 0001 ist der Exponent der Meter.



Hinweis

Alle physikalischen Mengen werden intern als Basiseinheiten gespeichert, und zwar unabhängig davon, als welche Einheit sie angezeigt werden. Tabelle 9-2, Basiseinheiten, zeigt die neun Basen, die durch Indizes 0-8 für Radiant-Candela dargestellt werden.

Array

Der Bezeichner für ein Array ist 0x40. Unmittelbar nach dem Bezeichner befindet sich ein Wort mit der Anzahl von Dimensionen des Arrays. Dann enthält ein Long-Integer-Wert für jede Dimension die Größe dieser Dimension in Elementen. Und schließlich erscheint der Typ-Deskriptor für das Element nach jeder der Dimensionsgrößen. Der Elementtyp kann jeder Typ mit Ausnahme eines Arrays sein. Die Dimensionsgröße für eine Dimension kann FFFFFFFF (-1) sein. Dies bedeutet, daß die

Array-Dimensionsgröße variabel ist. Momentan sind alle Arrays von variabler Größe. Die tatsächliche Dimensionsgröße wird mit den Daten gespeichert und ist stets größer als oder gleich Null. Nachstehend finden Sie einen Typ-Deskriptor für ein eindimensionales Array aus Double-Fließkommazahlen:

```
000E 0040 0001 FFFF FFFF 0004 000A
```

000E ist die Länge des gesamten Typ-Deskriptors einschließlich des Typ-Deskriptors für das Element. Das Array ist von variabler Größe, so daß die Dimensionsgröße FFFFFFFF ist. Beachten Sie, daß der Typ-Deskriptor für das Element (0004 000A) genauso erscheint wie für ein Skalar desselben Typs.

Nachstehend finden Sie ein Beispiel eines Typ-Deskriptors für ein zweidimensionales Array aus Booleschen Werten.

```
0012 0040 0002 FFFF FFFF FFFF FFFF 0004 0021
```

Cluster

Der Bezeichner für ein Cluster ist 0x50. Unmittelbar nach dem Bezeichner befindet sich ein Wort mit der Anzahl von Objekten im Cluster. Nach diesem Wort befindet sich der Typ-Deskriptor für jedes Element in der *Cluster-Anordnung*. Betrachten wir einmal ein Cluster aus zwei Integer-Werten: ein vorzeichenbehafteter Word-Integer-Wert und ein vorzeichenloser Long-Integer-Wert:

```
000E 0050 0002 0004 0002 0004 0007
```

000E ist die Länge des Typ-Deskriptors einschließlich der Element-Typ-Deskriptoren.

Nachstehend finden Sie einen Typ-Deskriptor für einen Multiplot-Graphen (die numerischen Typen können unterschiedlich ausfallen):

```
0028 0040 0001 FFFF FFFF...1D Array aus
001E 0050 0001...1 Komponenten-Cluster aus
0018 0040 0001 FFFF FFFF...1D-Array aus
000E 0050 0002...2 Komponenten-Cluster aus
0004 000A... Double-Fließkommazahl
0004 0003... Long-Integer-Wert
```

Geglättete Daten

Mit zwei internen Funktionen werden Daten vom Format im Speicher in ein passenderes Format zum Schreiben in eine Datei bzw. zum Lesen von einer Datei konvertiert.

Da Strings und Arrays in Handle-Blöcken gespeichert werden, sind Cluster mit diesen Typen *strukturunabhängig*. Im allgemeinen werden Daten in *freier* Form gespeichert. Ein Cluster mit einer Double-Fließkommazahl und einem String wird z.B. als Fließkommazahl mit 8 Byte, gefolgt von einem Handle mit 4 Byte zum String gespeichert. Die String-Daten werden nicht an der angrenzenden Speicherstelle der Extended-Fließkommazahl gespeichert. Daher müssen Sie die Daten von zwei unterschiedlichen Stellen holen, wenn Sie die Clusterdaten auf ein Speichermedium schreiben möchten. Natürlich könnten bei einem arbiträr komplexen Datentyp die Daten an vielen verschiedenen Stellen gespeichert werden.

Wenn Daten in einer VI-Datei oder Datenprotokoll-Datei gespeichert werden, werden die Daten mit der Funktion *In Binärstring wandeln* in einen einzelnen String *konvertiert*, bevor sie gespeichert werden. Auf diese Art werden selbst Daten aus einem arbiträr komplexen Cluster strukturabhängig gemacht; sie werden somit nicht in mehreren Teilen gespeichert. Wenn Ihre G-Entwicklungsumgebung eine solche Datei von einem Speichermedium lädt, muß sie diese Operation umgekehrt durchführen — sie muß einen einzelnen String lesen und ihn in seine interne, möglicherweise strukturunabhängige Form *zurückwandeln* (von *Binärstring wandeln*).

Die geglätteten Daten werden in eine Standardform *normalisiert*, so daß die Daten in unveränderter Form von VIs verwendet werden, die auf einer beliebigen Plattform ausgeführt werden. Dabei werden numerische Daten in *big endian*-Form gespeichert (das most-significant Byte zuerst), und Extended-Fließkommazahlen werden als 16-Byte-Mengen unter Verwendung des Extended-Formats von Sun gespeichert, das weiter vorne in diesem Abschnitt beschrieben wurde.



Hinweis

Wenn Daten in eine Datei geschrieben werden, die zum Einsatz in einer Anwendung gedacht ist, die nicht mit Hilfe G erstellt wurde, müssen Sie die Daten nach der Wandlung eventuell transformieren. Gleichmaßen müssen Sie Ihre Daten eventuell transformieren, bevor Sie “Von Binärstring wandeln” durchführen, wenn Daten von einer Datei gelesen werden, die aus einer Anwendung stammt, die nicht mit G erstellt wurde. Andere Windows-Anwendungen erwarten normalerweise, daß numerische Daten in der little endian-Form sind (das least-significant Byte zuerst). Andere Windows und

Macintosh-Anwendungen erwarten normalerweise, daß Extended-Fließkommazahlen jeweils im zuvor beschriebenen 80-Bit- bzw. 96-Bit-Format sind.

Mit den Blockdiagrammfunktionen “In Binärstring wandeln” und “Von Binärstring wandeln” (die in der **Online-Hilfe** durch die Auswahl von **Hilfe»Online-Referenz** verfügbar sind) können Sie die Funktionen “In Binärstring wandeln” und “Von Binärstring wandeln” bei Daten durchführen, so wie es Ihre G-Entwicklungsumgebung intern beim Speichern und Laden von Daten tut.

Skalare

Die geglättete Form eines beliebigen numerischen Typs sowie des Booleschen Typs enthält nur die Daten im big endian-Format. Ein Long-Integer-Wert mit dem Wert -19 wird z.B. als `FFFF FFED` kodiert. Eine Double-Fließkommazahl mit einem Wert, der $\frac{1}{4}$ entspricht, ist `3FD0 0000 0000 0000`. Ein Boolescher TRUE-Wert ist ein beliebiger Wert, der nicht Null ist. Ein Boolescher FALSE-Wert ist `00`.

Die Dateiform für Extended-Zahlen basiert auf der SPARC Quadruple-Precision-Form, die ein 16-Bit Biased-Exponent ist, gefolgt von einer Mantisse, die 112 Bit lang ist. In dieser Form ist das Bit links vom Binärpunkt in der Mantisse ein implizites Bit (und wird stets als solches angesehen), das in der Darstellung nicht erscheint.

Strings, Handles und Pfade

Da Strings, Handles und Pfade größenveränderlich sind, geht der geglätteten Form ein normalisierter Long-Integer-Wert voraus, der ihre Länge in Byte aufzeichnet. Bei Pfaden gehen dieser Länge vier Zeichen voraus: `PTH0`. Ein Stringtyp mit dem Wert `ABC` wird z.B. zu `0000 0003 4142 43` gewandelt.

Das geglättete Format ähnelt dem Format des Strings im Speicher. Handles und Pfade haben jedoch keinen Längenwert, der ihnen vorausgeschickt wird, wenn sie gespeichert werden, so daß dieser Wert von der tatsächlichen Größe der Daten im Speicher abgeleitet und bei der Wandlung von Daten vorangestellt wird.

Arrays

Den Daten für ein geglättetes Array werden normalisierte Long-Integer-Werte vorangestellt, die die Größe aller Dimensionen der Arrays in Elementen aufzeichnen. Die am langsamsten variierende Dimension kommt zuerst, gefolgt von darauffolgenden, schneller variierenden Dimensionen in der entsprechenden Reihenfolge, und zwar so, wie die Dimensionsgrößen gespeichert sind. Die Daten folgen unmittelbar auf diese Dimensionsgrößen in derselben Reihenfolge, in der sie gespeichert wurden. Außerdem werden diese Daten gewandelt, wenn es erforderlich ist. Nachstehend finden Sie ein Beispiel eines zweidimensionalen Arrays aus sechs 8-Bit-Integer-Werten.

```
{ {1, 2, 3}, {4, 5, 6} } wird gespeichert als 0000 0002
0000 0003 0102 0304 0506.
```

Nachstehend finden Sie ein Beispiel eines konvertierten eindimensionalen Arrays aus Booleschen Variablen:

```
{T, F, T, T} wird gespeichert als 0000 0004 0100 0101.
(01 ist der bevorzugte Wert für TRUE.)
```

Cluster

Ein geglätteter Cluster ist die Verknüpfung der geglätteten Daten seiner Elemente (in Cluster-Anordnung). Ein geglätteter Cluster eines Word-Integers mit dem Wert 4 (Dezimalwert) und eines Long-Integers mit dem Wert 12 ist 0004 0000 000C.

Ein geglätteter Cluster aus einem String ABC und einem Word-Integer mit dem Wert 4 ist 0000 0003 4142 4300 04.

Ein konvertierter Cluster eines Word-Integers mit dem Wert 7, ein Cluster eines Word-Integers mit dem Wert 8 und ein Word-Integer mit dem Wert 9 ist 0007 0008 0009.

Zur Durchführung der Funktion “Von Binärstring wandeln” bei diesen Daten müssen Sie nur das umgekehrte Verfahren einsetzen. Die geglättete Form eines Datenteils kodiert *nicht* den Datentyp; dafür ist der Typ-Deskriptor erforderlich. Die Funktion “Von Binärstring wandeln” erfordert es, daß ein Datentyp als Eingang verbunden wird, so daß die Funktion den String richtig dekodieren kann.

Häufig gestellte Fragen zu G

Dieser Anhang enthält Antworten auf einige Fragen, die häufig von Benutzern von G gestellt werden.

Diagramme und Graphen

Wie verbinde ich Daten mit einem Diagramm oder Graphen?

Rufen Sie das Hilfe-Fenster auf und bewegen Sie das Verbindungswerkzeug über ein Graph-Terminal im Blockdiagramm, um eine kurze Beschreibung über die Verbindung von grundlegenden Graphentypen anzeigen zu lassen. Im Verzeichnis `examples\general\graphs` gibt es sehr gute Beispiele zu Graphen und Diagrammen.

Was ist der wesentliche Unterschied zwischen Diagrammen und Graphen?

Diagramme und Graphen unterscheiden sich in der Art und Weise, wie in ihnen Daten angezeigt und aktualisiert werden. VIs mit Graphen sammeln die Daten normalerweise in einem Array und plotten sie anschließend auf dem Graphen, ähnlich wie ein Spreadsheet, das die Daten zuerst speichert und dann ein Plot von ihnen erstellt. Im Gegensatz dazu hängt ein Diagramm den bestehenden Datenpunkten in der Anzeige neue Datenpunkte an. In dieser Weise sehen Sie die aktuellen Ablesungen oder Messungen im Zusammenhang mit zuvor erfaßten Daten. Die Länge des Diagramm-Historie-Puffers kann mit Hilfe einer Popup-Option auf dem Diagramm eingestellt werden. Natürlich ist es auch möglich, einen Historie-Puffer mit Graph-Anzeigeelementen zu implementieren; dies muß jedoch im Blockdiagramm erfolgen. Diese Funktionen sind bereits in das Diagramm integriert.

Wie kippe ich die Achsen im Diagramm oder Graphen?

Zum Kippen der Skalen für die X-Achse oder Y-Achse verwenden Sie das Attribut X Flipped oder Y Flipped. Wenn das Attribut X Flipped auf TRUE eingestellt ist, wird das Minimum der X-Achsenskala auf rechts und das Maximum auf links eingestellt. Dementsprechend wird, wenn die

Eigenschaft `Y Flipped` auf `TRUE` eingestellt ist, das Minimum der Y-Achsen-skala auf oben und das Maximum auf unten eingestellt.

Wie richte ich ein Diagramm oder einen Graphen so ein, daß Informationen im Zeitformat angezeigt werden?

Graphen und Diagramme können numerische Informationen im relativen oder absoluten Zeitformat anzeigen. Verwenden Sie die Popup-Option **X-Skala»Formatieren...** oder **Y-Skala»Formatieren...**, um das X-Achsen- oder Y-Achsenformat auf **Relative Zeit** einzustellen.

Wie richte ich die X-Achse eines Diagramms so ein, daß Echtzeit angezeigt wird?

Siehe das Beispiel-VI mit der Bezeichnung `Real-Time Chart.vi`, das sich in `examples\general\graphs\charts.llb` befindet. Es schlüsselt das Cluster Datum/Zeit von der Funktion "Sekunden in Datum/Zeit" in "Stunden, Minuten und Sekunden" auf. Dann werden diese Zahlen in die Anzahl von Sekunden umgewandelt, die seit Mitternacht verstrichen sind. Schließlich wird diese Zahl als Eingabe für die Eigenschaft `Xo` und `Delta X` eines Diagramms verwendet, wobei die Option **X-Skala»Formatieren** auf **Relative Zeit** eingestellt wird. Oder Sie können **Absolute Zeit** in der Option **Format & Präzision** verwenden.

Wie kann ich ein Diagramm programmatisch löschen?

Verbinden Sie ein leeres Array mit der Eigenschaft "Diagrammlegende". Der Datentyp dieses leeren Arrays ist derselbe wie der Datentyp, der mit dem Diagramm verbunden ist. Ein sehr gutes Beispiel, durch welches diese Technik verdeutlicht wird, befindet sich in `examples\general\graphs\charts.llb\How to Clear Charts & Graphs.vi`.

Wie vermeide ich ein Blinken des Graphen bei jeder Aktualisierung?

Verwenden Sie die Option **Graphik glätten**, um zu verhindern, daß Graphen blinken. Diese Option befindet sich im Untermenü **Datenoperationen** des Popup-Menüs "Graph". Mit der Option **Graphik glätten** wird der Graph in einem nicht am Bildschirm angezeigten Puffer neu gezeichnet, bevor das Bild in die Anzeige kopiert wird. Mit dieser Option wird zwar eine geglättete Graphik für die Bedienelemente erzeugt, sie ist jedoch im allgemeinen langsamer und erfordert mehr Speicherplatz.

Die Option **Graphik glätten** kann global mit dem Dialogfeld **Bearbeiten»Voreinstellungen»Frontpanel** eingestellt werden.

Für individuelle Graphen kann sie von den jeweiligen Popup-Menüs ein- oder ausgeschaltet werden.

Wie aktualisiere ich einen Graphen, ohne ihn zu löschen?

Alle Graphen (Kurvenform, XY und Intensität) werden stets gelöscht, bevor neue Daten in sie geschrieben werden; es ist jedoch einfach, die zuvor zum Graphen geschriebenen Daten zu behalten und bei jedem Schreiben neue Daten anzuhängen. Das Beispiels-VI `examples\general\arrays.llb\Separate Array Values.vi` demonstriert die Verwendung der Funktion "Array erstellen", um einem Array neue Werte anzuhängen. Wenn Sie LabVIEW-Benutzer sind, siehe Kapitel 5, *Arrays, Cluster und Graphen* im *LabVIEW Benutzerhandbuch*.

Wie erstelle ich einen Balkengraphen?

Der Historie-Menüpunkt **Allgemeine Plots** listet Balkenplots als eine Stilart auf. Die anderen Plotattribute ermöglichen die Verwendung einer unterschiedlichen Basislinie und horizontaler Balkenplots. Siehe auch `examples\general\graphs\bargraph.llb`. Diese VI-Bibliothek enthält einen Muster-Balkengraphen und zwei VIs zur Erstellung von Balkengraphen mit Hilfe eines Arrayeingangs. Außerdem können Balkengraphen mit dem G Picture Control Toolkit erstellt werden.

Wie kann ich Polarplots und Smith-Diagramme erstellen?

Das G Picture Control Toolkit enthält Beispiele mit Routinen zur Erstellung von Polarplots und Smith-Diagrammen. Das Toolkit ist ein vielseitig verwendbares Graphikpaket zur Erstellung von arbiträren Frontpanel-Anzeigen. Die Bild-VI-Bibliothek, die im Lieferumfang des Toolkits enthalten ist, implementiert einen allgemeinen Satz mit zweidimensionalen Zeichenbefehlen zum Erstellen von graphischen Bildern.

Wie erstelle ich ein Label für die Y-Achse, die um 90 Grad gedreht ist?

BridgeVIEW und LabVIEW können Text nicht drehen. Zur Verwendung von gedrehtem Text müssen Sie diesen zunächst in einer Anwendung erstellen, die über diese Fähigkeit verfügt, z.B. Windows Paintbrush, und ihn dann in einem passenden Format speichern (weitere Informationen über das Importieren von Graphiken finden Sie im Abschnitt *Anpassen von Bedienelementen durch den Import von Grafiken* in Kapitel 8, *Einführung in Frontpanel-Objekte*). Dann können Sie diese Graphik importieren und sie in die Nähe der Y-Achse des Graphen bzw. des Diagramms plazieren.

Wie plaziere ich ein Text-Label auf einen Graph-Cursor?

Zum programmatischen Zugriff verwenden Sie zunächst das Attribut “Cursor Name Visible”, um den Cursornamen anzuzeigen. Anschließend können Sie die Eigenschaft “Cursor Name” verwenden, um dem Cursor einen Namen zuzuweisen. Für den interaktiven Gebrauch ermöglicht das Menü “Cursor-Stil” die Steuerung des Namens in der Cursor-Anzeige des Graphen.

Wenn ich meinem Graphen einen Cursor hinzugefügt habe, wie kann ich diesen wieder entfernen?

Sie müssen das Array löschen, das die Cluster mit den Information über jeden Cursor enthält. Wählen Sie **Datenoperationen»Leeres Array** aus dem Popup-Menü der Cursor-Anzeige oder schreiben Sie programmatisch ein leeres Array in den Attributknoten Cursor List.

Fehlermeldungen und Systemabstürze

Was muß ich tun, wenn ein Dialogfeld mit der Meldung erscheint, daß nicht genügend Arbeitsspeicher für meine Anwendung vorhanden ist?

LabVIEW und BridgeVIEW weisen Speicher nach Bedarf zu, aber Arrays und Strings müssen in einem zusammenhängenden Speicherblock gespeichert werden. Wenn Ihre LabVIEW- oder BridgeVIEW-Anwendung nicht in der Lage ist, einen unbenutzten (physischen oder virtuellen) Speicherblock zu finden, der für den String oder das Array groß genug ist, erscheint ein Dialogfeld mit der Meldung, daß der erforderliche Speicher nicht zugewiesen werden konnte.

Wenn Sie Änderungen an Ihren VIs speichern möchten und die Anwendung nicht mehr über genügend Arbeitsspeicher verfügt, können Sie Ihre VIs an einer anderen Stelle speichern oder sicherstellen, daß Sie eine Sicherungskopie haben. Wenn Sie dann das Programm mit mehr Arbeitsspeicher neu starten, können Sie diese VIs in den Speicher laden und prüfen, ob das vorherige Speichern der VIs erfolgreich war; dann können Sie mit der VI-Entwicklung fortfahren.

Was muß ich tun, wenn LabVIEW oder BridgeVIEW während des Druckvorganges abstürzt? (Windows)

Dieser Absturz ist eventuell auf den Videotreiber oder Druckertreiber zurückzuführen. Informationen über die Vorgehensweise in einem solchen Fall finden Sie unter der nachstehenden Frage bezüglich zufälliger Systemabstürze auf Windows.

Was muß ich tun, wenn ich während der Ausführung von LabVIEW oder BridgeVIEW eine Meldung erhalte, in der ein Quellcode und eine Zeilennummer aufgeführt werden?

Hierdurch wird angegeben, daß ein Fehler aufgetreten ist und daß die Anwendung nicht fortfahren kann. Bitte verständigen Sie National Instruments über diese Meldung und erläutern Sie, welcher Vorgang diese Meldung ausgelöst hat.

Was muß ich tun, wenn bei LabVIEW oder BridgeVIEW häufig zufällige Systemabstürze auftreten? (Windows)

Bei diesen Systemabstürzen kann es sich um allgemeine Schutzfehler oder Fehlermeldungen handeln, die eine Quellcodedatei und eine Zeilennummer enthalten. Häufig sind zufällige Abstürze ein Anzeichen dafür, daß es Probleme mit dem Videotreiber des Computers gibt. Um festzustellen, ob dies der Fall ist, verwenden Sie die Standard-VGA-Einstellung als Videotreiber. Zum Ändern des Videotreibers wählen Sie "VGA" aus der Liste der Videotreiber in Windows-Setup. LabVIEW und BridgeVIEW verwenden die standardmäßige Windows-API für graphische Aufrufe; viele Videotreiber entsprechen jedoch nicht hundertprozentig diesem Standard. Wenn die Abstürze nicht mit dem standardmäßigen VGA-Treiber auftreten, dann ist es ratsam, den Videotreiber zu aktualisieren. Sie können wahrscheinlich die neueste Version der Videotreibersoftware vom Hersteller Ihres PC oder vom Hersteller der Videokarte beziehen.

Falls die Situation durch die Aktualisierung des Videotreibers nicht bereinigt werden kann, wenden Sie sich bitte an National Instruments.

Was muß ich tun, wenn ein Speicherparitätsfehler, gefolgt von einem Allgemeinen Schutzfehler auftritt? (Windows)

Speicherparitätsfehler sind das Ergebnis eines fehlerhaften Speichers in Ihrem Computer. Dabei kann es sich um virtuellen Speicher handeln, was auf ein Problem auf Ihrer Festplatte hinweist, oder um physischen Speicher, was auf ein fehlerhaftes SIMM hinweist. Zur Überprüfung Ihrer Festplatte

verwenden Sie ein standardmäßiges Datenträgerprogrammpaket wie z.B. Norton Utilities. Zur Prüfung Ihres physischen Arbeitsspeichers tauschen Sie die SIMMs in Ihrem Computer der Reihe nach gegeneinander aus und starten den PC nach jedem Austausch neu. Wenn das fehlerhafte SIMM in die unterste Speicherbank plaziert wird, kann der Computer nicht gestartet werden. Dann empfiehlt es sich, das SIMM zu ersetzen.

Plattformthemen und Kompatibilitäten

Was ist zur Übertragung von VIs zwischen Plattformen erforderlich?

Wenn eine VI-Datei in eine andere Plattform übertragen wird, kann das LabVIEW- oder BridgeVIEW-Programm diese lesen, ohne daß sie konvertiert werden muß. Wenn ein VI geöffnet wird, wird es für die neue Plattform neu kompiliert. Dies bedeutet, daß das Blockdiagramm eines VIs enthalten sein *muß*, wenn es auf eine andere Plattform übertragen werden soll. Bei VIs mit CINs müssen Sie die CINs auf der neuen Plattform neu kompilieren und dann mit dem portierten VI neu verknüpfen.

Plattformspezifische Funktionen (z.B. Apple Events für Macintosh, DDE für Windows und so weiter) müssen ausgeschaltet werden, bevor ein VI auf eine andere Plattform übertragen wird.

VIs können mit Hilfe von Netzwerken, Modems und Speichermedien zwischen Plattformen übertragen werden. VIs werden auf allen Plattformen im selben Speicherformat gespeichert. Wenn VIs mit Hilfe von FTP oder eines Modems in einem Netzwerk übertragen werden, muß eine binäre Übertragung spezifiziert werden.

Wenn Speichermedien für die Übertragung eingesetzt werden, sind zum Lesen der Speichermedien auf anderen Plattformen Dienstprogramme zum Konvertieren erforderlich. Mit Konvertierungs-Dienstprogrammen kann das Format von Dateien, die auf einem Speichermedium gespeichert sind, geändert werden, da jede Plattform (Macintosh, Windows, Sun) Dateien in einem anderen Format auf einem Speichermedium speichern. Die meisten Dateikonvertierungs-Dienstprogramme *lesen* nicht nur Dateien von anderen Plattformen, sondern *schreiben* auch Dateien im Format des Speichermediums dieser Plattform. Es gibt z.B. Dienstprogramme wie MacDisk und TransferPro für PCs, mit denen Macintosh-Speichermedien in das PC-Format und umgekehrt übertragen werden. Auf dem Macintosh sind z.B. DOS Mounter und Apple File Exchange zwei Dienstprogramme, mit denen Dateien auf DOS-formatierten Speichermedien in das Macintosh-Format und umgekehrt übertragen werden. Für die Sun- und

HP-Systeme gibt es PC File System (PCFS), mit dem SunOS und HP-UX formatierte DOS-Speichermedien lesen und schreiben können.

Zur Erleichterung des Portierens zwischen Plattformen können Sie Ihre VIs in einer VI-Bibliothek speichern. Zum leichteren Speichern aller VIs in einer Bibliothek können Sie die Option **Development Distribution** aus dem Dialogfeld “Mit Optionen speichern” auswählen. Auf diese Weise lassen sich alle VIs, Bedienelemente und externen Subroutinen, die nicht in der `vi.lib` enthalten sind, in einer einzigen Bibliothek speichern.

Drucken

Wie drucke ich ein einzelnes Bedienelement vom Frontpanel (zum Beispiel einen Graphen)?

Zum Drucken eines einzelnen Graphen vom Frontpanel erstellen Sie ein SubVI mit einem Graphen auf dem dazugehörigen Frontpanel.

1. Ändern Sie den Graphen von einem Anzeigeelement in ein Bedienelement.
2. Öffnen Sie das SubVI und wählen Sie **Ausführen»Drucken nach Abschluß**.
3. Weisen Sie dem SubVI einen Anschluß zu und übergeben Sie Daten vom Graphen auf dem Haupt-VI zum Graphen auf dem SubVI.

Jedesmal, wenn Ihr Haupt-VI das SubVI aufruft, druckt es automatisch den Graphen aus.

Wie kann ich einen String drucken?

Verwenden Sie das VI `Seriellen Anschluß initialisieren`, um den Anschluß, an dem der Drucker angeschlossen ist, zu initialisieren (LPT1, LPT2 usw. auf dem PC oder dem Druckeranschluß am Mac), und dann das VI `Auf seriellen Anschluß schreiben`, um den String zum initialisierten Anschluß zu schreiben. Der Drucker sieht die Daten am Anschluß und druckt sie. Hierzu sind zwar im allgemeinen Kenntnisse der Befehlssprache Ihres Druckers erforderlich; diese Vorgehensweise funktioniert jedoch sehr gut bei einer Reihe von Anwendungen, die von LabVIEW- und BridgeVIEW-Benutzern entwickelt wurden. Wenn Sie LabVIEW-Benutzer sind, finden Sie weitere Einzelheiten im Anhang B, *Häufig gestellte Fragen des LabVIEW Benutzerhandbuchs*.

(Windows und UNIX) Mit dem VI “Systembefehl ausführen” können Sie eine Datei über eine Befehlszeilenfunktion ausdrucken. Das VI befindet sich in **Funktionen»Kommunikation»AppleEvent**. Wenn Sie z.B. unter Windows NT/95 ein Textdokument ausdrucken möchten, müssen Sie das VI “Systembefehl ausführen” mit dem folgenden Befehl verwenden:

```
notepad /p <Dokumentname>
```

Der Windows-Editor Notepad ist auf 32K begrenzt. Für größere Dokumente müssen Sie einen anderen Texteditor verwenden.

(Macintosh) Sie können das VI “Sendet AE Dokument drucken” dazu verwenden, eine andere Anwendung zum Drucken eines Dokuments anzuweisen. Das VI befindet sich in **Funktionen»Kommunikation»AppleEvent**. Weitere Einzelheiten finden Sie im Abschnitt *Verwendung von alternativen Druckmethoden*, in Kapitel 5, *Drucken und Dokumentieren von VIs*. Außerdem können Sie die programmatische Druckoption verwenden, wie bei der vorangegangenen Frage beschrieben wurde.

Wie drucke ich alle Daten auf einem Kurvenformdiagramm?

Das Kurvenformdiagramm druckt nur die angezeigten Daten, wenn das Frontpanel ausgedruckt wird. Zum Drucken aller Daten in einem Diagramm, einschließlich der Daten im Historie-Puffer, muß zunächst die X-Achse des Diagramms automatisch skaliert werden, indem das entsprechende Popup-Menü am Diagramm aufgerufen wird, die Diagrammpalette angezeigt und die X-Achse fixiert oder die X-Achse programmatisch über einen Diagramm-Attributknoten eingestellt wird.

Warum stürzt LabVIEW oder BridgeVIEW beim Drucken ab?

Bei Windows kann dieser Absturz mit dem Videotreiber oder dem Druckertreiber zusammenhängen.

Informationen zur Vorgehensweise finden Sie unter der Frage bezüglich zufälliger Systemabstürze auf Ihrer Plattform im Abschnitt *Fehlermeldungen und Systemabstürze* in diesem Anhang.

Wie verwende ich PostScript-Drucken?

Wenn Ihr Drucker PostScript-fähig ist, müssen Sie zuerst den PostScript-Druckertreiber für Ihren Drucker installieren und wählen. In LabVIEW oder BridgeVIEW rufen Sie **Bearbeiten»Voreinstellungen»Drucken** auf und wählen Sie **PostScript drucken**.

Wenn Sie einen PostScript-Drucker haben, verfügen Sie über folgende Vorteile:

- PostScript-Ausdrucke geben den Bildschirm genauer wieder.
- PostScript erzeugt Muster und Linienstile genauer.

Wenn ich “PostScript drucken” auswähle, warum erscheint der Text zusammengedrängt oben auf der Seite?

Die Funktion “Drucken” übersetzt die VI-Druckdaten in das PostScript-Format (.ps) und schickt sie als PostScript-Text an den Druckertreiber von Windows. Wenn Ihr Druckertreiber die Funktion “PostScript drucken” nicht unterstützt, enthält der Ausdruck den tatsächlichen Text der PostScript-Datei anstelle eines graphischen Bildes. Weitere Informationen finden Sie unter der vorherigen Frage zu PostScript.

Was muß ich tun, wenn der Text auf Labels und Frontpanel-Bedienelementen beim Ausdrucken von VIs abgeschnitten ist?

Dies passiert dann, wenn die Größe der Schriftart für den Drucker nicht mit der Größe der Schriftart auf dem Monitor übereinstimmt.

- Wenn Ihr Drucker die PostScript-Sprache unterstützt, verwenden Sie einen PostScript-Druckertreiber, um auf Ihrem Drucker Ausdrücke mit besserer Qualität zu erzeugen. Weitere Informationen über das PostScript-Drucken finden Sie unter der vorherigen Frage.
- Vergrößern Sie die Labels und Frontpanel-Bedienelemente, damit der Text bei einer Vergrößerung aufgrund nicht übereinstimmender Schriftarten dennoch in das Bedienelement oder Label paßt.
- Finden Sie eine Schriftart, die sowohl auf dem Monitor als auch auf dem Drucker dieselbe Größe hat. Zahlreiche Druckertreiber verfügen über Schriftarten-Ersetzungsalgorithmen, um dieses Verfahren zu erleichtern.
- **(Windows)** Verwenden Sie Bitmap Drucken, so daß der Ausdruck genau dem Bild entspricht, das auf dem Monitor erscheint.

Wie wähle ich Bitmap Drucken? (Windows)

Wählen Sie **Bearbeiten»Voreinstellungen** und anschließend **Drucken**. Die letzte Option auf dem Bildschirm ist **Bitmap Drucken**.

Verschiedenes

Was ist Info-LabVIEW und wie werde ich Abonnent?

Info-LabVIEW ist ein von Benutzern gefördertes und unterstütztes Netzwerk aus LabVIEW-Benutzern. Benutzer können Informationen an eine E-Mail-Adresse schicken (siehe unten); die Nachricht wird dann an alle Benutzer weitergeleitet, die Abonnenten dieser Liste sind. Andere Benutzer können dann eine Antwort an die Gruppe oder eventuell direkt an die Person senden, die die ursprüngliche Nachricht geschickt hat.

Verschiedene Hauptbenutzer sowie einige Mitarbeiter von National Instruments sind Abonnenten dieser Liste und diskutieren verschiedene Themen. Insgesamt gesehen hat die Info-LabVIEW-Gruppe sehr guten Anklang gefunden. Sie ist zwar offiziell nicht mit National Instruments verbunden, jedoch erfolgen von Zeit zu Zeit offizielle Mitteilungen vom Unternehmen an die Gruppe. Zumeist finden die Gespräche nur zwischen Benutzern statt.

Wenn Sie die Liste abonnieren möchten, schicken Sie eine E-Mail an folgende Adresse:

`info-labview-request@pica.army.mil`

Benutzer können die Meldungen auf zweierlei Weise erhalten — im Standardformat und im gesammelten Format. Das Standardformat bedeutet, daß Sie die Meldungen dann erhalten, wenn sie geschickt werden. Das gesammelte Format bedeutet, daß Sie ein Paket am Ende jeden Tages erhalten, das alle Meldungen des Tages umfaßt. Wenn Sie eine Sammelmeldung pro Tag erhalten möchten, so geben Sie “Digest Format” in den E-Mail-Nachrichten an, die an `info-labview-request` gesendet werden.

Nachdem Sie die Liste abonniert haben, erhalten Sie Nachrichten von anderen Abonnenten. Der Datenverkehr in der Gruppe ist sehr rege und umfaßt ca. 10 bis 30 Nachrichten pro Tag. Zum Übertragen von Informationen in die Liste schicken Sie die E-Mail an folgende Adresse.

`info-labview@pica.army.mil`

Wie kann ich VIs dynamisch laden und ausführen?

Wenn ein SubVI im Blockdiagramm eines anderen VIs enthalten ist, wird das SubVI in den Speicher geladen, sobald das aufrufende VI in den Speicher geladen wird. Aus Gründen der Speichereffizienz ist es ratsam, VIs dynamisch vom Speicher zu laden und zu entladen, während Ihr Programm ausgeführt wird. Sie können dazu die VI-Serverfunktion verwenden. Weitere Informationen zu diesem Thema finden Sie in Kapitel 21, *VI-Server*, in diesem Handbuch.

(Macintosh) Zusätzlich zu dem plattformunabhängigen VI-Server können Sie Apple Events auf dem Macintosh verwenden, um VIs dynamisch zu laden und auszuführen. Die VIs befinden sich in der Palette **Funktionen»Kommunikation»AppleEvent** oder der dazugehörigen Unterpalette **LabVIEW Specific Apple Events** und heißen:

- Sendet AE Finder-öffnen
- Sendet AE VI öffnen, ausführen, schließen
- Sendet AE VI ausführen
- Sendet AE VI schließen

Wie ersetze ich ein SubVI durch ein anderes VI mit demselben Namen?

LabVIEW und BridgeVIEW nehmen auf alle VIs durch ihre Namen Bezug; daher werden auf keinen Fall zwei VIs mit demselben Namen geladen, unabhängig vom Pfad, in welchem sich das VI befindet. Dasselbe gilt auch für SubVIs. Wenn Sie auf einem SubVI **Ersetzen** wählen, um es durch ein VI mit demselben Namen zu ersetzen, stellt G fest, daß eine Kopie des VIs bereits im Speicher ist, und ersetzt das VI durch sich selbst.

Um die neue Kopie des SubVIs laden zu können, müssen Sie zuerst die alte Kopie aus dem Speicher löschen. Die VIs sind im **Windows**-Menü aufgeführt. Am leichtesten läßt sich das SubVI aus dem Speicher löschen, wenn Sie das SubVI und alle VIs schließen, die das SubVI aufrufen. Öffnen Sie die neue Kopie des SubVIs im Speicher, rufen Sie **Datei»VI-Info anzeigen...** auf, um zu überprüfen, ob es sich um die gewünschte Version handelt, und öffnen Sie dann das Haupt-VI erneut. Wenn das VI versucht, die Verknüpfung mit dem SubVI herzustellen, findet es eine Kopie im Speicher (die neue Version), verwendet diese Version und sucht daher nicht nach der vorherigen Version.

Die Lösung für derartige Probleme besteht darin, SubVIs stets eindeutige Namen zuzuweisen.

Das Laden von zwei VIs, die zwei verschiedene SubVIs mit demselben Namen aufrufen, führt zu Problemen.

Zum Beispiel rufen `main1.vi` und `main2.vi` unterschiedliche SubVIs auf, die beide als `subVI.vi` bezeichnet werden.

1. `main1.vi` wird in den Speicher geladen, `subVI.vi` wird ebenfalls in den Speicher geladen, da es von `main1.vi` aufgerufen wird.
2. `main2.vi` wird in den Speicher geladen, die Linker-Informationen fordern LabVIEW oder BridgeVIEW dazu auf, `subVI.vi` zu laden. Die Anwendung stellt fest, daß sie `subVI.vi` bereits in den Speicher geladen hat, und versucht, `main2.vi` mit diesem `subVI.vi` zu verknüpfen. Wenn das Anschlußfeld genau gleich ist, wird die Verknüpfung wahrscheinlich hergestellt (aber das Verhalten kann sehr ungewöhnlich ausfallen). Wenn das Anschlußfeld anders ist, erhalten Sie einen Linkfehler.

Lösung: SubVIs sollten stets eindeutige Namen erhalten.

Wie wirkt sich Priorität auf die Ausführung von VIs aus?

Das G-Ausführungssystem kann mehrere VIs oder Subdiagramme parallel ausführen. VIs können eine von fünf Prioritäten erhalten:

Hintergrundpriorität (niedrigste), normale Priorität, über normaler Priorität, hohe Priorität und zeitkritische Priorität (höchste). Eine weitere Priorität, die Subroutinen-Priorität, zwingt ein VI zur Ausführung bis zum Abschluß, ohne daß sein Ausführungsthread-System mit anderen VIs gemeinsam genutzt wird. Jedes Ausführungssystem hat eine Warteschlange mit Code-Abschnitten der VIs im System. Die Warteschlange ist nach Priorität angeordnet, wobei VIs mit höherer Priorität vor VIs mit niedrigerer Priorität ausgeführt werden. Solange sich Objekte mit hoher Priorität in der Warteschlange befinden, können Objekte mit niedrigerer Priorität nicht zum Anfang der Warteschlange gelangen und werden daher nicht ausgeführt. Dies bedeutet, daß VIs mit niedriger Priorität ständig zurückgestellt werden können, wenn VIs mit höherer Priorität kontinuierlich ausgeführt werden, ohne daß für ein Timeout oder eine andere asynchrone Aktivität eine Pause eingelegt wird. Aus diesem Grund sollten Sie vorsichtig sein, wenn Sie die Priorität eines VIs hochstufen.

Die Ausführungsschlange in G hat für jede der fünf Prioritäten zwei Eingangspunkte. Für die Subroutinen-Priorität gibt es keinen Eingangspunkt. Ein VI mit Subroutinen-Priorität beginnt mit der Ausführung, wenn es von einem VI aufgerufen wird, und dominiert das Ausführungsthread-System, bis es zum Aufrufer-VI mit einer anderen

Priorität zurückkehrt. Es kann andere VIs mit Subroutinen-Priorität aufrufen, jedoch nicht VIs mit anderen Prioritäten. VIs mit Subroutinen-Priorität können die dazugehörigen Anzeigeelemente nicht aktualisieren, sie können keine Warte-Operationen oder sonstigen asynchronen Operationen durchführen und können daher nicht in eine Warteschlange plaziert werden. Wenn ein solches VI mit der Ausführung beginnt, ist es in diesem Sinne das VI mit der höchsten Priorität in einem Ausführungssystem.

In Multithread-Systemen gibt es mehrere Ausführungssysteme, wobei jedes System über eine Ausführungsschlange und einen oder mehrere Threads verfügt. Die Threads werden so eingestellt, daß das Betriebssystem einen Thread mit niedriger Priorität unterbrechen kann, wenn ein Thread mit höherer Priorität ausführungsbereit ist. Die VI-Priorität wird dazu verwendet, die Warteschlange des Ausführungssystems auszuwählen, in die das VI am besten plaziert werden soll. Ein VI mit Subroutinen-Priorität, das in einem Multithread-System aufgerufen wird, dominiert den Thread, in welchem es aufgerufen wird, kann jedoch andere VIs in anderen Threads nicht an ihrer Ausführung hindern. Dies bedeutet, daß Sie VIs mit Subroutinen-Priorität auf Systemen mit anderen Threads mit gleicher oder höherer Priorität nicht zum Blockieren der Ausführung aller anderen VIs verwenden können.

Auf Multithread-Systemen ist die Benutzerschnittstelle ein Thread mit normaler Priorität. Bei manchen Systemen kann sie durch VIs zurückgestellt werden, die mit höherer Priorität ausgeführt werden und keine Pause für ein Timeout oder eine andere asynchrone Aktivität einlegen. Es ist daher unter Umständen nicht möglich, ein Bedienelement so zu ändern, daß ein VI mit hoher Priorität gestoppt wird, da die Benutzerschnittstelle keine Gelegenheit erhält, das Bedienelement zu beachten. Deswegen müssen Sie vorsichtig sein, wenn Sie die Priorität eines VIs hochstufen.

Wie können Interrupts von G ohne die Verwendung von Polling behandelt werden?

Occurrences sind eine Methode, mit der G einen Code-Abschnitt dazu veranlaßt, auf ein bestimmtes Ereignis zu warten. Sie können z.B. einen CIN schreiben, der einen Prozeß startet (in Windows z.B. über ein DLL), der ein Occurrence erzeugt, wenn im DLL ein Ereignis auftritt. Ein bestimmter Abschnitt des Diagrammcodes wird ausgeführt, und ein anderer Abschnitt oder ein VI wartet auf das Ereignis. Wenn das Ereignis eintritt, protokolliert es das Occurrence in LabVIEW oder BridgeVIEW,

und der entsprechende Code beginnt mit der Ausführung, sobald er in der Warteschlange an der Reihe ist.

Intern betrachten LabVIEW und BridgeVIEW die Warteschlange, nachdem sie einen Block mit atomarem Code abschließt. Dies bedeutet, daß vor dem Occurrence alles, was ausgeführt wird, noch abgeschlossen werden muß; dann wird der Interrupt behandelt.



Hinweis

Für diesen Prozeß gibt es keine bestimmte Zeitgrenze, ganz besonders dann nicht, wenn das VI CINs enthält, die sehr lange brauchen können, bis sie abgeschlossen sind.

Wie halte ich bei eingebetteten While-Schleifen beide Schleifen an, ohne daß irgendetwas in der äußeren Schleife ausgeführt wird, nachdem die innere Schleife angehalten hat?

Setzen Sie den Code, der bei der letzten Iteration nicht ausgeführt werden soll, in die Case-Struktur. Es ist am besten, wenn das Bedingungs-Terminal der inneren Schleife mit dem Auswahl-Terminal der Case-Struktur verbunden ist.

Was ist die Panelanordnung?

Bearbeiten»Panelanordnung bestimmt die Anordnung der Objekte auf dem Frontpanel. Mit anderen Worten, wenn Sie die <Tab>-Taste auf der Tastatur verwenden, um sich zwischen Objekten hin- und herzubewegen, ändert sich der Fokus von Objekt zu Objekt in der Reihenfolge, die durch die Panelanordnung bestimmt wird. Nur Bedienelemente können Fokus erhalten; die <Tab>-Taste kann den Fokus nicht auf Anzeigeelemente umschalten. Standardmäßig ist die Panelanordnung die Reihenfolge, in der Objekte am Frontpanel erstellt werden.

Mit der Frontpanel-Datenprotokollierung kann die Panelanordnung die Reihenfolge festlegen, in welcher die verschiedenen Objekte innerhalb der Datei in einen Cluster komprimiert werden.

Wie kann ich LabVIEW oder BridgeVIEW anweisen, ein bestimmtes VI automatisch zu starten?

Standardmäßig starten und erstellen LabVIEW und BridgeVIEW ein VI namens Untitled1.vi. Sie möchten eventuell Ihre Anwendung so einrichten, daß ein bestimmtes VI automatisch gestartet wird. Außerdem können Sie das VI so einstellen, daß es nach dem Laden ausgeführt wird, so daß durch den Start von LabVIEW oder BridgeVIEW nicht nur Ihr VI

geöffnet, sondern auch mit dessen Ausführung begonnen wird. Stellen Sie die entsprechenden **Ausführungsoptionen** in **VI-Einstellungen...** so ein, daß ein VI aufgeführt wird, wenn es geöffnet wird. Wenn Sie eine Bibliothek (.llb) angeben, die beim Start Ihrer Anwendung geöffnet werden soll, öffnet diese alle VIs, die mit **Höchste Ebene** markiert sind. Verwenden Sie **Datei»VI-Bibliothek bearbeiten...**, um die VIs mit **Höchste Ebene** zu markieren. Und wenn Sie schließlich eine Bibliothek angeben und beim Start keine VIs mit **Höchste Ebene** markiert sind, dann fordert ein Datei-Dialogfeld den Benutzer dazu auf, ein VI innerhalb der angegebenen .llb auszuwählen.

Befolgen Sie die nachstehenden Anweisungen für die relevante Plattform, um LabVIEW oder BridgeVIEW anzuweisen, ein bestimmtes VI zu starten.

(Windows) Wählen Sie das LabVIEW- oder BridgeVIEW-Icon und wählen Sie **Datei»Eigenschaften** (im Programm-Manager), und ändern Sie den Pfadnamen in **Befehlszeile**, um auf Ihr VI zu zeigen. Damit LabVIEW z.B. `test.vi` automatisch lädt, stellen Sie die **Befehlszeile** folgendermaßen ein:

```
c:\labview\labview.exe test.vi
```

Wenn `test.vi` sich in einer Bibliothek befindet, die `test.llb` heißt, dann lautet die Zeile folgendermaßen:

```
c:\labview\labview.exe "test.llb\test.vi"
```

Es ist eventuell notwendig, den vollständigen Pfad zum VI anzugeben. Wenn Sie ein VI von der Befehlszeile (**Start»Ausführen...**) starten möchten, lautet die entsprechende Syntax folgendermaßen:

```
c:\labview\labview.exe <Pfad zu VI relativ zum  
Labview-Verzeichnis>
```

Um z.B. das `Readme.vi` im Verzeichnis `c:\labview\examples directory` zu starten, lautet der Befehl:

```
c:\labview\labview.exe examples\readme.vi
```

Wenn das VI sich in einem anderen Pfad befindet, müssen Sie den vollständigen Pfad zum VI angeben:

```
c:\labview\labview.exe c:\coolapp\mycool.vi
```

Wenn ein Verzeichnis im Pfad Leerzeichen enthält, müssen Sie den Pfad in Anführungszeichen setzen:

```
c:\labview\labview.exe "c:\cool application\mycool.vi"
```

Wenn sich das VI in einer LLB befindet, können Sie folgendes tun:

1. Geben Sie den vollständigen Pfad in Anführungszeichen an:

```
c:\labview\labview.exe "c:\coolapp\eagle.llb\mycool.vi"
```

oder

2. Ändern Sie die VI-Einstellungen für das VI auf **Nach Öffnen ausführen**. Wählen Sie anschließend **Datei>VI-Bibliothek bearbeiten...** und markieren Sie das VI mit **Höchste Ebene**.

In diesem Fall müssen Sie nur den Pfad zur llb angeben:

```
c:\labview\labview.exe "c:\coolapp\eagle.llb"
```

Wenn Sie möchten, daß LabVIEW automatisch ein VI startet, ändern sie die Verknüpfungseigenschaften und geben Sie den Pfad ins Ziel wie oben aufgeführt an.

(Macintosh) Es gibt keine Methode, die LabVIEW-Anwendung anzuweisen, ein bestimmtes VI auf dem Macintosh automatisch zu starten; Sie können jedoch ein VI starten, indem Sie im "Finder" auf sein Icon doppelklicken. Wenn auf Ihrem Computer mehrere Kopien von LabVIEW installiert sind, bestimmt der Finder, welche Kopie gestartet wird, wenn Sie auf das VI doppelklicken (Sie können nicht steuern, welche Kopie gestartet wird). Der Finder startet z.B. das Run-Time-System, wenn sowohl das Run-Time-System als auch das Full Development System installiert sind. Wenn Sie System 7 oder später ausführen, können Sie Drag & Drop verwenden, um ein oder mehrere VIs oder VI-Bibliotheken zu starten.

(UNIX) LabVIEW reagiert auf Befehlszeilenoptionen und kann beim Öffnen ein bestimmtes VI starten. Sie können daher folgendes eingeben:

```
labview /usr/home/test.vi
```

oder

```
labview /usr/home/test.llb/test.vi
```

um das Test-VI zu starten, je nach korrektem Pfad. Außerdem können Sie ein einfaches Skript verwenden, um einen Befehl zu erstellen, der LabVIEW mit einem bestimmten VI startet.

Wie kann ich die Einträge eines Enumerated Types programmatisch ändern?

Sie können den Typ (die Strings) eines Enumerated Type nicht programmatisch ändern; ebensowenig können Sie ein Integer-Bedienelement in ein Double- oder ein String-Bedienelement in ein Pfad-Bedienelement ändern. Die Strings in einem Enum sind ein Teil seines Datentyps und können somit nur während der Bearbeitungszeit geändert werden. Es ist möglich, die Strings des Enum über einen Attributknoten zu lesen, aber Sie können sie nicht mit einem Attributknoten schreiben.

Wenn Sie die Textwerte im Enum programmatisch ändern möchten, verwenden Sie statt dessen ein Textring-Bedienelement. Sie können das Ring-Bedienelement verwenden, um die Strings programmatisch über die Eigenschaft "Strings[]" zu lesen und schreiben.

Wie kann ich die Menüleisten eines VIs verbergen?

Alle VI-Eigenschaften, einschließlich der Option, ob die Menüleiste und Symbolleiste angezeigt werden sollen, werden über das Dialogfeld in **VI-Einstellungen...** eingestellt. Zum Aufrufen des Dialogfeldes "VI -Einstellungen" rufen Sie das Popup-Menü des VI-Icons in der oberen rechten Ecke des Frontpanels auf und wählen Sie **VI-Einstellungen....** Weitere Einzelheiten finden Sie in Kapitel 6, *Einrichten von VIs und SubVIs*.

Wie deaktiviere ich Maus-Interrupts, um die Leistung zu verbessern?

Interrupts, die durch Bewegungen der Maus oder das Klicken auf ein Objekt verursacht werden, beanspruchen CPU-Zeit. In einigen extrem zeitkritischen Anwendungen kann diese Interrupt-Aktivität zu einem Datenverlust oder einem anderen Problem in der Anwendung führen. Bis auf das Ausstecken der Maus gibt es keine empfohlene Methode zum Deaktivieren von Maus-Interrupts. Wenn Sie sicherstellen, daß sich die Maus während der zeitkritischen Phasen Ihrer Anwendung nicht bewegt, können Sie diese Auswirkungen auf ein Mindestmaß reduzieren.



Kundenbetreuung

Um Ihnen im Falle technischer Probleme die Zusammenarbeit mit uns zu erleichtern, sind in diesem Anhang Formulare enthalten, die Ihnen dabei helfen, die technischen Informationen zusammenzustellen, die wir zur Lösung Ihrer technischen Probleme benötigen. Außerdem ist ein Formular enthalten, auf dem Sie Kommentare zur Produktdokumentation abgeben können. Wenn Sie sich mit uns in Verbindung setzen, benötigen wir die Informationen auf dem Formular für Technische Unterstützung sowie auf dem Konfigurationsformular, soweit dies in Ihrem Handbuch enthalten ist, mit dessen Hilfe wir Ihre Fragen zur Systemkonfiguration so schnell wie möglich beantworten.

National Instruments bietet technische Unterstützung auf elektronischem Wege und per Fax und Telefon und läßt Ihnen somit die von Ihnen benötigten Informationen schnellstmöglich zukommen. Zu unserem elektronischen Service gehört ein Bulletin-Board, eine FTP-Site, ein Faxabrufsystem sowie E-Mail-Unterstützung. Versuchen Sie bei Hardware- oder Softwareproblemen erst unser elektronisches Unterstützungssystem. Sollten Ihre Fragen durch die hier zur Verfügung stehenden Informationen nicht beantwortet werden, bieten wir Ihnen durch unsere mit Anwendungsingenieuren besetzten technischen Support-Zentren per Fax und Telefon Hilfe.

Elektronischer Service

Bulletin Board Support

National Instruments hat BBS- und FTP-Sites, die Ihnen rund um die Uhr zur Verfügung stehen und auf denen sich Dateien und Dokumente mit Antworten zu den am häufigsten vorkommenden Kundenfragen befinden. Von diesen Sites können Sie auch die neuesten Instrumententreiber sowie Updates und Beispielprogramme herunterladen. Aufgezeichnete Anweisungen zur Benutzung des BBS- und FTP-Service sowie automatisierte BBS-Informationen erhalten Sie unter der Tel.Nr. USA 1-512 795 6990. Sie können auf diesen Service folgendermaßen zugreifen:

Vereinigte Staaten: 1-512 794 5422

Bis zu 14.400 Baud, 8 Datenbits, 1 Stoppbit, keine Parität

Großbritannien: 01635 551422

Bis zu 9.600 Baud, 8 Datenbits, 1 Stoppbit, keine Parität

Frankreich: 01 48 65 15 59

Bis zu 9.600 Baud, 8 Daten bits, 1 Stoppbit, keine Parität

FTP-Unterstützung

Melden Sie sich zum Zugriff auf unsere FTP-Site bei unserem Internet-Host `ftp.natinst.com`, `anonymous` an und benutzen Sie Ihre E-Mail-Adresse, wie z.B. `joesmith@anywhere.com` als Paßwort. Die Unterstützungsdateien und -dokumente befinden sich in den Verzeichnissen `/support`.

Faxabruf-Unterstützung

Bei der Faxabfrage handelt es sich um ein Informationssystem, das rund um die Uhr zugänglich ist und eine Bibliothek an Dokumenten zu einem umfassenden Bereich technischer Informationen enthält. Sie können auf die Faxabfrage von einem Telefon mit Tonwahl unter der folgenden Nummer zugreifen: 1-512 418 1111.

Unterstützung per E-Mail

Sie können dem Anwendungstechnikerteam Fragen zur technischen Unterstützung per E-Mail an unsere unten aufgeführte Internet-Adresse zukommen lassen. Bitte vergessen Sie nicht, Ihren Namen, Anschrift und Telefonnummer anzugeben, damit wir Ihnen unsere Lösungen und Vorschläge mitteilen können.

ni.germany@natinst.com

Unterstützung per Telefon und Fax

National Instruments unterhält auf der ganzen Welt Niederlassungen. In nachstehender Liste finden Sie die Nummer für technische Unterstützung für Ihr Land. Sollte in Ihrem Land kein Büro von National Instruments bestehen, setzen Sie sich bitte mit der Stelle in Verbindung, von der Sie die Software gekauft haben, für die Sie Unterstützung benötigen.

Land	Telefon	Fax
Australien	03 9879 5166	03 9879 6277
Belgien	02 757 00 20	02 757 03 11
Brasilien	011 288 3336	011 288 8528
Dänemark	45 76 26 00	45 76 26 02
Deutschland	089 741 31 30	089 714 60 35
Finnland	09 725 725 11	09 725 725 55
Frankreich	01 48 14 24 24	01 48 14 24 14
Großbritannien	01635 523545	01635 523154
Hongkong	2645 3186	2686 8505
Israel	03 6120092	03 6120095
Italien	02 413091	02 41309215
Japan	03 5472 2970	03 5472 2977
Kanada (Ontario)	905 785 0085	905 785 0086
Kanada (Québec)	514 694 8521	514 694 4399
Korea	02 596 7456	02 596 7455
Mexiko	5 520 2635	5 520 3282
Niederlande	0348 433466	0348 430673
Norwegen	32 84 84 00	32 84 86 00
Österreich	0662 45 79 90 0	0662 45 79 90 19
Singapur	2265886	2265887
Spanien	91 640 0085	91 640 0533
Schweden	08 730 49 70	08 730 43 70
Schweiz	056 200 51 51	056 200 51 55
Taiwan	02 377 1200	02 737 4644
Vereinigte Staaten	512 795 8248	512 794 5678

Formular für technische Unterstützung

Fotokopieren Sie dieses Formular und bringen Sie es jedes Mal, wenn Sie Änderungen an Ihrer Software oder Hardware vornehmen, auf den neuesten Stand. Verwenden Sie die ausgefüllte Kopie dieses Formulars als Nachschlagequelle für Ihre derzeitige Konfiguration. Wenn Sie dieses Formular sorgfältig ausfüllen, bevor Sie sich wegen technischer Unterstützung mit National Instruments in Verbindung setzen, helfen Sie damit unseren Anwendungsingenieuren, Ihre Fragen effizienter zu beantworten.

Sollten Sie Hardware- oder Softwareprodukte von National Instruments verwenden, die mit diesem Problem zusammenhängen, legen Sie bitte Konfigurationsformulare aus den jeweiligen Handbüchern bei. Verwenden Sie, wenn nötig, zusätzliche Seiten.

Name _____

Firma _____

Adresse _____

Fax (____) _____ Tel. (____) _____

Computermarke _____ Modell _____ Prozessor _____

Betriebssystem (inkl. Versionsnr.) _____

Taktgeschw. _____ MHz RAM _____ MB Anzeigeadapter _____

Maus ___ja ___nein Andere installierte Adapter _____

Festplatten-Kapazität _____ MB Marke _____

Verwendete Instrumente _____

National Instruments Hardware-Produktmodell _____ Revision _____

Konfiguration _____

National Instruments Softwareprodukt _____ Version _____

Konfiguration _____

Beschreibung des Problems: _____

Bitte alle Fehlermeldungen auführen: _____

Durch folgende Schritte wird das Problem wiederholt: _____

LabVIEW-Hardware- und Software-Konfigurationsformular

Notieren Sie die Einstellungen und Revisionen Ihrer Hardware und Software, die sich in der Zeile rechts von jedem Artikel befindet. Füllen Sie bei jeder Änderung an Ihrer Software- oder Hardware-Konfiguration jedes Mal eine neue Kopie dieses Formulars aus, und benutzen Sie dieses Formular als Nachschlagequelle für Ihre derzeitige Konfiguration. Wenn Sie dieses Formular sorgfältig ausfüllen, bevor Sie sich wegen technischer Unterstützung mit National Instruments in Verbindung setzen, helfen Sie damit unseren Anwendungsingenieuren, Ihre Fragen effizienter zu beantworten.

National Instruments Produkte

DAQ-Hardware _____

Interrupt-Level der Hardware _____

DMA-Kanäle der Hardware _____

Basis-I/O-Adresse der Hardware _____

Programmierungswahl _____

HiQ-, NI-DAQ-, LabVIEW- oder LabWindows-Version _____

Andere Karten in Ihrem System _____

Basis-I/O-Adresse anderer Karten _____

DMA-Kanäle anderer Karten _____

Interrupt-Level anderer Karten _____

Andere Produkte

Computer-Hersteller- und Modell _____

Mikroprozessor _____

Taktfrequenz oder -geschwindigkeit _____

Typ der installierten Videokarte _____

Betriebssystemversion _____

Betriebssystemmodus _____

Programmiersprache _____

Programmiersprachenversion _____

Andere Karten in Ihrem System _____

Basis-I/O-Adresse anderer Karten _____

DMA-Kanäle anderer Karten _____

Interrupt-Level anderer Karten _____

Formular für Kommentare zur Dokumentation

National Instruments bittet Sie um Ihre Meinung zu der mit unseren Produkten mitgelieferten Dokumentation. Mit Hilfe dieser Informationen sind wir besser in der Lage, die von Ihnen benötigten Qualitätsprodukte zu liefern.

Titel: *Referenzhandbuch zur Programmierung in G*

Ausgabe: July 1998

Artikelnummer: 321989A-01

Ihre Meinung zu Vollständigkeit, Klarheit und Aufbau des Handbuchs.

Falls Sie im Handbuch Fehler entdecken, geben Sie bitte die Seite an und beschreiben Sie den Fehler.

Wir bedanken uns für Ihre Hilfe.

Name _____

Titel _____

Firma _____

Adresse _____

Tel. (____) _____ Fax (____) _____

E-Mail-Adresse _____

Senden an: Technical Publications
National Instruments Germany GmbH
Konrad-Celtis-Str. 79
81369 München

Fax an: Technical Publications
National Instruments Germany GmbH
(089) 714 60 35

Glossar

Präfix	Bedeutung	Wert
m-	Milli-	10^{-3}
μ -	Mikro-	10^{-6}
n-	Nano-	10^{-9}

Symbole

∞	Unendlich.
π	Pi.
Δ	Delta. Differenz. Δx ist der Wert, um den x sich von einem Index zum nächsten ändert.

A

Ablaufinvariante Ausführbarkeit	Modus, in welchem Aufrufe an mehrere Instanzen eines SubVIs mit verschiedenem und separatem Datenspeicher parallel ausgeführt werden können.
Absoluter Pfad	Datei- oder Verzeichnispfad, der die Speicherstelle in Relation zur höchsten Ebene des Dateisystems beschreibt.
Aktives Fenster	Fenster, das momentan so eingestellt ist, daß Benutzereingaben akzeptiert werden; normalerweise das vorderste Fenster. Die Titelleiste eines aktiven Fensters ist hervorgehoben. Ein Fenster wird zum aktiven Fenster, indem man darauf klickt oder indem man es aus dem Fenstermenü auswählt.
Aktuelles VI	VI, dessen Frontpanel, Blockdiagramm oder Icon-Editor das aktive Fenster ist.
Anschluß	Teil des VIs oder Funktionsknotens, der die dazugehörigen Eingangs- und Ausgangsterminals enthält, durch welche die Daten zum und vom Knoten übergeben werden.

Anschlußfeld	Bereich in der oberen rechten Ecke eines Frontpanelfensters, in dem das VI-Terminal-Pattern angezeigt wird. Er liegt dem Icon-Feld zugrunde.
ANSI	American National Standards Institute (US-Normungsinstitut).
Anzahl-Terminal	Das Terminal einer For-Schleife, dessen Wert bestimmt, wie oft die For-Schleife ihr Subdiagramm ausführt.
Anzeigeelement	Frontpanel-Objekt, das Ausgaben anzeigt.
Array	Angeordnete, indizierte Liste mit Datenelementen desselben Typs.
Array-Shell	Frontpanel-Objekt, in dem ein Array untergebracht ist. Es besteht aus einer Index-Anzeige, einem Datenobjektfenster und einem optionalen Label. Es können unterschiedliche Datentypen eingegeben werden.
ASCII	American Standard Code for Information Interchange (amerikanischer Standardcode für Informationsaustausch).
Asynchrone Ausführung	Modus, in welchem mehrere Prozesse Prozessorzeit gemeinsam nutzen. Ein Prozeß wird z.B. ausgeführt, während ein anderer Prozeß beim Geräte-I/O auf Interrupts wartet oder während ein Prozeß auf einen Takt-Tick wartet.
ATE	Akronym für "Automated test equipment" (Automatisierte Testausrüstung).
Attributknoten	Spezielle Blockdiagrammknoten, die zur Steuerung von Aussehen und Funktionalität von Bedien- und Anzeigeelementen verwendet werden können.
Auffangterminal	Terminal, das Daten aufnimmt. Wird auch als Ziel-Terminal bezeichnet.
Auswahlrechteck	Ein beweglicher, gestrichelter Rand, der ausgewählte Objekte umgibt.
Auto-Indizierung	Fähigkeit von Schleifenstrukturen, Arrays an ihren Rändern zu zerlegen und zusammenzufügen. Wenn ein Array bei aktivierter Funktion "automatische Indizierung" in die Schleife eintritt, zerlegt die Schleife es automatisch mit Skalaren, die aus eindimensionalen Arrays extrahiert wurden, mit eindimensionalen Arrays, die aus zweidimensionalen Arrays extrahiert wurden und so weiter. Schleifen fügen Daten in Arrays zusammen, wenn sie die Schleife verlassen, und zwar im umgekehrten Verfahren.

Automatische Größenanpassung	Automatische Größenanpassung von Labels, damit der eingegebene Text paßt.
Automatische Skalierung	Fähigkeit der Achsen, sich auf den Bereich von Plot-Werten anzupassen. Bei Graph-Achsen bestimmt diese Funktion die Höchst- und Mindestachsenwerte.

B

Bedien- und Anzeigeelemente vom Typ String	Frontpanel-Objekte, die zum Manipulieren und Anzeigen von Eingabe- und Ausgabertext verwendet wird.
Bedienelement	Frontpanel-Objekt zur interaktiven Eingabe von Daten in ein VI bzw. zur programmatischen Eingabe in ein SubVI.
Bedienwerkzeug	Werkzeug, das zur Eingabe von Daten in Bedienelemente sowie zu deren Bedienung verwendet wird. Ähneln im Aussehen einem Zeigefinger.
Bedingtes Terminal	Das Terminal einer While-Schleife, das einen Booleschen Wert enthält, der bestimmt, ob das VI eine weitere Iteration durchführt.
Benutzerdefinierte Konstante	Blockdiagrammobjekt, das einen von Ihnen eingestellten Wert ausgibt.
Benutzerspezifische PICT-Bedienelemente	Bedien- und Anzeigeelemente, dessen Teile durch von Ihnen vorgegebene Graphiken und Anzeigen ersetzt werden können.
Beschreibungsfeld	Online-Dokumentation für ein G-Objekt.
Beschriftungswerkzeug	Werkzeug, das zum Erstellen von Labels und zur Eingabe von Text in Textfenster verwendet wird.
Blockdiagramm	Bildliche Beschreibung oder Darstellung eines Programms oder Algorithmus. In G ist das Blockdiagramm, das aus ausführbaren Icons, den sogenannten Knoten besteht und Verbindungen enthält, mit denen Daten zwischen den Knoten transportiert werden, das Äquivalent zum Quellcode für das VI. Das Blockdiagramm ist im Blockdiagrammfenster des VIs resident.
BNF	Backus-Naur Form. Eine häufig verwendete Darstellung für Sprachgrammatiken in der Computerwissenschaft.

Boolesche Bedien- und Anzeigeelemente	Frontpanel-Objekte, die zum Manipulieren und Anzeigen oder Eingeben und Ausgeben von Booleschen Daten (TRUE oder FALSE) dienen. Es stehen mehrere Stilarten zur Verfügung, wie z.B. Schalter, Tasten und LEDs.
Breakpoint	Eine Pause in der Ausführung, die zum Debugging verwendet wird. Ein Breakpoint wird eingestellt, indem Sie mit dem Breakpoint-Werkzeug aus der Werkzeug -Palette auf ein VI, einen Knoten oder eine Verbindung klicken.
Breakpoint-Werkzeug	Werkzeug, das zum Einstellen eines Breakpoints in einem VI, Knoten oder einer Verbindung verwendet wird.
Byte-Stream-Datei	Datei, die Daten als Sequenz aus ASCII-Zeichen oder Byte speichert.

C

Case	Ein Subdiagramm einer Case-Struktur.
Case-Struktur	Bedingte Verzweigungskontrollstruktur, die nur eines ihrer Subdiagramme basierend auf seinem Eingang ausführt. Sie ähnelt den Anweisungen IF-THEN-ELSE und CASE in Kontrollflußsprachen.
CIN	<i>Siehe</i> Code Interface Node.
Cluster	Ein Satz aus angeordneten, nicht-indizierten Datenelementen eines beliebigen Datentyps einschließlich numerischer, Boolescher, String-, Array oder Cluster-Datentypen. Die Elemente müssen entweder alle Bedienelemente oder alle Anzeigeelemente sein.
Cluster-Shell	Frontpanel-Objekt, das die Elemente eines Clusters enthält.
Code Interface Node (CIN)	Spezieller Blockdiagrammknoten, mit dem Sie konventionellen, textbasierten Code mit einem VI verknüpfen können.
CPU	Central Processing Unit (Zentraleinheit).

D

Darstellung	Untertyp des numerischen Datentyps, wobei es vorzeichenbehaftete und vorzeichenlose Byte-, Word und Long-Integer-Werte sowie Single-, Double- und Extended-Fließkommazahlen in realer und komplexer Form gibt.
Datenabhängigkeit	Zustand in einer Datenflußprogrammiersprache, in welcher ein Knoten nicht ausgeführt werden kann, bis er Daten von einem anderen Knoten erhält. <i>Siehe auch</i> Künstliche Datenabhängigkeit.
Datenerfassung	Auch: DAQ. Prozeß der Erfassung von Daten, normalerweise von A/D- oder digitalen Eingangssteckkarten.
Datenfluß	Programmiersystem, das aus ausführbaren Knoten besteht und in dem Knoten nur dann ausgeführt werden, wenn sie alle erforderlichen Eingabedaten erhalten haben, und das nach ihrer Ausführung die Ausgabe automatisch erzeugt. G ist ein Datenfluß-System.
Datenprotokoll-Datei	Datei, die Daten als Sequenz aus Datensätzen eines einzelnen, arbiträren Datentyps speichern, den Sie bei Erstellung der Datei vorgeben. Alle Datensätze in einem Datenprotokoll müssen zwar vom selben Typ sein, dieser Typ kann jedoch komplex sein; Sie können z.B. vorgeben, daß jeder Datensatz aus einem Cluster mit einem String, einer Zahl und einem Array bestehen muß.
Datenprotokollierung	Im allgemeinen die Erfassung von Daten und das gleichzeitige Speichern der Daten in einer Datei auf einem Speichermedium. I/O-Funktionen von G-Dateien können Daten protokollieren.
Datenspeicherformate	Die Anordnung und Darstellung von Daten, die gespeichert wurden.
Datentyp-Deskriptor	Code, der Datentypen identifiziert; wird zur Datenspeicherung und -darstellung verwendet.
Dauerausführung	Ausführungsmodus, in welchem ein VI wiederholt ausgeführt wird, bis der Bediener es anhält. Diese Funktion wird durch Klicken auf die Schaltfläche zur Dauerausführung aktiviert.
Defektes VI	VI, das nicht kompiliert oder ausgeführt werden kann; wird durch einen unterbrochenen Pfeil in der Taste "Ausführung" dargestellt.
Diagramm	<i>Siehe</i> Oszilloskopdiagramm, Streifendiagramm und Sweepdiagramm.

Dialogfeld	Ein interaktiver Bildschirm mit Abfragen, wobei Sie zusätzliche Informationen eingeben, die zur Vervollständigung eines Befehls benötigt werden.
Dimension	Größe und Struktureigenschaft eines Arrays.
DUT	Akronym für "Device under test" (Gerät im Test).

E

Eindimensional	Mit einer Dimension, wie es bei einem Array der Fall ist, das nur über eine Zeile mit Elementen verfügt.
EOF	End-of-File (Ende der Datei). Zeichen-Offset am Ende der Datei in Relation zum Anfang der Datei (d.h. das EOF ist die Größe der Datei).
Externe Routine	<i>Siehe</i> Freigegebene externe Routine.

F

Farbwerkzeug	Werkzeug, mit dem Vordergrund- und Hintergrundfarben eingestellt werden.
FFT	Akronym für "Fast Fourier Transform" (schnelle Fourier-Transformation).
For-Schleife	Iterative Schleifen-Struktur, die ihr Subdiagramm eine festgelegte Anzahl von Malen ausführt. Entspricht im konventionellen Code: <pre>For i=0 to n-1, do</pre>
Formatumwandlung	Die automatische Konvertierung, die von G zum Ändern der numerischen Darstellung eines Datenelements durchgeführt wird.
Formatumwandlungspunkt	Glyph auf einem Knoten oder Terminal, der angibt, daß sich die numerische Darstellung der Datenelemente an diesem Punkt ändert.
Formelknoten	Knoten, der Formeln ausführt, die Sie als Text eingeben. Besonders nützlich für lange Formeln, die zur Erstellung in Blockdiagrammform zu schwerfällig sind.
Freies Label	Label auf dem Frontpanel oder Blockdiagramm, das zu keinem anderen Objekt gehört.

Frontpanel	Die interaktive Benutzerschnittstelle eines VIs. Wurde entsprechend dem vorderen Bedienfeld von physischen Instrumenten modelliert und besteht aus Schaltern, Schieberegler, Zählern, Graphen, Diagrammen, Meßgeräten, LEDs und anderen Bedien- und Anzeigeelementen.
Funktion	Integriertes Ausführungselement, vergleichbar mit einem Operator, einer Funktion oder Anweisung in einer konventionellen Sprache.
Funktionenpalette	Palette mit Blockdiagrammstrukturen, Konstanten, Kommunikationsfunktionen und VIs.
G	
G	Graphische Programmiersprache, die in LabVIEW und BridgeVIEW verwendet wird.
Gehäuse	Nicht bewegliches Teil von Bedien- und Anzeigeelementen am Frontpanel, das Schieberegler und Skalen enthält.
Gemeinsam genutzte externe Routine	Subroutine, von mehreren CIN-Coderessourcen gemeinsam genutzt werden kann.
Gewandelte Daten	Daten eines beliebigen Typs, die in einen String konvertiert wurden, in der Regel, um sie in eine Datei zu schreiben.
Glyph	Ein kleines Bild oder Icon.
GPIB	General Purpose Interface Bus (Schnittstellenbus für allgemeine Zwecke) ist die gebräuchliche Bezeichnung für das Kommunikationsschnittstellensystem, das im ANSI/IEEE-Standard 488.1-1987 und ANSI/IEEE-Standard 488.2-1987 definiert wurde. Hewlett-Packard, Erfinder dieses Busses, bezeichnet ihn als HP-IB.
Graph-Bedienelement	Frontpanel-Objekt, das Daten in einer kartesischen Ebene anzeigt.
Größenanpassung-Handles	Abgewinkelte Handles an der Ecke von Objekten, die die Punkte zur Größenanpassung bezeichnen.

H

Handle	Zeiger, der auf einen Speicherblock zeigt; verwaltet Referenz-Arrays und -Strings. Ein Array aus Strings ist ein Handle für einen Speicherblock, der Handles für Strings enthält.
Hex	Hexadezimalzahl. Ein System mit der Basiszahl 16.
Hierarchiefenster	Fenster, das die Hierarchie von VIs und SubVIs graphisch anzeigt.
Hierarchische Palette	Menü, das Paletten und Subpaletten, enthält
Highlight-Funktion	Debugging-Funktion, mit der die VI-Ausführung animiert wird, um den Datenfluß im VI zu illustrieren.
Hilfefenster	Spezielles Fenster, in dem die Namen und Positionen der Terminals für eine Funktion oder ein SubVI, die Beschreibung von Bedien- und Anzeigelementen, die Werte von Universalkonstanten und Beschreibungen und Datentypen von Bedienelement-Eigenschaften angezeigt werden. Durch das Fenster wird außerdem die Online-Referenz für G aufgerufen.
Hz	Hertz. Zyklen pro Sekunde.
I	
I/O	Eingang/Ausgang. Die Datenübertragung von oder zu einem Computersystem mit Kommunikationskanälen, Bediener-Eingabegeräten und/oder Datenerfassung und Steuerschnittstellen.
Icon	Graphische Darstellung eines Knotens auf einem Blockdiagramm.
Icon-Editor	Schnittstelle, die der des Paint-Programms zur Erstellung von VI-Icons ähnelt.
Iconfeld	Bereich in der oberen rechten Ecke des Frontpanels und Blockdiagramms, in dem das VI-Icon angezeigt wird.
IEEE	Akronym für "Institute for Electrical and Electronic Engineers" (Institut der Elektro- und Elektronikingenieure).
Inf	Digitaler Anzeigewert für eine Fließkommadarstellung von Unendlich.

Inplace-Ausführung	Fähigkeit einer Funktion oder eines VIs, Speicher neu zu verwenden anstatt neuen Speicher zuzuweisen.
Instrumententreiber	VI, das ein programmierbares Instrument steuert.
Iterationsterminal	Das Terminal einer For-Schleife oder While-Schleife, das die aktuelle Anzahl von abgeschlossenen Iterationen enthält.

K

Kein Pfad	Ein vordefinierter Wert für das Pfad-Bedienelement, der bedeutet, daß der Pfad ungültig ist.
Keine Refnum	Ein vordefinierter Wert, der bedeutet, daß die Refnum ungültig ist.
Klonen	Anfertigen einer Kopie eines Bedienelements oder eines anderen G-Objekts durch Klicken der Maustaste auf das Objekt und gleichzeitigem Drücken der <Strg>-Taste (Windows), <Optionstaste> (Macintosh), <Meta>-Taste (Sun) oder <Alt>-Taste (HP-UX) und Ziehen der Kopie zur neuen Position. (UNIX) Sie können ein Objekt auch klonen, indem Sie mit der mittleren Maustaste auf das Objekt klicken und die Kopie dann zur neuen Position ziehen.
Knoten	Ausführungselemente eines Blockdiagramms, das aus Funktionen, Strukturen und SubVIs besteht.
Knoten bündeln	Funktion, mit der Cluster aus verschiedenen Elementtypen erstellt werden.
Kompilieren	Prozeß, mit dem Code der höchsten Ebene in maschinenausführbaren Code konvertiert wird. VIs werden automatisch kompiliert, bevor sie nach ihrer Erstellung oder nach einer Änderung zum erstenmal ausgeführt werden.
Konstante	<i>Siehe</i> Universalkonstante und benutzerdefinierte Konstante.
Kontrollfluß	Programmierungssystem, in welchem die sequentielle Reihenfolge von Anweisungen die Ausführungsreihenfolge bestimmt. Die meisten konventionellen textbasierten Programmiersprachen, wie C, Pascal und BASIC, sind Kontrollflußsprachen.

Konvertierung	Änderung des Typs eines Datenelements.
Künstliche Datenabhängigkeit	Zustand in einer Datenflußprogrammiersprache, in welchem das Eingehen von Daten, nicht die Datenwerte, die Ausführung eines Knotens triggert.

L

Label	Textobjekt, das zur Benennung oder Beschreibung anderer Objekte oder Bereiche auf dem Frontpanel oder Blockdiagramm dienen.
LED	Light-emitting diode (Luminiszenzdiode).
Leeres Array	Array, das null Elemente hat, jedoch über einen definierten Datentyp verfügt. Beispielsweise ist ein Array, das über ein numerisches Bedienelement in seiner Datenanzeige verfügt, jedoch keine definierten Werte für irgendein Element hat, ein leeres numerisches Array.
Legende	Objekt, das einem Diagramm oder Graphen zugehörig ist, auf dem die Namen und Plotstile von Plots auf diesem Diagramm oder Graphen angezeigt werden.

M

Matrix	Zweidimensionales Array.
MB	Megabyte (Speicherkapazität).
Menüleiste	Horizontale Leiste, die Namen von Hauptmenüs enthält.
Mnemonic	Ein String, der mit einem Integer-Wert assoziiert ist.
Modulare Programmierung	Programmierung, die austauschbare Computer-Routinen verwendet.
Multithread	Eine Multithread-Anwendung ist eine Anwendung, bei der mehrere verschiedene Threads unabhängig voneinander ausgeführt werden. Bei einem Computer mit Mehrfachprozessor können die unterschiedlichen Threads gleichzeitig auf unterschiedlichen Prozessoren laufen.

N

Namen-Label	Das Label eines Frontpanel-Objekts, das zur Benennung des Objekts sowie zu seiner Unterscheidung von anderen Objekten dient. Wird auf dem Terminal, auf lokalen Variablen und Attributknoten verwendet, die Teil des Objekts sind. <i>Siehe</i> auch Untertitel-Label.
NaN	Digitaler Anzeigewert für eine Fließkommadarstellung von <i>Keine Zahl</i> , was in der Regel das Ergebnis einer undefinierten Operation, wie z.B. $\log(-1)$, ist.
Nicht darstellbare Zeichen	ASCII-Zeichen, die nicht angezeigt werden können, so wie Null, Rückwärtsschritt, Tabulator usw.
Numerische Bedien- und Anzeigeelemente	Frontpanel-Objekte, die zum Manipulieren und Anzeigen von numerischen Eingabe- und Ausgabedaten dienen.

O

Objekt	Allgemeiner Begriff für alle Elemente auf dem Frontpanel oder Blockdiagramm, einschließlich Bedienelementen, Knoten, Verbindungen und importierten Bildern.
Oszilloskopdiagramm	Numerisches Anzeigeelement, das entsprechend der Funktionsweise eines Oszilloskops modelliert ist.

P

Palette	Menü aus Bildern, die mögliche Optionen darstellen.
Palette der Bedienelemente	Palette mit Bedien- und Anzeigeelementen am Frontpanel; Elemente-Palette.
Pixmap	Ein Standardformat zum Speichern von Bildern, in welchem jedes Pixel durch einen Farbwert dargestellt wird. Ein Bitmap ist eine schwarz-weiße Version eines Pixmap.
Plattform	Computer plus Betriebssystem.
Plot	Eine graphische Darstellung eines Array aus Daten, die entweder auf einem Graphen oder einem Diagramm angezeigt wird.

Polymorphismus	Fähigkeit eines Knotens, sich automatisch an Daten unterschiedlicher Repräsentierung, Struktur oder unterschiedlichen Typs anzupassen.
Popup-Menü, aufrufen	Ein besonderes Menü aufrufen, indem Sie mit der rechten Maustaste (bei Windows und UNIX-Plattformen) auf ein Objekt klicken oder indem Sie die Befehlstaste gedrückt halten und das Objekt anklicken (Macintosh). Die Menüoptionen beziehen sich speziell auf das angeklickte Objekt.
Popup-Menüwerkzeug für Objekte	Werkzeug, das zum Aufrufen des Popup-Menüs für ein Objekt verwendet wird.
Positionierwerkzeug	Werkzeug, das zum Verschieben, Auswählen und zur Größenanpassung von Objekten verwendet wird.
Probe	Debugging-Funktion zum Prüfen von Zwischenwerten in einem VI.
Probe-Werkzeug	Werkzeug, das zur Erstellung von Proben auf Verbindungen verwendet wird.
Programmatisches Drucken	Automatisches Drucken des Frontpanels eines VIs nach seiner Ausführung.
Pseudocode	Vereinfachte sprachunabhängige Darstellung des Programmiercodes.
Pulldown-Menüs	Menüs, die von einer Menüleiste aus aufgerufen werden. Pulldown-Menüoptionen sind normalerweise allgemeiner Natur.

Q

Quell-Terminal Terminal, das Daten ausgibt.

R

Rahmen Subdiagramm einer Sequenzstruktur.

Refnum Ein Identifikator, den G mit einer Datei assoziiert, wenn Sie ihn öffnen. Sie verwenden die Datei-Refnum, um anzugeben, daß eine Funktion oder ein VI eine Operation an der offenen Datei durchführen soll.

Ring-Bedienelement Spezielles numerisches Bedienelement, das 32-Bit Integer-Werte, die mit 0 beginnen und der Reihe nach hochgezählt werden, mit einer Reihe von Text-Labels oder Graphiken assoziiert.

Rollwerkzeug Werkzeug, das zum Rollen durch Fenster verwendet wird.

S

Schieber Beweglicher Teil von Bedien- und Anzeigeelementen vom Typ Schieberegler.

Schieberegister Optionaler Mechanismus in Schleifenstrukturen, der zur Übergabe des Wertes einer Variable von einer Iteration einer Schleife zur darauffolgenden Iteration verwendet wird.

Sequenz-Variable Terminal, das Daten zwischen den Rahmen einer Sequenzstruktur übergeben.

Sequenzstruktur Programmkontrollstruktur, die ihre Subdiagramme in numerischer Reihenfolge ausführt. Wird im allgemeinen dazu verwendet, nicht datenabhängige Knoten zur Ausführung in einer gewünschten Reihenfolge zu zwingen.

Skala Teil mit Schaltverhalten von Bedien- und Anzeigeelementen vom Typ Diagramm und Graph, der bei bekannten Intervallen eine Reihe von Markierungen oder Punkte besitzt, die Maßeinheiten darstellen.

Skalar Zahl, die durch einen Punkt auf einer Skala dargestellt werden kann. Ein einzelner Wert im Gegensatz zu einem Array. Boolesche Skalare und Cluster sind explizit einfache Instanzen ihrer jeweiligen Datentypen.

Streifendiagramm Ein numerisches Plotting-Anzeigeelement, das einem Papierstreifenschreiber nachempfunden ist, der beim Plotten von Daten nach unten läuft.

Struktur Programmkontrollelement, wie z.B. Sequenz, Case, For-Schleife oder While-Schleife.

Subdiagramm Blockdiagramm innerhalb des Rahmens einer Struktur.

SubVI VI, das im Blockdiagramm eines anderen VIs verwendet wird; vergleichbar mit einer Subroutine.

Sweepdiagramm Ähnlich dem Oszilloskop-Diagramm, mit der Ausnahme, daß eine vertikal verlaufende Linie alte und neue Daten voneinander abgrenzt.

Symbolleiste Leiste mit Befehlsschaltflächen, die zur Ausführung und zum Debugging von VIs verwendet werden können.

T

Tabellengesteuerte Ausführung Eine Methode zur Ausführung, mit welcher einzelne Tasks separate Cases in einer Case-Struktur darstellen, die in einer While-Schleife eingebettet ist. Sequenzen werden als Arrays aus Case-Nummern angegeben.

Terminal Objekt oder Bereich eines Knotens, durch welches bzw. welchen Daten übergeben werden.

Tip-Strip Kleine gelbe Textstreifen, die den Terminalnamen bezeichnen und die Identifizierung von Funktions- und Knotenterminals zur Verbindung erleichtern.

Top-Level VI VI auf der höchsten Ebene der VI-Hierarchie. Dieser Begriff unterscheidet das VI von seinen SubVIs.

Tunnel Terminal für den Dateneintritt oder -austritt in einer Struktur.

Typ-Deskriptor *Siehe* Datentyp-Deskriptor.

U

Universalkonstante Nicht bearbeitbares Blockdiagrammobjekt, das ein bestimmtes ASCII-Zeichen oder eine standardmäßige numerische Konstante, zum Beispiel π , ausgibt.

Untertitel-Label Label auf einem Frontpanel-Objekt, das zur Benennung des Objekts in der Benutzerschnittstelle dient. Dieser Name kann in andere Sprachen übersetzt werden, ohne daß der Quellcode des Blockdiagramms davon berührt wird. *Siehe* auch Namen-Label.

UUT Akronym für "Unit under test" (Einheit im Test).

V

Verbindung	Datenpfad zwischen Knoten.
Verbindungswerkzeug	Werkzeug, das zur Definition von Datenpfaden zwischen Quell- und Auffangterminals verwendet wird.
VI	<i>Siehe</i> Virtuelles Instrument.
VI-Anwendungsklasse	Eine Referenz auf eine Anwendung, die virtuelle Instrumente laden kann, die einen Zugriff auf die Eigenschaften und Methoden der Anwendung ermöglichen.
VI-Bibliothek	Spezielle Datei, die eine Sammlung von zusammenhängenden VIs für eine bestimmte Verwendung enthält.
VI-Gerüst	Ein nicht funktioneller Prototyp eines SubVIs. Er verfügt über Eingänge und Ausgänge, ist jedoch unvollständig. Er wird in der frühen Planungsphase eines VI-Designs als Platzhalter für ein VI verwendet, das später noch entwickelt wird.
VI-Klasse	Eine Referenz zu einem virtuellen Instrument, das Zugriff auf VI-Eigenschaften und Methoden ermöglicht.
VI-Server	Mechanismus zur programmatischen Bedienung von VIs und G-Anwendungen. Kann auch zur Fernbedienung von VIs oder G-Anwendungen verwendet werden.
VI-Stringdatei	Eine Tagged-Textdatei mit allen Lokalisierungs-Strings, die im Frontpanel eines VIs vorkommen.
Virtuelles Instrument	Programm in LabVIEW oder BridgeVIEW; wird so bezeichnet, weil es das Aussehen und die Funktion eines echten Instruments nachahmt.

W

Wandeln	Ändern des Typ-Deskriptors eines Datenelements, ohne das Speicherbild der Daten zu ändern.
Werkzeug	Spezieller Cursor, den Sie zur Durchführung spezifischer Operationen verwenden.

Werkzeug “Farbe kopieren”	Kopiert Farben zum Einfügen mit dem Färbewerkzeug.
Werkzeugpalette	Palette mit Werkzeugen, die zur Bearbeitung und zum Debugging von Frontpanel- und Blockdiagrammobjekten verwendet werden können.
Wettlaufsituation	Tritt dann auf, wenn zwei oder mehr Codeteile, die parallel ausgeführt werden, den Wert derselben gemeinsam genutzten Ressource ändern, normalerweise eine globale oder lokale Variable.
While-Schleife	Schleifenstruktur, die einen Codeabschnitt solange wiederholt, bis eine Bedingung erfüllt wird. Vergleichbar mit einer Do-Schleife oder einer Repeat-Until-Schleife in konventionellen Programmiersprachen.

Z

Ziehen	Einen Maus-Cursor auf dem Bildschirm ziehen, um Objekte auszuwählen, zu verschieben, zu kopieren oder zu löschen.
Ziel-Terminal	<i>Siehe</i> Auffangterminal.
Zweidimensional; 2D	Objekte mit zwei Dimensionen, wie z.B. in einem Array mit mehreren Zeilen und Spalten.

Stichwörterverzeichnis

Symbole

`\` Option Code-Anzeige, SER.: *Siehe* auch umgekehrte Schrägstrich-(\) Codes., 11-2, 11-3

Zahlen

32-bit Fließkommazahlen mit einfacher Genauigkeit. *Siehe* Single-Fließkommazahlen (SGL).

64-bit Fließkommazahlen mit doppelter Genauigkeit. *Siehe* Double- (DBL) Fließkommazahlen.

A

Abbruchtaste. *Siehe* Taste Stopp.

Abfrage von Datensätzen

Abfrageprogrammierung, 4-7 bis 4-9

Abgeleitete Einheiten (Tabelle), 9-28

Ablaufen. *Siehe* Ausführen.

Ablaufinvariante Ausführbarkeit

Beispiele

Speicher-VI, das nicht auf die Freigabe von Daten ausgelegt ist, 26-16 bis 26-17

VIs mit Wartefunktion, 26-15 bis 26-16

Absolute Zeit und Datum formatieren

digitale Anzeigen, 9-10 bis 9-11

Graphenanzeigeelemente, 15-14 bis 15-17

Achsenmarker, Ändern, 9-15 bis 9-18

ActiveX-Container, 16-2 bis 16-5

Dialogfeld ActiveX-Objekte auswählen, 16-3 bis 16-5

eingebettete Bedienelemente, 16-2

eingebettete Dokumente, 16-2

Objekte einfügen, 16-3

Zweck und Verwendung, 16-3

ActiveX-Palette, Erstellen von, 16-5

ActiveX-Unterpalette, 16-1

AESEnd Print Document-VI. *Siehe* Sendet AE Dokument-drucken.

Aktion Latch, wenn losgelassen, 24-14

Aktion Schaltet wenn losgelassen, 24-14

Aktualisierungsmodus für Oszilloskopdiagramm, 15-26

Aktualisierungsmodus für Streifendiagramm, 15-26

Alle Terminals trennen, 3-10

<Alt-b>, unterbrochene Verbindungen entfernen, 4-9

<Alt-f>, Dialogfeld Suche aufrufen, 3-23

<Alt-h>, Hilfetaste, 1-7

<Alt-klicken>

Aktion Hierarchie anzeigen ausführen, 3-21

Farben kopieren und übertragen, 2-28

<Alt-m>, zwischen Ausführungs- und Bearbeitungsmodus wechseln, 6-5

<Alt-Return>, neue Zeilen einbetten, 7-10

<Alt>-Taste

Attributknoten klonen, 22-4

Blockdiagramm eines SubVIs in den Vordergrund holen, 4-22

Herausspringen, 4-21

Hineinspringen, 4-21

Kurzbefehl Überspringen, 4-21

Menüpunkte auswählen (Hinweis), 6-10

Neuen Skalenmarker erstellen, 9-18

Objekte klonen, 2-14

Werkzeuge des Icon-Editors in Pipette umwandeln, 3-5

zwischen Array-Elementen wechseln, 14-14

zwischen Cluster-Elementen wechseln, 14-22

- <Alt-Umschalt>, temporäre Werkzeugpalette aufrufen, 2-4
- An Typ anpassen, für Aufruf ext. Bibliotheken, 25-7
- Ändern der Größe
 - Tabellen, Zeilen und Spalten, 11-7 bis 11-8
- Anhalten von VIs, 4-4
- Anpassen der Menüleisten. *Siehe* Menü-Editor ;Umgang mit Menüauswahl.
- Anpassen der Umgebung von G. *Siehe* G-Umgebung, Anpassen.
- Anpassungsmodus, 24-6 bis 24-19
 - Auswählen, 24-6
 - Bedienelemente als Elemente, 24-16 bis 24-17
 - Das Elemente-Fenster des Bedienelement-Editors, 24-8 bis 24-9
 - gestalterische Elemente, 24-10 bis 24-15
 - Hinzufügen von gestalt. Elementen zu Bedienelementen, 24-17 bis 24-18
 - Popup-Menüs für verschiedene Elemente, 24-10
 - Textelemente, 24-15
 - unabhängige Bestandteile, 24-7 bis 24-8
- Anschlüsse
 - Ausgangsanschlüsse, 1-5
 - Definition, 1-6, 3-2
 - Eingangsanschlüsse, 1-5
 - Terminalmuster
 - auswählen und modifizieren, 3-6
 - definieren, 3-5 bis 3-6
 - erforderliche, empfohlene und optionale Verbindungen, 1-8, 3-9
 - Hierarchiefenster, 3-15 bis 3-22
 - räumliche Verteilung ändern, 3-7
 - Terminals Bedien- und Anzeigeelementen zuweisen, 3-7 bis 3-8
 - Verbindungen bestätigen, 3-10
 - Verbindungen löschen, 3-9 bis 3-10
 - Terminalmuster. *Siehe* auch Terminalanschlüsse., 3-6
 - Überblick, 1-6
 - verglichen mit Parametern, 1-7
 - Zugriff auf Option Anschluß anzeigen, 1-7
- Anwendungen, Verwalten, 5-15, 27-1 bis 27-13
 - Historie, Pflegen
 - Beispiel (Abbildung), 27-11, 27-12
 - Drucken der Historie-Informationen, 27-14
 - Historie-Fenster, 27-10 bis 27-14
 - Kommentare aufzeichnen (Hinweis), 27-10
 - Revisionsnummern, 27-12 bis 27-13
 - Rücksetzen der Historie-Informationen, 27-13
 - Verwandte Dialogfeldopt. für VI-Einstellung und Voreinstellungen, 27-14
 - Sichern von Dateien, 27-2
 - Verteilung von VIs, 27-3 bis 27-8
 - Anwendungsverteilung, 27-7
 - Aufbewahrung der Originalkopien, 27-9
 - Diagramme entfernen, 27-8
 - Entwicklungsvertrieb, 27-7
 - geänderte VIs, 27-7
 - mehrere Entwickler, 27-9 bis 27-14
 - neues Paßwort anwenden, 27-8
 - Option Mit Optionen speichern, 27-3, 27-6 bis 27-8
 - Paßwörter entfernen, 27-8
 - paßwortgeschützte VIs, 27-4
 - Speichern anpassen, 27-8
 - Speichern von VIs in Bibliotheken, 2-33 bis 2-34, 27-1 bis 27-2
 - ungeänderte VIs, 27-8
 - Vorlage, 27-8

- Anwendungen, Verwalten. *Siehe* auch Code aus anderen Sprachen aufrufen.
- Anwendungs- oder VI-Refnum, 12-5
- Anwendungseigenschaften und -methoden. *Siehe* Eigenschaften und Methoden.
- Anwendungselementtyp, Menü-Editor, 6-10
- Anwendungsschriftart (Font), 2-19
- Anzeige VI-Aufrufer wird ausgeführt, 4-1
- Anzeigeelemente
 - allgemeine Optionen, 8-2 bis 8-4
 - ausschneiden, kopieren und einfügen, 8-3
 - Definition, 1-3
 - ersetzen, 8-4 bis 8-6
 - in Bedienelemente ändern, 8-3, 18-12
 - Löschung nicht zulässig, 2-15
 - Terminals zu Bedienelementen zuweisen, 3-7 bis 3-8
 - verbergen, 2-15
- AppleEvent-VIs, zum Ausführen anderer Anwendungen, 25-2
- Arbeitsspeicher
 - Fragen zu, B-4
- Arrays und Array-Bedienelemente, 14-2 bis 14-20
 - Arraygröße feststellen, 14-14
 - ausführen, 14-11 bis 14-17
 - Beispiele, 14-2 bis 14-5
 - Beschreibung, 14-2 bis 14-5
 - Datenspeicherformate, A-4
 - Datentyp, 25-6
 - Definition, 14-2
 - dreidimensional, 14-11
 - eindimensional, 14-3, 14-10
 - erstellen, 14-5 bis 14-11
 - Array-Dimensionen, 14-8
 - Array-Shell mit zulässigen Elementen kombinieren, 14-6
 - Arraytyp definieren, 14-6
 - Popup-Menü der Indexanzeige, 14-7
- G-Arrays und Arrays in anderen Systemen, 14-17 bis 14-20
- gewandelte Daten, A-14
- Graph-Array (Beispiel), 14-3
- Größe ändern, 14-13, 14-14
- in Einzelelement- oder Tabellenform anzeigen, 14-10 bis 14-11
- Indexanzeige
 - Array-Shell, 14-5
 - in Einzelelement- oder Tabellenform anzeigen, 14-10 bis 14-11
 - interpretieren, 14-9
 - Popup-Menü, 14-7
- Indizes, 14-2 bis 14-5
- Kurvenform-Array
 - Beispiel, 14-3
 - zweidimensional (Beispiel), 14-4
- LabVIEW-Arrays und andere Systeme, 14-17 bis 14-20
- leere Arrays, 14-12
- multidimensional, 14-11
- Popup-Menü der Indexanzeige, 14-7
- Standardgröße und -werte einstellen, 14-12 bis 14-13
- String-Array (Beispiel), 14-2
- Typ-Deskriptoren, A-10
- verschieben, 14-14
- Verwendung, 14-5
- Zellen auswählen, 14-15
- zweidimensional, 14-3, 14-9
- zwischen den Elementen wechseln, 14-11
- Array-Shell
 - auf das Frontpanel plazieren, 14-6
 - Bedienelemente erstellen, 14-6
 - Elementanzeige, 14-6
 - Indexanzeige, 14-6
 - Komponenten einer, 14-6
 - undefiniert, 14-12
- Attribut Aktiver Plot, 22-10
- Attribut Blinken, 22-8

- Attribut Bounds (Grenzen)
 - (schreibgeschützt), 22-9
- Attribut Color Table (Farbtabelle), 15-37
- Attribut Farbtabelle, 15-37
- Attribut Z Scale Info
 - Color Array (Array aus Farben), 15-37
 - High Color, 15-37
 - Low Color, 15-37
- Attributknoten, 22-1 bis 22-6
 - alle Attribute gleichzeitig einstellen, 22-7
 - Beispiele, 22-10 bis 22-13
 - Anbieten von Optionen für Benutzer, 22-13
 - Ändern der Plotfarbe eines Diagramms, 22-10
 - Beispiele in der examples.llb (Beispielbibliothek), 22-6
 - boolesch, 22-11
 - Einstellen der Strings für Ring-Steuerungen, 22-11 bis 22-12
 - Listenelemente, 22-12
 - programmatisches Lesen der Cursor, 22-14
 - erstellen, 22-1 bis 22-6
 - Attribute lesen oder schreiben, 22-2
 - mehr als einen Knoten, 22-4
 - Terminale mit Attributen assoziieren, 22-5
- Grundattribute, 22-7 bis 22-10
 - aktiver Plot, 22-10
 - Blinken, 22-8
 - Bounds (Grenzen)
 - (schreibgeschützt), 22-9
 - Disabled (Deaktiviert), 22-7
 - Doppelklicken, 22-12
 - Key Focus (Fokus), 22-8
 - Plotfarbe, 22-10
 - Position, 22-9
 - Hilfenfenster, 1-10, 22-6
 - Popup-Menü, 22-1
 - Popup-Menü Suchen, 3-27
 - suchen, 22-6
 - Zyklen in SubVIs vermeiden
 - Attributknoten in Case-Strukturen, 3-13
 - Attributknoten in Schleifen, 3-12
- Auffangterminals
 - Definition, 1-5
 - Verbindungen, 1-6
- Aufrufen von Code aus anderen Sprachen, 25-2 bis 25-14
 - AppleEvent-VIs, 25-2
 - Ausführen anderer Anwendungen von Ihren VIs aus, 25-1 bis 25-3
 - Code Interface Nodes (CIN), 25-3
 - Verwendung von LabWindows/CVI-Funktions-Panel-Konverter, 25-9 bis 25-14
 - VI System Exec, 25-1
- Aufrufen von Code aus anderen Sprachen.
 - Siehe* auch Anwendungen, Verwalten; Funktion Aufruf von Bibliotheken.
- Aufrufkette, Lesen, 4-21
- Aufrufkettengang, 4-21
- Aufzählungskonstanten, 17-6
- Ausführen von VIs
 - während der Ausführung bearbeiten, 4-2 bis 4-3
 - Popup-Menüs und Optionen, 4-2 bis 4-3
 - zwischen Ausführungs- und Bearbeitungsmodus wechseln (Hinweis), 6-5
- Ausführung mittels Steuerungsfluß
 - Definition, 1-6
- Ausführung von VIs, 4-1
 - mehrere VIs, 4-2
 - VI-Aufrufer, 4-1
 - wiederholte Ausführung von VIs, 4-4

- Ausführungsmodus
 - Probleme, die durch Verbergen der Menüleiste und Werkzeugpalette ausgelöst werden (Hinweis), 6-5
 - Ausführungsoptionen, 6-2 bis 6-3
 - Ablaufinvariante Ausführung, 6-3
 - Nach Abarbeitung schließen, 6-2
 - Nach Öffnen ausführen, 6-3
 - Option Frontpanel nach dem Laden anzeigen, 6-2
 - Priorität, 6-3
 - Ausführungspalette. *Siehe* Werkzeugpalette, Ausführung.
 - Ausgangsanschlüsse, 1-5
 - Ausgangsterminals
 - Definition, 1-5
 - Verbindungen, 1-6
 - Ausgefüllte Schieberegler
 - Zweck und Verwendung, 9-20
 - Ausgefülltes Rechteck-Werkzeug,
 - Icon-Editor, 3-4
 - Ausrichtungsring, 2-12
 - Ausschließen von Diagrammabschnitten, 4-32
 - Auswahl aufheben, 2-8
 - Auswählen-Werkzeug, Icon-Editor, 3-4
 - Auswahlrechteck, 2-8
 - Auto-Indizierung
 - Definition, 19-7
 - Erschöpfung der Speicherressourcen (Hinweis), 19-9
 - For-Schleifenzähler, 19-8 bis 19-9
 - While-Schleifen, 19-9
 - Zweck und Verwendung, 19-7
 - Autom. Skalierung
 - Cluster, 14-24
 - Autom. Skalierung X, 15-12, 15-17
 - Autom. Skalierung Y, 15-12, 15-17
 - Automation RefNum, 12-5
 - Automatische Größenanpassung
 - Labels, 2-26
 - Automatisches Aktualisieren von Typendefinitionen, 24-21
 - Automatisierung. *Siehe* Automation.
 - Auto-Probe während Highlight-Option, auswählen, 7-13
- ## B
- Basiseinheiten (Tabelle), 9-28
 - Bedien- und Anzeigeelement vom Typ ActiveX, 16-2
 - Bedien- und Anzeigeelemente mit Schieberegler
 - ausgefüllte Schieberegler, 9-20 bis 9-21
 - Schieberegler-Skala
 - logarithmischer Skalenabstand, 9-15
 - Bedien- und Anzeigeelemente vom Typ Pfad
 - Zweck und Verwendung, 12-1 bis 12-2
 - Bedien- und Anzeigeelemente vom Typ String, 11-1 bis 11-6
 - Eingabe von nicht anzeigbaren Zeichen, 11-2
 - Option Auf eine Zeile begrenzen, 11-6
 - Option Paßwort-Anzeige, 11-6
 - Optionen, 11-3 bis 11-6
 - Überblick, 11-1 bis 11-2
 - Verwendung des Rollbalkens, 11-3
 - Bedien- und Anzeigeelemente. *Siehe* auch spezielle Typen.
 - Anpassen durch Import von Grafiken, 8-10
 - Anpassen von Steuerungen in Dialogfeldern, 8-9 bis 8-10
 - Anzeige- in Bedienelement ändern, 18-12
 - automatisches Erstellen, 17-13
 - Bedienelemente als Elemente, 24-16
 - Definition, 1-3

- Drucken (Option Bedienelemente), 5-7
 - Option Datentypinformationen aufnehmen, 5-7
- Ersetzen von Bedienelementen, 8-4 bis 8-6
- freie Labels, 2-17
- Optionen des Popup-Menüs Bedien- und Anzeigeelemente, 8-2 bis 8-4
 - Attributknoten erstellen, 8-3
 - ersetzen, 8-4 bis 8-6
 - In Anzeigeelement ändern, 8-3
 - In Bedienelement ändern, 8-3
 - Popup-Menü (Abbildung), 8-3
 - Tastenbelegung, 8-6 bis 8-8
 - Terminal suchen, 8-3
 - Untermenü Anzeigen, 8-3
 - Untermenü Datenoperationen, 8-3
- Panelreihenfolge, 8-8 bis 8-9
- Tags für VI-Stringdateien (Tabelle), 29-8
- Terminals, 17-2
 - Symbole (Tabelle), 17-2
- Zuweisen von Terminals, 3-7
- Bedien- und Funktionspalette, Anpassen, 7-33 bis 7-38
 - Ansichten, 7-37
 - Erstellen von Unterpaletten, 7-36 bis 7-37
 - Hinzufügen von VIs und Bedienelementen zu user.lib und instr.lib, 7-34
 - Installieren und Ändern von Ansichten, 7-34
 - Paletten-Editor, 7-35
 - Verschieben von Unterpaletten, 7-37
- Bedienelemente
 - allgemeine Optionen, 8-2 bis 8-6
 - ausschneiden, kopieren und einfügen, 8-3
 - Bedienelemente als Elemente, 24-16 bis 24-17
 - Definition, 1-3
 - ersetzen, 8-4 bis 8-6
 - in Anzeigeelemente ändern, 8-3
 - Key Navigation... (Dialogfeld), 8-6 bis 8-8
 - Löschung nicht zulässig, 2-15
 - Panelreihenfolge, 8-8 bis 8-9
 - Terminals zu Bedienelementen zuweisen, 3-7
 - verbergen, 2-15
- Bedienelemente vom Typ Ring, 13-8 bis 13-12
 - Ändern von Größe und Text eines Text- & Bild-Rings, 13-11 bis 13-12
 - Definition, 13-8
 - Zweck und Verwendung, 13-8 bis 13-9
- Bedienelemente von Dialogfeldern, Anpassen, 8-9
- Bedienelemente. *Siehe* auch Elemente.
- Bedienelement-Editor
 - Abbildung, 24-2
 - Das Elemente-Fenster, 24-8 bis 24-9
 - durch Doppelklick öffnen, 7-10
 - Zweck und Verwendung, 24-2 bis 24-3
- Bedienelemente-Palette
 - Abbildung, 1-4, 8-1
 - Auswahl von Bedienelementen, 4-26
 - Hinzufügen von benutzerspezifischen Bedienelementen, 24-1, 24-5
 - Numerische Option, 9-1
 - Objekte auswählen, 2-1 bis 2-4
 - Optionen, 8-1
 - Popup-Menü (Hinweis), 2-2
 - temporäre Kopie (Hinweis), 2-2
 - verfügbare Bedienelemente, 8-1 bis 8-2
 - Zugriff auf die Cluster-Shell, 14-21
- Bedienwerkzeug
 - Abbildung, 2-5
 - Erhöhen oder Verringern von numerischen Bedienelementen, 9-2
- Bedingungsterminal, 19-4

- Befehl
 - Abbrechen, 4-4
 - Stopp, 4-4
- Befehl Abstand zwischen Elementen einfügen, 14-15
- Befehl Aktuellen Wert als Standard, 14-23
- Befehl Alle Terminale trennen, 3-10
- Befehl Ausführen, 4-4
- Befehl Ausschneiden, Menü Bearbeiten, 2-14
- Befehl Autom. Skalierung, 14-24
- Befehl Bild importieren, 8-11, 24-11
- Befehl Dieses Terminal trennen, 3-10
- Befehl Dimension hinzufügen, 14-8
- Befehl Entfernen, 17-10
- Befehl Hinzufügen, 17-10
- Befehl Leeres Array, 14-13
- Befehl Objekt deaktivieren, 13-10
- Befehl Objekt entfernen, 13-10
- Befehl Untertitel (Caption), 2-16
- Befehl Wiederherstellen, 7-32
- <Befehl-b>, unterbrochene Verbindungen entfernen, 4-9
- <Befehl-f>, Dialogfeld Suche aufrufen, 3-23
- <Befehl-h>, Hilfetaste, 1-7
- <Befehl-M> Zugriffstaste für Bearbeiten/Ausführen (Hinweis), 6-5
- <Befehlstaste-klicken>
 - Farben kopieren und übertragen, 2-28
- <Befehl>-Taste
 - Kurzbefehl Herausspringen, 4-21
 - Kurzbefehl Hineinspringen, 4-21
 - Kurzbefehl Überspringen, 4-21
 - zwischen Arrayelementen wechseln, 14-14
 - zwischen Cluster-Elementen wechseln, 14-22
- <Befehl-Umschalt>, temporäre Werkzeugpalette aufrufen, 2-4
- Beispiele in examples.llb.vi (Beispielbibliothek-VI), 19-14
- Benutzerdefinierte Konstanten, 17-4 bis 17-8
- Benutzerspezifische Bedienelemente
 - Aktuelles Element, 24-8
 - Bedienelemente als Elemente, 24-16 bis 24-17
 - Bedienelement-Editor, 24-2
 - Das Elemente-Fenster des Bedienelement-Editors, 24-8 bis 24-9
 - Erstellen von Icons, 24-5
 - gestalt. Elemente, 24-10 bis 24-15
 - mehrere Bilder, 24-13
 - unabhängige Bilder, 24-14 bis 24-15
 - gültige benutzerspezifische Bedienelemente, 24-4
 - Hinzufügen von gestalt. Elementen zu Bedienelementen, 24-17 bis 24-18
 - Hinzufügen zur Palette Elemente, 24-1
 - Öffnen, 24-6
 - Popup-Menüs für verschiedene Elemente, 24-10
 - Speichern, 24-5
 - Textelemente, 24-15
 - unabhängig von Quelldatei, 24-6
 - unabhängige Bestandteile, 24-7 bis 24-8
 - Verwenden von Änderungen von benutzerspezifischen Bedienelementen, 24-4 bis 24-5
 - Verwendung von, 24-5
 - Vorbehalte gegen, 24-18 bis 24-19
 - Zweck und Verwendung, 24-1
- Benutzerspezifisches Anpassen von LabVIEW. *Siehe* Voreinstellungen.
- Benutzerspezifisches Format, Dialogfeld Dokumentation drucken, 5-4 bis 5-5
- Bereichsfehler
 - Ergebnisse von Werten außerhalb des Bereichs (Hinweis), 4-18
 - Korrektur, 4-18
- Bereichsfehler-Anzeigeelement, 4-18

- Bereichsgrenzen
 - Boolesche Bereichsprüfung, 10-4
 - numerische Bedien- und Anzeigeelemente, 9-5 bis 9-8
 - Extended Fließkommazahlen (Hinweis), 9-5
 - numerische Bereichsprüfung, 9-6 bis 9-8
 - Tabelle, 9-5
- Bereichsoptionen für numerische Bedien- und Anzeigeelemente, 9-5
 - Dialogfeld Datenbereich, 9-6
 - Fließkommazahlen (Tabelle), 9-5
 - numerische Bereichsprüfung, 9-6 bis 9-8
 - Berichtigen ungültiger Werte, 9-8
 - Option Ignorieren, 9-6
 - Option Unterdrücken, 9-7
- Beschreibungen, 5-7
 - bearbeiten, 2-29
 - erstellen, 2-29 bis 2-31
 - im Hilfefenster anzeigen, 1-7
- Beschreibungen. *Siehe* auch Option Dokumentation.
 - Dialogfeld VI-Setup erstellen
 - Objekte, 2-29
 - VIs, 2-30 bis 2-31
 - Unmöglichkeit, Beschreibungen vom Diagramm d. aufruf. VIs aus zu bearbeiten (Hinweis), 2-29
- Beschriftungswerkzeug, 2-5, 2-17
- Betrachten-Menü, Hierarchiefenster, 3-17 bis 3-18
- Bibliotheken. *Siehe* VI-Bibliotheken.
- Bibliothekverzeichnis, Pfad angeben für, 7-3
- Biegungen in einer Verbindung, 18-6
- Big endian-Daten, A-12
- Bilder
 - importieren, 8-11
 - Unterschiede beim Portieren von VIs zwischen Plattformen, 29-2
- Bild-Ring, 13-11 bis 13-12
- Bleistift-Werkzeug, Icon-Editor, 3-4
- Blinken
 - Blinkgeschwindigkeit von Objekten des Frontpanels, 7-11
 - Hintergrundfarbe, 7-15
 - Vordergrundfarbe, 7-15
- Blockdiagramm, 17-1 bis 17-12
 - Abbildung, 1-5
 - Abschnitte ausschließen, 4-32
 - Anzeigen, 2-4
 - Arbeitsbereich hinzufügen, 2-26
 - Datenfluß, 1-6
 - Definition, 1-4
 - Drucken, 5-7
 - Einfügen von Objekten, 17-12
 - Ersetzen von Objekten, 17-12
 - Farbenvoreinstellungen, 7-14
 - Funktionen, 17-9 bis 17-11
 - Knoten, 17-1, 17-8 bis 17-12
 - Komponenten des, 17-1
 - Konstanten, 17-2, 17-4 bis 17-8
 - Plazierung und Größenanpassung von Strukturen im, 19-5
 - Strukturen, 1-6, 17-11 bis 17-12
 - Terminals, 1-5, 17-2 bis 17-8
 - Terminals für Bedien- und Anzeigeelemente, 17-2
 - Überblick, 1-5, 17-1 bis 17-12
 - Verbindungen, 1-6
 - Verwendungszweck, 1-4
 - Voreinstellungen, 7-11
- Blockdiagramm. *Siehe* auch Verbinden von Blockdiagrammen.
- Boolesche Datenspeicherformate, A-1
- Boolesche Bedien- und Anzeigeelemente, 10-1 bis 10-6
 - Abbildung der Palette, 10-1
 - Abbildung von Elementen, 10-2
 - Ändern der Beschriftung, 10-3 bis 10-4
 - Anpassen mit importierten Bildern, 10-6

Bedienen, 10-1 bis 10-2
 Beispiel eines booleschen
 Attributknotens, 22-11
 beschriftete Boolesche Elemente, 10-3
 Boolesche Datenbereichsprüfung, 10-4
 Erstellen, 10-1 bis 10-2
 Konfiguration, 10-3 bis 10-6
 Lampen-Simulation, 10-2
 LED-Simulation, 10-2
 Option Boolesch hervorheben, 6-4
 Palette, 8-3, 10-4 bis 10-6
 Popup-Menü-Optionen
 (Abbildung), 10-3
 Schaltverhalten, Konfiguration,
 10-4 bis 10-6
 Latch bis zum Loslassen, 10-6
 Latch während gedrückt, 10-5
 Latch wenn losgelassen, 10-6
 Schaltet bis losgelassen, 10-5
 Schaltet wenn gedrückt, 10-5
 Schaltet wenn losgelassen, 10-5
 Simulation als Drucktaste, 10-2
 Simulation als Kippschalter, 10-2
 Simulation als Schiebeschalter, 10-2
 zwischen TRUE und FALSE hin- und her
 schalten, 10-2
 Breakpoint, Plazieren
 Anzeigen von Breakpoints (Tabelle), 4-28
 Beispiel, 4-29
 Einstellung, 4-27 bis 4-31
 Breakpoint-Werkzeug, 2-5, 4-28
 BridgeVIEW
 Portieren von Anwendungen, 29-15
 zur G-Palette wechseln (Hinweis), 2-1
 Byte-Stream-Datei-RefNum, 12-4

C

Case-Strukturen
 Ausgangsdaten von allen gefordert, 19-16
 Ausschließen von
 Diagrammabschnitten, 4-32
 Auswahlwerte, 19-14 bis 19-18
 außerhalb des Bereichs
 (Hinweis), 19-15
 sortieren, 19-18
 vom verkehrten Typ in rot angezeigt
 (Hinweis), 19-15
 Bearbeitung, 19-20
 Dialogfeld Cases neu anordnen, 19-17
 Hinzufügen von Subdiagrammen,
 19-18 bis 19-22
 Icon für, 19-13
 Löschen von Subdiagrammen, 19-22
 Neuanordnen von Subdiagrammen, 19-21
 Optionen des Popup-Menüs
 (Abbildung), 19-17
 Überblick, 19-13
 Verbinden mit Typ Enum, 19-14
 Verbindungsprobleme, 19-23
 Verschiebungen zwischen
 Subdiagrammen, 19-21
 Werte außerhalb des Bereichs
 (Hinweis), 19-15
 Zweck und Verwendung, 17-11,
 19-15 bis 19-18
 Zyklen in SubVIs vermeiden,
 3-13 bis 3-14
 CFM. *Siehe* Code Fragment Manager
 (Macintosh).
 CGS-Einheiten (Tabelle), 9-29
 Change to Control (Option In Bedienelement
 ändern), 8-3
 Change to Indicator (Option In
 Anzeigeelement ändern), 8-3

CIN (Code Interface Node), 25-3, 26-6, 26-7
Code von anderen Sprachen aufrufen,
25-3 bis 25-14
synchrone Ausführung, 26-7

Cluster, 14-20 bis 14-34
Autom. Skalierung, 14-24
Cluster-Anordnung, A-5 bis A-7
Clusterelemente aufschlüsseln, 14-20
Cluster-Elemente bedienen,
14-23 bis 14-24
Cluster-Elemente ersetzen, 14-34
Datenspeicherformate, A-5 bis A-7
Definition, 14-20
erstellen, 14-21
gewandelte Daten, A-14
Größe ändern, 14-24
Punkte, 14-20
Reihenfolge von Cluster-Elementen,
einstellen, 14-23 bis 14-24
Standardwerte, einstellen, 14-23
Typ-Deskriptoren, A-11
Typendefinition, 24-23
Verbindungsmuster, 14-20
verglichen mit Arrays, 14-20
verschieben, 14-24
Verwendungszweck, 14-20
zerlegen, 14-30 bis 14-34
Funktion Aufschlüsseln, 14-30
Funktion Cluster in Array,
14-33 bis 14-34
Funktion Nach Namen aufschlüsseln,
14-31 bis 14-33
zusammensetzen, 14-25 bis 14-30
Funktion Array in Cluster,
14-29 bis 14-30
Funktion Bündeln, 14-25
Funktion Nach Namen bündeln,
14-26 bis 14-28

Cluster. *Siehe* auch Cluster-Elemente.

Cluster-Elemente
aufschlüsseln, 14-20
ersetzen, 14-34
konfigurieren und verwenden,
14-22 bis 14-24
Reihenfolge festlegen, 14-23

Cluster-Shell, 14-21

Code Fragment Manager (CFM)

Cursor des Graphen, 15-28 bis 15-33
arrayartiges Verhalten, 15-29
Cursor-Palette (Abbildungen),
15-28 bis 15-29
Fadenkreuze, 15-31
Fixierschaltfläche, 15-32
Komponenten, 15-28 bis 15-29
Löschen, 15-30
Namen anzeigen, 15-31
programmatisches Lesen
(Beispiel), 22-14
Punktstile, 15-31
Steuern des Aussehens, 15-30 bis 15-32
Verschieben, 15-30
Zweck und Verwendung, 15-28

Cursor. *Siehe* Cursor des Graphen.

CVI-Funktions-Panel-Konverter.
Siehe LabWindows/ CVI-Funktions-Panel-
Konverter.

D

Darstellung von numerischen Werten
Ändern, 9-4
Konstanten, 17-8

Datei-RefNums, 12-4

Dateiverwaltung
Anordnung von Dateien in
VI-Bibliotheken, 2-33 bis 2-35,
27-1 bis 27-2
Sichern von Dateien, 27-2

- Datenabhängige Ausführung, 1-6
- Datenbubbles während der Ausführung, auswählen, 7-13
- Datenerfassungs-Ausführungssysteme, 26-5, 26-7
- Datenfluß
 - Definition, 1-6
 - verglichen mit Ausführung mittels Steuerungsfluß, 1-6
- Datenfluß-Programmierung
 - Datenpuffer, 28-17 bis 28-19
- Datengetriebene Ausführung. *Siehe* Datenabhängige Ausführung.
- Datenprotokoll-Datei-RefNum, 12-4
- Datenprotokollierung. *Siehe* Frontpanel-Datenprotokollierung.
- Datenspeicherformate
 - Arrays, A-4
 - Boolesche Ausdrücke, A-1
 - Cluster, A-5 bis A-7
 - numerische, A-3
 - Byte-Integer-Zahlen, A-3
 - Double-Zahlen, A-3
 - Extended-Zahlen, A-2
 - Handles, A-5
 - Long-Integer-Zahlen, A-3
 - Single-Zahlen, A-3
 - Word-Integer-Zahlen, A-3
 - Pfade, A-5
 - Strings, A-5
- Datentabellen. *Siehe* Tabellen.
- Datentypen
 - Arrays, 25-6
 - Aufrufen von Funktionen, die nicht-G-Datentypen erwarten, 25-8
 - Beispiele, A-10
 - Datentypen für XY-Graphen
 - Einzelplot-Graph, 15-4
 - Mehrplot-Graph, 15-10 bis 15-11
 - Datentypen von Kurvengraphen
 - Einzelplot-Graph, 15-3
 - Mehrplot-Graph, 15-5 bis 15-9
 - Drucken von Datentypinformationen aufnehmen, 5-7
 - nicht num. (Tabelle), A-9
 - Numerisch, 25-6
 - numerische Skalar-Datentypen (Tabelle), A-8 bis A-9
 - Optionen des Listenfelds, 13-3
 - Speichererwägungen
 - Erzeugen von Daten des richtigen Typs, 28-27 bis 28-28
 - konsistente Datentypen, 28-26 bis 28-28
 - Durchsuchen von Strings (Beispiel), 28-31 bis 28-33
 - Erstellen von Arrays (Beispiel), 28-29 bis 28-30
 - Vermeiden ständiger Größenänderung von Daten, 28-28
 - Speicherung physikalischer Mengen (Hinweis), A-10
 - Strings, 25-7
 - Vermeiden komplizierter Datentypen in Strukturen, 28-36 bis 28-37
 - void, 25-6
- Debugging von VIs, 4-9 bis 4-32
 - ausführbare VIs, 4-14 bis 4-16
 - Erkennen undefinierter Daten, 4-16
 - Schritte zur Lösung von Problemen, 4-14 bis 4-16
 - Warnungen verstehen, 4-16
 - Ausschließen von Diagrammabschnitten, 4-32
 - Deaktivieren von

- EinzelSchrittausführung eines VI,
4-19 bis 4-21
 - Beispiel, 4-19 bis 4-20
 - Kurzbefehl Anhalten, 4-19
 - Kurzbefehl Ausführen, 4-19
 - Kurzbefehl Herausspringen,
4-20, 4-21
 - Kurzbefehl Hineinspringen,
4-20, 4-21
 - Kurzbefehl Überspringen, 4-20, 4-21
 - Verwendung der Schritttasten,
4-19, 4-21
- Erkennen undefinierter Daten,
4-16 bis 4-17
- gestörte VIs
 - anhand von Fehlerlisten auffinden,
4-9 bis 4-10
 - Fehlermeldungen (Tabelle),
4-11 bis 4-14
 - häufige Gründe, 4-10 bis 4-11
 - Korrektur des
VI-Bereichsfehlers, 4-18
 - Wiederherstellen gestörter VIs,
4-9 bis 4-16
- Highlight-Funktion für die Ausführung,
4-22 bis 4-24
- LabVIEW-Funktionen, 4-19 bis 4-31
- Lesen von Aufrufketten, 4-21
- Plazieren von Breakpoints, 4-27 bis 4-30
 - Anzeige von Breakpoints
(Tabelle), 4-28
 - Beispiele, 4-29
 - einstellen, 4-27 bis 4-31
- Probe-Werkzeug, 4-24 bis 4-27
 - Erstellen von Proben, 4-26 bis 4-27
 - Verwendung, 4-24 bis 4-26
- Techniken, 4-14 bis 4-16
- Unterdrücken der Ausführung,
4-30 bis 4-31
 - Ansicht der Hierarchiefenster
während der Unterdrückung, 4-31
 - Erkennen automatischer
Unterdrückung, 4-30
 - Optionen, 4-30
 - Taste Ausführen, 4-31
 - Taste Springe zum Anfang, 4-31
 - Taste Zurück zu Aufrufendem, 4-31
 - Verwendung von
Werkzeugpaletten-Tasten bei der
Unterdrückung, 4-31
- Dezimaltrennzeichen
 - Auswählen, 7-10
 - Dezimalpunkt und -komma, 29-14
- Diagrammanzeigeelemente
 - Attribute
 - Aktiver Plot, 22-10
 - Plotfarbe, 22-10
 - Fragen zu, B-1 bis B-4
 - Intensitätsdiagramm, 15-33 bis 15-38
 - Abbildungen, 15-34 bis 15-35
 - Definieren der Farbuordnung,
15-37 bis 15-38
 - Optionen, 15-35 bis 15-38
 - Popup-Menü (Abbildung), 15-36
 - Intensitätsgraph, 15-38
 - Kurvendiagramm, 15-21 bis 15-27
 - Aktualisierungsmodus,
15-25 bis 15-26
 - Datentypen, 15-22 bis 15-24
 - gestapelte verglichen mit
überlagerten Plots, 15-27
 - Optionen, 15-24 bis 15-27
- Diagramm-Fenster, 1-4
- Diagramm-Identifikator, Anzeigefenster des
Subdiagramms, 19-13

- Dialogfeld Benutzerspezifische Druckeinstellungen, 5-5
 - Abbildung, 5-6
 - Bedienelemente, 5-7
 - Blockdiagramm, 5-7
 - Frontpanel, 5-6
 - Icon und Beschreibung, 5-6
 - Liste der SubVIs, 5-7
 - VI-Hierarchie, 5-7
 - VI-Historie, 5-7
- Dialogfeld Cases neu anordnen, 19-17
- Dialogfeld Debug-Voreinstellungen, 7-13
 - Auto-Probe während Highlight-Option, 7-13
 - Datenbubbles während der Highlight-Funktion anzeigen, 7-13
 - standardmäßig Warnungen im Fehlerfenster anzeigen, 7-13
 - Warnen bei Objekten, die in der Studentenversion nicht verfügbar sind, 7-13
- Dialogfeld Druckeinstellungen anpassen....
Siehe Option Dokumentation drucken.
- Dialogfeld Formatieren
 - Option Achsenstil, 15-15
 - Option Format & Präzision, 15-14 bis 15-17
 - Option Gitteroptionen, 15-15
 - Option Skalierung, 15-15
 - Option Skalierungsfaktoren, 15-15
- Dialogfeld Server: Exportierte VIs, 7-28 bis 7-30
 - Abbildung, 7-28
 - Optionsfeld Zugriff erlauben, 7-28
 - Optionsfeld Zugriff verweigern, 7-28
 - Taste Entfernen, 7-28
 - Taste Hinzufügen, 7-28
- Dialogfeld Server: Konfiguration, 7-23 bis 7-24
 - Abbildung, 7-23
 - Anwendungen und Einstellungen, 7-24
 - Taste Hinzufügen, 7-25
 - VI-Aufrufe, 7-24
 - VI-Methoden und Einstellungen, 7-24
- Dialogfeld Server: TCP/IP-Zugriff, 7-24 bis 7-27
 - Abbildung, 7-24
 - fehlender Zugriff auf den DNS-Server (Hinweis), 7-27
 - Strikte Überprüfung, 7-26
 - Taste Entfernen, 7-25
 - Taste Hinzufügen, 7-25
 - Taste Zugriff erlauben, 7-25
 - Taste Zugriff verweigern, 7-25
 - Zugriffsliste, 7-27
- Dialogfeld Suchen, 3-23 bis 3-29
- Dialogfeld Voreinstellung für Farben, 7-14 bis 7-15
 - Abbildung, 7-14
 - Blinken Hintergrund, 7-15
 - Blinken Vordergrund, 7-15
 - Blockdiagramm, 7-14
 - Formatumwandlungspunkt, 7-14
 - Frontpanel, 7-14
 - Menühintergrund, 7-14
 - Menütext, 7-14
 - Rollbalken, 7-14
 - Standardfarben, 7-15
 - weitere Farben, 7-15
- Dialogfeld Voreinstellungen für Blockdiagramme, 7-11
 - Abbildung, 7-11
 - Anzahl Rückgängig pro VI, 7-12, 7-32
 - Bei Platzierung Namen der SubVIs anzeigen, 7-12
 - Fenstertitel in Funktionspalette verwenden, 7-12
 - Tip-Strips über Terminals anzeigen, 7-12
 - Transparente Namen-Label verwenden, 7-12

- Verbindungshilfen anzeigen, 7-12
- Verbindungspunkte an Kreuzungen anzeigen, 7-11, 7-12
- Dienstprogramm Suchen, 3-29
- Digitale Anzeige, verbergen, 2-15
- Digitale Bedien- und Anzeigeelemente, 9-2 bis 9-11
 - Abbildung, 9-1, 9-2
 - Ändern der Darstellung von numerischen Werten, 9-4
 - Anzeige von Integer-Werten in anderen Radizes, 9-4
 - Bedienwerkzeug, 9-2
 - Bereichsoptionen, 9-5 bis 9-6
 - Bereich der erweiterten Fließkommazahlen (Hinweis), 9-5
 - Dialogfeld Datenbereich, 9-6
 - Fließkommazahlen (Tabelle), 9-5
 - numerische Bereichsprüfung, 9-6 bis 9-8
 - Erhöhen und Verringern, 9-3
 - Format und Präzision digitaler Anzeigen, 9-8 bis 9-11
 - absolute Zeit und/oder Datum, 9-10 bis 9-11
 - Beispiel, 9-9
 - Dialogfeld Format und Präzision, 9-9
 - gültige Werte, 9-2
 - numerische Bereichsprüfung, 9-6 bis 9-8
 - Popup-Menüoptionen, 9-3
 - Überblick, 9-2
 - Zweck und Verwendung, 9-2 bis 9-3
- Dimension hinzufügen, 14-8
- Dokumentation (National Instruments)
 - Gliederung des Handbuchs, xxiii bis xxvii
 - Anhänge, Glossar und Index, xxvii
 - Blockdiagramm-Programmierung, xxv
 - Fortgeschrittene Themen, xxvi
 - Frontpanel-Objekte, xxiv
 - Grundlegende Konzepte der Sprache G, xxiii bis xxiv
 - Konventionen in diesem Handbuch, xxvii bis xxix
- Dokumentation, Drucken. *Siehe* Option und Dialogfeld Dokumentation drucken.
- Dokumenttypen, 16-3
- Double- (DBL) Fließkommazahlen, Auswahl von, 9-4
- dreidimensionale Arrays, 14-11
- Drucken in G, 5-1
 - alternative Methoden, 5-10
 - Bitmap, 7-17
 - Dialogfeld Voreinstellungen (Abbildung), 7-16
 - Drucker-Dithering zulassen, 7-17
 - Einstellung, 5-1 bis 5-2
 - Farbe/Graustufen, 7-17
 - Fragen zu, B-7 bis B-9
 - Option Fenster drucken, 5-1
 - PostScript drucken, 7-17
 - programmatisches Bedienelement
 - Aktivieren, 5-8
 - Steuern des Ausdruckzeitpunkts, 5-8
 - Seitenränder, 7-17
 - Standarddruck, 7-16
 - Voreinstellungen festlegen, 7-16 bis 7-18
- Drucken/Exportieren von Bedienelement- und VI-Beschreibungen in eine RTF- oder HTML-Datei, 5-10 bis 5-14
 - Dialogfeld Dokumentation als HTML-Datei drucken (Abbildung), 5-12
 - Dialogfeld Dokumentation als RTF-Datei drucken (Abbildung), 5-12
 - Dialogfeld Drucken (Abbildung), 5-11
 - GIF-Format, 5-13
 - Beispiele für Bildbenennungsschemata, 5-14

JPEG-Format, 5-13
 resultierende JPEG-Dateien
 (Tabelle), 5-14
 PNG-Format, 5-13
 Druckfensteroption
 Drucken des Fensterinhalts, 5-3
 dynamisches Laden von VIs, B-11

E

Editiermodus
 durch Verbergen der Menüleiste und
 Werkzeugpalette ausgelöste
 Probleme, 6-5
 Effiziente Datenstrukturen. *Siehe* Leistung.
 Eigenschaft Alle VIs im Speicher, 21-10
 Eigenschaften und Methoden
 bearbeiten
 Anwendungen, 21-9
 Anwendungen und VIs, 21-10
 VIs, 21-8
 von Servern, Anwendungen das Lesen
 erlauben, 7-24
 von VIs, Anwendungen das Lesen
 erlauben, 7-24
 Eigenschaftsknoten, 21-4 bis 21-7
 Eindimensionale Arrays, 14-3, 14-10
 Eingabetaste, zum Beenden der
 Texteingabe, 7-10
 Eingabetaste. *Siehe* Enter-Taste.
 Eingangsanschlüsse, 1-5
 Eingangsterminals
 Definition, 17-2
 Einheiten, 9-26 bis 9-34
 abgeleitete Einheiten (Tabelle), 9-28
 Basiseinheiten (Tabelle), 9-28
 CGS-Einheiten (Tabelle), 9-29
 Eingabe von Einheiten, 9-30
 Informationen, 9-27
 polymorphe Einheiten, 9-33

sonstige Einheiten (Tabelle), 9-30
 strikte Typenprüfung, 9-31 bis 9-32
 zusätzliche Einheiten, die mit
 SI-Einheiten verwendet werden
 (Tabelle), 9-29
 Einstellungen SubVI, 6-7
 Einteilungsring, 2-13
 Einzelbilder, in Struktursequenzen, 19-18
 Einzelschrittausführung eines VI,
 4-19 bis 4-21
 Anhalten, 4-19
 Beispiel, 4-19 bis 4-20
 Kurzbefehl Herausspringen, 4-20, 4-21
 Kurzbefehl Hineinspringen, 4-21
 Kurzbefehl Überspringen, 4-20, 4-21
 Verwendung der Schrittasten, 4-19, 4-21
 VIs ausführen, 4-19
 Elemente von Bedienelementen
 Bedienelemente als Elemente,
 24-16 bis 24-17
 gestalterische Elemente, 24-10 bis 24-15
 Hinzufügen zu benutzerspezifischen
 Bedienelementen, 24-17 bis 24-18
 mit mehreren Bildern, 24-13
 mit unabhängigen Bildern,
 24-14 bis 24-15
 Textelemente, 24-15
 Elemente. *Siehe* auch Bedienelemente.
 Elementpalette
 Graphenpalette, 15-1 bis 15-2
 Zugriff auf die Cluster-Shell, 14-21
 Empfohlene Verbindungen, 1-8, 3-9
 <Entf>-Taste
 Löschen von Verbindungen, 18-6
 Enter-Taste (Eingabetaste), 9-2, 14-24
 Entfernen von Objekten. *Siehe* Löschen von
 Objekten.
 Entfernen von Strukturen, ohne Inhalte zu
 löschen, 19-26
 <Entf>-Taste
 Löschen von Objekten, 2-15

- Enum. *Siehe* Aufzählung.
- Enumerated Datentypen, programmatische Änderung, B-17
- Enumerated Type-Bedienelemente
 - Abbildung, 13-12
 - Definition, 13-12
 - Zweck und Verwendung, 13-12 bis 13-13
- Erforderliche Verbindungen, 1-8, 3-9
- Erstellen des Frontpanels, 8-1 bis 8-11
 - Anpassen von Steuerungen in Dialogfeldern, 8-9 bis 8-10
 - Bedienelemente-Palette, 8-1
 - Ersetzen von Bedienelementen, 8-4 bis 8-6
 - importierte Bilder zum Anpassen von Bedienelementen, 8-11
 - Key Navigation, 8-6 bis 8-8
 - Option Panelreihenfolge, 8-8
 - Optionen für Bedien- und Anzeigeelemente, 8-2 bis 8-4
 - Tastenbelegung für Bedienelemente, 8-6 bis 8-8
- Erstellen von SubVIs
 - anhand einer Auswahl erstellen, 3-10 bis 3-14
 - Anzahl der Verbindungen, 3-12
 - Beispiel, 3-14
 - Regeln und Empfehlungen, 3-11 bis 3-14
 - Zyklen vermeiden, 3-13 bis 3-14
 - Attributknoten in Case-Strukturen, 3-13 bis 3-14
 - Attributknoten in Schleifen, 3-12
 - Auffinden durch G, 3-12
 - Frontpanel-Terminal in Case-Struktur, 3-13
 - lokale Variable in Case-Strukturen, 3-13
 - lokale Variable und Frontpanel-Terminals innerhalb von Schleifen, 3-13
 - unlogische Auswahlsätze, 3-13
- Erstellen des Icons, 3-2 bis 3-5
 - Abbruch-Taste, 3-5
 - Icon-Editor (Abbildung), 3-2
 - Icons ausschneiden, kopieren und einfügen (Hinweis), 3-5
 - OK-Taste, 3-5
 - Option Icon bearbeiten, 3-2
 - Pipette, 3-5
 - Schwarzweiß-Icons gegenüber Farb-Icons, 3-3
 - Werkzeug Ausgefülltes Rechteck, 3-4
 - Werkzeug Auswählen, 3-4
 - Werkzeug Bleistift, 3-4
 - Werkzeug Füllbehälter, 3-4
 - Werkzeug Linie, 3-4
 - Werkzeug Pipette, 3-4, 3-5
 - Werkzeug Rechteck, 3-4
 - Werkzeug Text, 3-4
 - Werkzeug
 - Vordergrund/Hintergrund, 3-4
- hierarchischer Aufbau, 3-1
- Popup-Menü Suchen
 - Attributknoten, 3-29
 - globale und lokale Variablen, 3-29
- Suchen von VIs, Objekten und Text, 3-23 bis 3-29
 - Dialogfeld Suchen, 3-23 bis 3-27
 - Eingrenzen des Suchbereichs, 3-27
 - Fenster Suchergebnisse, 3-28
 - Suchen des nächsten und vorherigen Suchobjektes, 3-29
 - Textsuchoptionen, 3-26
 - VIs und andere Objekte, 3-24

- Terminal-Verbindungen, 3-5 bis 3-10
 - Auswählen und Ändern von Terminal-Mustern, 3-6
 - bestätigen, 3-10
 - erforderliche, empfohlene und optionale Verbindungen, 3-9
 - löschen, 3-9
 - weißes Terminal zeigt unvollkommenen Anschluß an (Hinweis), 3-8
 - Zuweisen von Bedien- und Anzeigeelementen, 3-7
- Erstellen von VIs, 2-1 bis 2-31
 - Ausrichten von Objekten, 2-12
 - Auswählen von Objekten, 2-7
 - <Umschalttaste> drücken und klicken, 2-8
 - aus Paletten, 2-1 bis 2-4
 - Mehrfachauswahl mit dem Auswahlrechteck, 2-8
 - Beschriften von Objekten, 2-15 bis 2-24
 - freie Labels, 2-17 bis 2-19
 - Texteigenschaften, 2-19 bis 2-24
 - Blockdiagramm und Frontpanel nebeneinander anordnen (Hinweis), 2-3
 - Duplizieren von Objekten, 2-14
 - Einteilen von Objekten, 2-13
 - Erstellen von Objektbeschreibungen, 2-29
 - Erstellen von VI-Beschreibungen, 2-30
 - Größenanpassung
 - Labels, 2-26
 - von Frontpanel, Blockdiagramm und Arbeitsbereich, 2-25
 - G-Umgebung, 2-1 bis 2-7
 - Menüs, 2-6
 - Popup-Menüs, 2-6
 - von BridgeVIEW zur G-Palette wechseln (Hinweis), 2-1
 - Werkzeugpalette, 2-4 bis 2-6
 - Löschen von Objekten, 2-15
 - Positionieren von Objekten, 2-10 bis 2-12

- Speichern von VIs, 2-31 bis 2-35
 - farbig kennzeichnen, 2-26 bis 2-28
 - individuelle VI-Dateien, 2-31
 - VI-Bibliotheken (.LLB), 2-33
- Ziehen und Ablegen von VIs, Bildern und Text, 2-9
- Erweiterbare Funktionen, 17-10
- Extended- (EXT) Fließkommazahlen
 - Auswahl von, 9-4
 - Bereichsbeschränkung (Hinweis), 9-5
 - Extended-Form in LabVIEW, A-13

F

- Farbenkopierwerkzeug, 2-6
- Farbenvoreinstellungen, 7-14
- Farbenwerkzeug, 2-6, 2-26
- Farbfeld, 9-24 bis 9-25
 - Abbildung, 9-24
 - numerische Bedienelemente und Anzeigen, 9-25
- Farbfeld-Konstanten, 9-24, 17-6
- Farbicons gegenüber Schwarzweiß-Icons, 3-3
- Farbicons, erstellen, 3-3
- Farbige Kennzeichnung von Objekten, 2-26 bis 2-28
 - Anpassen von Farben, 2-27
 - Einschränkungen, 2-26
 - Farben kopieren und übertragen, 2-28
 - transparente Objekte, 2-27
 - Vordergrund- und Hintergrundfarben, 2-27
- Farbpalette, 2-26
 - Option Mehr..., 2-27
- Farbrampe, 9-25 bis 9-26
 - Zweck und Verwendung, 9-25 bis 9-26
- Fehler
 - in Formelknoten entdeckt (Tabelle), 20-10
 - mögliche Fehler im Fehlerlisten-Fenster, 4-9

- Fehlermeldungen und Systemabstürze, Fragen zu, B-4 bis B-6
- Fehler-Ringkonstanten, 17-6
- Fehlersuche. *Siehe* Fragen zu G.
- Fenster Suchergebnisse, 3-28 bis 3-29
- Fensteroptionen
 - <Return> Boolesch hervorheben, 6-4
 - Auf Fenstergröße anpassen, 6-5
 - Automatisch Zentrieren, 6-5
 - Option Dialogfeld, 6-4
 - Protokoll aktivieren/Drucken nach Abschluß, 6-5
 - Run-Time Popup-Menü zulassen (Kontrollkästchen), 6-4
 - Werkzeugpalette anzeigen, 6-5
- Fensterrollwerkzeug, 2-5
- Festlegung von Taskprioritäten, 26-8 bis 26-25
 - ablaufinvariante Ausführung
 - Beispiel eines Speicher-VIs, das nicht auf d. Freigabe von Daten ausgelegt ist, 26-16
 - Beispiel eines VIs mit Wartefunktion, 26-15
 - Überblick, 26-13
 - Abstimmung d. Zugriffs auf globale und lokale Variablen mit externen Ressourcen, 26-18
 - Allgemeine Empfehlungen, 26-24
 - funktionale globale Variablen, 26-19
 - Notification-VIs, 26-23
 - Occurrence-Funktionen, 26-23
 - Priorität durch VI-Einstellungen, 26-9
 - Prioritätsstufe "Subroutine", 26-12
 - Queue-VIs, 26-24
 - Rendezvous-VIs, 26-24
 - Semaphore, 26-23
 - Singlethread-Ausführungssystem, 26-10
 - Thread der Benutzeroberfläche, 26-10
 - Wartefunktionen und Taskprioritäten, 26-8
 - Wettbewerbsbedingungen, 26-18
- Formelknoten
 - Funktionsnamen, 20-5 bis 20-7
- Format & Präzision...
 - graphische Anzeigen
 - Numerisch formatieren, 15-15
 - Format von Digitalanzeigen, Ändern, 9-8 bis 9-11
 - Format-Icon Vollständige Dokumentation, 5-4, 27-14
- Formatierung von Datum
 - absolut
 - Bedienelemente von Graphen, 15-16 bis 15-17
 - digitale Anzeigen, 9-10 bis 9-11
 - Funktion Formatiere Datum-/Zeit-String, 29-15
- Formatumwandlungspunkt, Farbeinstellung, 7-14
- Formelknoten, 20-1 bis 20-10
 - Bakus-Naur Form-Notation, 20-8
 - Definition, 20-1
 - Eingangs- und Ausgangsvariablen, 20-2 bis 20-3
 - Fehler (Tabelle), 20-10
 - Fließkomma-Skalare, 20-4
 - Funktionsnamen (Tabelle), 20-5 bis 20-8
 - Kommentare, 20-1
 - Popup-Menü, 20-2
 - Syntax, 20-1, 20-8 bis 20-9
 - Syntaxfehler, 20-4
- Formelknoten Palette, 20-1
- For-Schleife
 - Auto-Indizierung, 19-7 bis 19-8
 - Icon für, 19-3
 - Nullmalige Ausführung, 19-9
 - Plazieren von Objekten in Strukturen, 19-4
 - Schieberegister, 19-10 bis 19-13

- Terminals in Schleifen, 19-6 bis 19-7
- Vermeiden unnötiger Berechnungen, 28-12 bis 28-14
- Zweck und Verwendung, 17-11, 19-4
- Fragen zu G
 - Diagramme und Graphen, B-1 bis B-4
 - Drucken, B-7 bis B-9
 - Fehlermeldungen und Systemabstürze, B-4 bis B-6
 - Plattformthemen und Kompatibilitäten, B-6 bis B-7
 - verschiedene Fragen, B-10 bis B-17
- Freie Form der Datenspeicherung, A-12
- Freie Labels, 2-15
- Frontpanel
 - Abbildung, 1-3
 - Arbeitsbereich hinzufügen, 2-26
 - Bedienelemente-Palette (Abbildung), 1-4
 - Datenprotokollierung, 4-5 bis 4-6
 - Definition, 1-3
 - Drucken (Option Frontpanel), 5-6
 - erstellen, 8-1 bis 8-11
 - Anpassen von Steuerungen in Dialogfeldern, 8-9
 - Bedienelemente-Palette, 8-1
 - Ersetzen von Bedienelementen, 8-4 bis 8-6
 - importierte Bilder zum Anpassen von Bedienelementen, 8-11
 - Option Panelreihenfolge, 8-8
 - Optionen für Bedien- und Anzeigeelemente, 8-2 bis 8-4, 8-11
 - Tastenbelegung für Bedienelemente, 8-6 bis 8-8
 - Farbenvoreinstellungen, 7-14
 - Grafiken importieren, 8-10
 - Hilfeinformationen, 1-8
 - Key Navigation (Dialogfeld), 8-6 bis 8-8
 - Panelreihenfolge, 8-8 bis 8-9
 - Popup-Panels erstellen. *Siehe* Popup-Panels, erstellen.
 - Regeln zur Speicherauslastung, 28-22 bis 28-23
 - Steuerungen in Dialogfeldern, 8-9 bis 8-10
 - Terminals
 - automatisch erstellt, 2-3
 - Zyklen in SubVIs vermeiden, 3-13
 - Überblick, 1-3
 - Verwenden von Änderungen von benutzerspezifischen Bedienelementen, 24-4 bis 24-6
 - zum Blockdiagramm wechseln, 1-4
- Frontpanel-Datenprotokollierung
 - automatische Datenprotokollierung aktivieren/deaktivieren, 6-6
 - Datensätze anzeigen, 4-6
 - Datensätze zur Löschung markieren, 4-6
 - Kurvenformdiagramm (Hinweis), 4-6
 - Log-Dateibindung ändern..., 4-5 bis 4-6
 - Löschen markierter Datensätze, 4-6
 - Menü Datenprotokollierung, 4-5 bis 4-7
 - programmatische Datenabfrage, 4-7 bis 4-9
 - Halo-Terminals für den Zugriff auf Datenbanken, 4-7 bis 4-8
 - I/O-Funktionen verwenden, 4-8
 - Zugreifen auf Datenbanken, 4-7 bis 4-8
- Füllbehälter-Werkzeug, Icon-Editor, 3-4
- Fülloptionen, 9-20
- Funktion Array erstellen, 17-10
- Funktion Array in Cluster, 14-29 bis 14-30
- Funktion Aufruf von Bibliotheken, 25-3 bis 25-7
 - Aufruf von Funktionen, die andere Datentypen erwarten, 25-8
 - Aufrufkonventionen (Windows), 25-5

- Dialogfeld Aufruf externer Bibliotheken, 25-4
- Option Konfigurieren, 25-4
- Parameterlisten erstellen, 25-5 bis 25-8
 - An Typ anpassen, 25-7
 - Datentyp Array, 25-6
 - Datentyp String, 25-7
 - Datentyp Void, 25-6
 - numerische Datentypen, 25-6
- Überblick, 25-2
- Funktion Aufschlüsseln, 14-30
- Funktion Bündeln, 14-25, 24-23
- Funktion Cluster in Array, 14-33 bis 14-34
- Funktion Nach Namen aufschlüsseln
 - Cluster-Elemente zerlegen, 14-31 bis 14-33
 - Zugriff auf Cluster-Elemente, 24-23
- Funktion Nach Namen bündeln
 - auf Clusterelemente zugreifen, 24-23
 - Cluster zusammensetzen, 14-26 bis 14-28
 - Zugriff auf Cluster-Elemente, 24-23
 - Zusammensetzen von Clustern, 14-24 bis 14-25
- Funktionen, 17-9 bis 17-11
 - Definition, 17-9
 - erweiterbare Funktionen, 17-10
 - Label anzeigen, 17-9
 - Standard-Label, 2-18
 - Überblick, 17-9
- Funktionenpalette
 - Abbildung, 17-9
 - anpassen. *Siehe* Bedien- und Funktionenpalette, anpassen.
 - Farbfeld-Konstante, 9-24
 - Kommunikation, 5-1
 - Objekte auswählen, 2-1 bis 2-4
 - Palette Erweitert, 25-3
 - Popup-Palette (Hinweis), 2-2
 - temporäre Kopie (Hinweis), 2-2

- Funktionsicon Hinzufügen, 1-5
- Funktionsicon Index-Array, 14-18
- Funktionsicon Subtrahieren, 1-5
- Funktionslabels, 2-19

G

- G konfigurieren. *Siehe* Dialogfeld Voreinstellungen.
- Gebundene Labels, 2-15
- Geräte-RefNum, 12-4
- Gestalt. Elemente, 24-10 bis 24-15
 - Definition, 8-2, 24-10, 24-14
 - Hinzufügen zu benutzerspez. Bedienelementen, 24-17 bis 24-18
 - Kopieren von LabVIEW-Elementen oder Dekorationen, 24-11
 - mehrere Bilder, 24-13
 - Optionen
 - Bild importieren, 24-11
 - Importiere mit gleicher Größe, 24-12
 - Kopieren in die Zwischenablage, 24-11
 - Originalgröße, 24-12
 - Umkehren, 24-12
 - Popup-Menü, 24-10
 - unabhängige Bilder, 24-14 bis 24-15
- Gestapelte Plots, 15-27
- Gestörte VIs
 - Auffinden von Fehlern, 4-9 bis 4-10
 - Fehlerbehebung, 4-11 bis 4-14
 - Fehlermeldungen (Tabelle), 4-11 bis 4-14
 - häufige Gründe für, 4-10 bis 4-11
 - Korrigieren von VI-Bereichsfehlern, 4-18
- Gewandelte Daten
 - Arrays, A-14
 - Cluster, A-14
 - Handles, A-13
 - Pfade, A-13

- Skalare, A-13
- Strings, A-13
- Überblick, A-12
- Gitteroptionen, 15-15
- Globale Attribute suchen, lokale Attribute, und Attributknoten, 3-29
- Globale Variablen
 - Beispiel, 23-2
 - Definition, 23-1
 - Erstellen von, 23-2
 - in VIs plazieren, 23-2
 - ins Hierarchiefenster aufnehmen, 3-18
 - Speichererwägungen, 28-24
 - wählen, 23-2
 - Zweck und Verwendung, 23-1
- Grafiken importieren., 8-10, 8-11
- Grafiken, importieren.
 - Siehe* auch Bilder importieren.
- Graphenanzeigeelemente
 - Cursor des Graphen, 15-28 bis 15-33
 - Definition, 15-1
 - Erstellen von Mehrplot-Graphen, 15-5 bis 15-11
 - Datentypen für XY-Graphen, 15-10 bis 15-11
 - Datentypen von Kurvengraphen, 15-5 bis 15-9
- Fragen zu, B-1 bis B-4
- Graphenoptionen, 15-11 bis 15-21
 - Abbildung, 15-12
 - Legendenoption, 15-19 bis 15-21
 - Popup-Menü des Graphen, 15-12
 - Scroll- und Zoomoptionen, 15-17 bis 15-19
 - Skalierungsoptionen, 15-12 bis 15-17
- Intensitätsgraph, 15-36 bis 15-38, 15-38
 - Datentyp, 15-38
 - Optionen, 15-38

- Kurven- und XY-Graphen, 15-2 bis 15-4, 16-2 bis 16-5
 - Abbildung, 15-2
 - Datentypen für XY-Graphen, 15-4
 - Datentypen von Kurvengraphen, 15-3
 - Erstellen von Einzelplot-Graphen, 15-3 bis 15-4
- Graphenpalette, xxv, 15-1 bis 15-33
- Größenänderungswerkzeug für Arrays, 14-10
- Größenanpassung
 - Arbeitsbereich hinzufügen, 2-26
 - Arrays, 14-14
 - Cluster, 14-24
 - Labels, 2-26
 - Objekte, 2-25 bis 2-26
 - Spalten, 11-7
 - Tabelle, 11-7
 - Zeile, 11-7
- Größenanpassungswerkzeug, 2-25
- Größenanpassungs-Ziehpunkte, 2-25
- Grundattribute. *Siehe* Attributknoten.

H

- Halo-Anschlüsse, für den Zugriff auf VIs, 4-7
- Handles
 - Datenspeicherformate, A-5
 - gewandelte Daten, A-13
- Häufig gestellt Fragen zu G. *Siehe* Fragen zu G.
- Hierarchie der VIs
 - anzeigen, 3-15 bis 3-21
 - Bedeutung, 3-1
 - Drucken, 5-7
- Hierarchiefenster, 3-15 bis 3-22
 - Funktionen, 3-15
 - Mausklicksequenzen bei Hierarchieknoten, 3-21
 - Mechanismus "Hierarchieknoten suchen", 3-21

- öffnen, 3-15 bis 3-16
- Optionen im Menü "Betrachten",
3-17 bis 3-18
- Popup-Menü von Hierarchieknoten,
3-19 bis 3-20
- Tasten der
Hierarchie-Werkzeugleiste, 3-18
- Verbindungen von SubVIs zu aufrufenden
VIs, 3-16
- Hierarchieknoten suchen, 3-22
- Highlight-Funktion
 - Daten-Bubbles während der
Highlight-Funktion, auswählen, 7-13
 - Debugging- Techniken, 4-22 bis 4-24
- Hilfe
 - anzeigen, 1-7
 - Erstellen Ihrer eigenen, 5-15
 - fixieren, Hilfemenü, 1-8
- Hilfefenster
 - Attributknoten, 1-10
 - Hilfe zum Verbinden, 18-5
 - Zweck und Verwendung, 1-7 bis 1-10
- Hilfemenü
 - Einfache Hilfe, 1-8
 - Hilfe anzeigen, 1-7
 - Hilfe fixieren, 1-8
 - Online-Referenz, 1-10
- Hintergrundfarben, 2-26
- Historie der VI-Entwicklung
 - Drucken, 5-7
 - Drucken der Historie-Informationen,
27-14
 - Historie-Fenster, 27-10 bis 27-14
 - Revisionsnummern, 27-12 bis 27-13
 - Rücksetzen der
Historie-Informationen, 27-13
 - Verwandte Dialogfeldoptionen für
VI-Einstellung und
Voreinstellungen, 27-14
- Historie-Voreinstellungen, 7-18 bis 7-20

Horizontal kippen, 3-7

I

- Icon-Editor, 3-2
 - Abbildung, 3-2
 - Abbrechen-Taste, 3-5
 - OK-Taste, 3-5
 - verfügbare Werkzeuge, 3-2 bis 3-5
- Icons
 - Definition, 1-6, 3-2
 - Drucken, 5-6
 - Drucken (Icon und Beschreibung), 5-6
 - Erstellen, 3-2 bis 3-5
 - für Bedienelemente, 24-5
 - von Farbicons, 3-4
 - von Werkzeugicons, 3-4
 - für Bedienelemente erstellen, 24-5
 - Standardicon (Abbildung), 1-7
 - zum Erstellen von SubVIs
 - Abbrechen-Taste, 3-5
 - Ausgefülltes Rechteck-Werkzeug,
3-4
 - Auswählen-Werkzeug, 3-4
 - Bleistift-Werkzeug, 3-4
 - die Icons Ausschneiden, Kopieren
und Einfügen (Hinweis), 3-3
 - Füllbehälter-Werkzeug, 3-4
 - Icon-Editor (Abbildung), 3-2
 - Linie-Werkzeug, 3-4
 - OK-Taste, 3-5
 - Option Icon bearbeiten, 3-2
 - Pipette-Werkzeug, 3-4
 - Rechteck-Werkzeug, 3-4
 - Text-Werkzeug, 3-4
 - Überblick, 1-6
 - Vordergrund/Hintergrund-Werkzeug,
3-4
 - Zweck, 1-6

- Indexanzeige für Arrays
 - Array in Einzelelement- oder Tabellenform anzeigen, 14-10 bis 14-11
 - Array-Shell, 14-6
 - interpretieren, 14-9
 - Popup-Menü, 14-7
 - Indexarrays, 14-13
 - Indizes für Arrays, 14-2 bis 14-5
 - Inf-Fließkommawert, 4-16, 9-2
 - Info-Labview (User Group), B-10
 - Informationsfeld Fehlerliste, 4-9
 - Inhalt einer VI-Bibliothek löschen, 2-35
 - Integer-Werte
 - Änderung der Darstellung von, 9-4
 - Anzeige von Integer-Werten in anderen Radizes, 9-4
 - Intensitätsdiagramm, 15-33 bis 15-38
 - Abbildungen, 15-34 bis 15-35
 - Optionen, 15-35 bis 15-38
 - Datenhistorie, 15-36
 - Definieren der Farbzuoordnung, 15-37 bis 15-38
 - Popup-Menü, 15-36
 - Intensitätsgraph, 15-38
 - Datentyp, 15-38
 - Optionen, 15-38
 - Interrupts
 - Behandlung, B-13
 - Iterationsterminal, 19-3
- K**
- Kein Pfad-Symbol, 12-2
 - Key Navigation... (Option), 8-6 bis 8-8
 - Dialogfeld (Abbildung), 8-7
 - Zweck und Anwendung, 8-6
 - Knoten
 - Datenfluß, 1-6
 - Definition, 1-5, 17-1
 - Funktionen, 17-9 bis 17-11
 - Strukturen, 17-11
 - Überblick, 17-8
 - Knoten Aufruf externer Bibliotheken, 25-3 bis 25-8
 - Aufrufen von Funktionen, die andere Datentypen erwarten, 25-8
 - Dialogfeld Aufruf externer Bibliotheken, 25-4
 - Erstellen von Parameterlisten, 25-5 bis 25-7
 - Datentyp Array, 25-6
 - Datentyp Numerisch, 25-6
 - Datentyp String, 25-7
 - Datentyp Void, 25-6
 - Überblick, 25-2
 - Kommentare
 - Abfrage auf, 6-6, 7-19
 - mit dem Editor erstellte Kommentare abspeichern, 7-19
 - Komplexe Double- (CDB) Fließkommazahlen, 9-4
 - Komplexe Extended- (CXT) Fließkommazahlen, 9-4
 - Komplexe Single- (CSG) Fließkommazahlen, 9-4
 - Konfigurieren von G. *Siehe* Voreinstellungen.
 - Konstanten, 17-4 bis 17-8
 - Arraykonstante, 17-4
 - automatisch erstellte K., 17-13
 - benutzerdefinierte
 - auf Paletten verfügbare K., 17-4
 - Aufzählungskonstanten, 17-6
 - erhöhen und verringern, 17-5
 - erzeugen, 17-4
 - Farbfeld-Konstanten, 17-6
 - Fehler-Ringkonstanten, 17-6
 - Listensymbol-Ringkonstante, 17-6
 - numerische K. (Abbildung), 17-5
 - numerische Konstanten, 17-6
 - Pfadkonstanten, 17-7
 - Ring-Konstanten, 17-6

- String-Konstante, Popup-Menü (Abbildung), 17-5
- String-Konstanten, 17-5
- Clusterkonstante, 17-4
- Darstellung, 17-8
- Definition, 17-4
- Farbfeld-Konstanten, 9-24, 17-6
- Größenanpassung, 17-5
- Labels der automatisch erstellten Konstanten, 7-23
- Popup-Menüs, 17-5
- universell
 - Definition, 17-4
 - numerische Konstanten, 17-8
 - String-Konstanten, 17-8
- Konvertierungsfunktionen LabVIEW/CVI.
Siehe Funktionspanel von LabVIEW/CVI-Konverter.
- Kooperatives Multitasking, 26-1 bis 26-2
- Kooperatives Multithreading, 26-2
- Kopfzeilen, Drucken von
 - Option Kopfzeile drucken, 5-5
- Kreuzung (von Verbindungen), 18-6
- Kurven- und XY-Graphen, 15-2 bis 15-21, 16-2
 - Abbildung, 15-2
 - Definition, 15-2
 - Erstellen von Einzelplot-Graphen, 15-3 bis 15-4
 - Datentypen für XY-Graphen, 15-4
 - Datentypen von Kurvengraphen, 15-3
 - Erstellen von Mehrplot-Graphen, 15-5 bis 15-11
 - Datentypen für XY-Graphen, 15-10 bis 15-11
 - Datentypen von Kurvengraphen, 15-5 bis 15-9

- Graphenoptionen, 15-11 bis 15-21
 - Abbildung, 15-12
 - Legendenoption, 15-19
 - Skalierungsoptionen, 15-12 bis 15-17
- Kurvendiagramm, 15-21 bis 15-27
- Cursor nicht unterstützt, 15-24
- Datentypen, 15-22 bis 15-24
- Optionen, 15-24 bis 15-27
 - Aktualisierungsmodus, 15-25 bis 15-26
 - Aktualisierungsmodus für Oszilloskopdiagramm, 15-26
 - Aktualisierungsmodus für Streifendiagramm, 15-26
 - Aktualisierungsmodus für Sweepdiagramm, 15-26
 - gestapelte verglichen mit überlagerten Plots, 15-27

L

- Labels, 2-15 bis 2-24
 - automatische Größenanpassung, 2-26
 - Boolsche Bedien- und Anzeigeelemente
 - Beschriftung, 10-3 bis 10-4
 - Definition, 2-15
 - erstellen, 2-17 bis 2-19
 - freie Labels, 2-17
 - Funktionslabels (Hinweis), 2-19
 - gebundene Labels, 2-15
 - Größenanpassung, 2-26
 - numerische Skalen, 9-18 bis 9-20
 - Text kopieren, 2-18
 - Textattribute ändern, 2-19 bis 2-24
 - Unterschiede beim Portieren von VIs
 - zwischen Plattformen, 29-2
 - verborgene Labels anzeigen, 2-19
- LabVIEW-Entwicklungssystem
 - Hilfe erhalten, 1-7 bis 1-10
 - Menüs, 2-1 bis 2-2

- Online-Referenz, 1-10
 - Erstellen Ihrer eigenen, 5-15
- Überblick, 1-1 bis 1-7
- VI's speichern, 2-31 bis 2-35
- Werkzeuge, 2-4 bis 2-6
- LabWindows/CVI-Funktions-Panel-
 - Konverter, 25-9 bis 25-14
- Leere Arrays, 14-12
- Legendenoption für Graphenanzeigeelemente, 15-19 bis 15-21
- Leistung, 28-22
 - Beschleunigung von VI's, 28-7 bis 28-14
 - Bildschirmanzeige, 28-8
 - Eingang/Ausgang, 28-7 bis 28-8
 - Overhead für SubVI's, 28-11
 - parallele Diagramme, 28-10
 - unnötige Berechnungen in Schleifen, 28-12 bis 28-14
 - effiziente Datenstrukturen, 28-34 bis 28-42
 - alternative Implementierung, 28-39 bis 28-41
 - globale Tabelle gemischter Datentypen, 28-38 bis 28-41
 - offensichtliche Implementierung, 28-38
 - statische globale Tabelle mit Strings, 28-41
 - Vermeiden komplizierter Datentypen, 28-36 bis 28-37
 - Speicherauslastung, 28-14 bis 28-33
 - allgemeine Regeln zur Verbesserung, 28-21 bis 28-22
 - Datenfluß-Programmierung und Datenpuffer, 28-17 bis 28-19
 - Freigabe durch LabVIEW, 28-25 bis 28-26
 - Frontpanels, 28-23
 - globale Variablen, 28-24
 - grundlegende Konzepte, 28-16 bis 28-19
 - lokale Variablen, 28-24
 - Überblick, 28-14
 - Überwachung, 28-20
 - VI-Komponenten, 28-16 bis 28-17
 - virtueller Speicher, 28-15
 - Wiederverwendung des Datenspeichers durch SubVI's, 28-24
 - Wiederverwendung von Eingangspuffer durch Ausgang, 28-26
 - VI-Leistungsprofile, 28-1 bis 28-7
 - Leistungserwägungen für
 - Bildschirmanzeige, 28-8
 - Leistungsprofile, 28-1 bis 28-7
 - Linearer Skalenabstand, 9-15
 - Linie-Werkzeug, Icon-Editor, 3-4
 - Listenfeld-Bedienelemente, 13-2 bis 13-8
 - Einzelauswahl-Listenfeld, 13-2
 - Pop-up-Menü, 13-4 bis 13-6
 - Auswahlmodus, 13-5 bis 13-6
 - Option Anzeigen, 13-4 bis 13-5
 - Logarithmischer Skalenabstand, 9-15
 - lokale Sequenz-Variablen
 - Hinzufügen, 19-19
 - Zuweisen mehrerer Werte zu, 19-22
- Lokale Variablen, 23-4 bis 23-5
 - Erstellen von, 23-4
 - unfähig zur Wiederverwendung des Datenspeichers, 28-24
- Löschen von Objekten
 - Subdiagramme, 19-22
- Löschen. *Siehe* auch Entfernen.
 - Cursor des Graphen, 15-28 bis 15-33
 - Datensätze, 4-6
 - Objekte, 2-15
 - SubdiagrammeLöschen von Objekten
 - Subdiagramme, 19-22
 - Terminal-Anschluß, 19-20
 - Verbindungen, 18-6 bis 18-8

von Strukturen, nicht aber deren
Inhalten, 2-15

von Strukturen, ohne deren
Elemente/Inhalte, 19-26

<Löschtaste>

Objekte löschen, 2-15

Verbindungen löschen, 18-6

M

Maus

Deaktivieren von Maus-Interrupts, B-17

rechte Taste für Zugriff auf
Popup-Menüs, 1-4

Symbol für das Verbinden von
Blockdiagrammen, 18-1

Verschiebungen zwischen
Subdiagrammen, 19-21

Mehrere Entwickler von VIs. *Siehe*
Anwendungen, Verwalten von.

Mehrplot-Graphen

Siehe Kurven- und XY-Graphen.

Menü Ausführen

Option Anpassungsmodus, 24-6

Option Bearbeitungsmodus, 24-6

Menü Bearbeiten, 7-20

Ganz nach hinten, 2-12

Ganz nach vorne, 2-12

Menü Bearbeiten, 6-9

Option Anwendername, 7-20

Option Ausschneiden, 2-14

Option Bedienelement bearbeiten...,
24-2, 24-16

Option Einfügen, 2-14

Option Eins nach hinten, 2-12

Option Kopieren, Menü Bearbeiten, 2-14

Option Löschen, 2-15

Option Palettensatz auswählen, 7-35

Option Panelreihenfolge, 8-8

Option Rückgängig machen, 7-32

Option Ungültige Verbindungen
entfernen, 4-9, 18-9, 18-12

Option Ungültige Verbindungen
entfernen (Hinweis), 18-6

Option Voreinstellungen..., 7-1

Option Wiederherstellen, 7-32

Options Eins nach vorne, 2-12

Paletten Bedienelemente und
Funktionen, 7-33

Menü Bedienen

Befehle zum Ausführen von VIs, 4-4

Menü Datei

Kopie speichern unter..., 2-31

Mit Optionen speichern..., 2-31

Option Änderungen verwenden, 24-4

Option Bild importieren, 8-11

Option Seite einrichten, 5-2

Option Umkehren..., 24-12

Option Zurückkehren..., 2-32

Speichern, 2-31, 24-21

Speichern unter..., 2-31, 24-5

VI-Bibliothek bearbeiten..., 2-35

Menü Dateioperationen

Aktualisierungsmodus, 15-25

Aktuellen Wert als Standard, 14-13, 14-23

Auswahl beenden, 14-15, 14-16, 15-30

Auswahl beginnen, 14-15, 14-16, 15-30

Autom. Skalierung X, 15-12, 15-17

Autom. Skalierung Y, 15-12, 15-17

Daten ausschneiden, 14-13

Graphik glätten, 15-13

Leeres Array, 14-13

Letztes Element anzeigen, 14-14

Rücksetzen auf Standard, 14-13, 14-23

Menü Datenoperationen

Auswahl anzeigen, 11-10

Beschreibung, 2-29

Daten ausschneiden, 11-9

Daten einfügen, 11-9

Daten kopieren, 11-9

- Optionen des Popup-Menüs von Bedienelementen, 4-2 bis 4-4
- Optionen des Popup-Menüs von Objekten, 4-2 bis 4-4
- Popup-Menüs von Bedien- und Anzeigeelementen des Frontpanels, 8-3
- Menü Datenprotokollierung, 4-5 bis 4-7
 - Abbildung, 4-5
 - Log-Dateibindungen, 4-5
 - Option Löschen, 4-6
- Menü Fenster
 - Fenster Das Elemente-Fenster anzeigen, 24-8
- Menü Projekt
 - ActiveX-Controls importieren, 16-5
 - aktivieren/deaktivieren, 6-5
 - SubVIs des VIs, 4-16
 - Suchen, 3-23
 - Suchergebnisse, 3-29
 - Ungeöffnete SubVIs, 4-16
 - VI-Hierarchie anzeigen, 3-15
 - VI-Strings exportieren, 29-7
 - VI-Strings importieren, 29-7
 - Weitersuchen, 3-29
- Menühintergrund, 7-14
- Menüleiste
 - Definition, 2-6
 - Farbenvoreinstellungen, 7-14
 - Probleme, die durch eine verborgene Menüleiste ausgelöst werden (Hinweis), 6-5
- Menüs für LabVIEW, 2-1 bis 2-2
 - Menüleiste, 2-6
 - Popup-Menüs, 2-1 bis 2-2, 2-6 bis 2-7
 - Pulldown-Menüs, 2-6
- <Meta-M> Zugriffstaste für Bearbeiten/Ausführen (Hinweis), 6-5
- Modulare Programmierung, 1-2

- Multitasking, 26-1 bis 26-7
 - ablaufinvariante Ausführbarkeit
 - Beispiele, 26-15 bis 26-17
 - kooperatives, 26-1 bis 26-2
 - preemptives, 26-1 bis 26-2
 - Schaltung von Tasks, 26-3 bis 26-4
 - synchrone/blockierende Knoten, 26-7
 - Wartefunktionen zur Festlegung von Taskprioritäten, 26-8
- Multithread-Ausführung. *Siehe* auch Singlethread-Ausführung.
 - allgemeine Empfehlungen zur Verwendung, 26-24
 - G-Ausführungssystem, 26-3
 - grundlegendes Ausführungssystem, 26-4
 - HP-UX und Solaris 1, 26-3
 - kooperatives, 26-2
 - Macintosh und Windows 3.1, 26-3
 - mehrere Ausführungssysteme, 26-5 bis 26-7
 - preemptives, 26-2
 - Überblick, 26-2
 - Windows95/NT, Solaris 2 und PowerMAX, 26-3
- Mutexe (Semaphore). *Siehe* Semaphore.

N

- NaN (keine Zahl)
 - Behandlung in digitalen Bedien- und Anzeigeelementen, 9-2
 - undefinierte Daten und, 4-17
- Netzwerkverbindung-RefNum, 12-4
- Neu anordnen
 - Objekte, 2-11 bis 2-12
 - nach hinten, 2-12
 - Nach vorne, 2-11
 - Subdiagramme, 2-11 bis 2-12

- Nicht anzeigbare Zeichen, Einfügen mit umgekehrten Schrägstrich-(\) Codes, 11-2, 11-4 bis 11-6
 - Beispiele, 11-4 bis 11-6
 - Tabelle mit Codes, 11-4
- Nicht darstellbare Zeichen. *Siehe* Nicht anzeigbare Zeichen.
- Nicht num. Datentypen (Tabelle), A-9
- Notation für Digitalanzeigen, 9-8
- Notifier-RefNum, 12-5
- Numerische Bedien- und Anzeigeelemente
 - Bedien- und Anzeigeelemente mit Schieberegler, 9-12 bis 9-21
 - Abbildung, 9-12
 - ausgefüllte Schieberegler, 9-20 bis 9-21
 - Popup-Menüoptionen (Abbildung), 9-14
 - Schieberegler mit Mehrfachwerten, 9-20 bis 9-21
 - Schieberegler-Skala, 9-14 bis 9-20
 - Zweck und Verwendung, 9-12 bis 9-14
 - digitale Bedien- und Anzeigeelemente, 9-2 bis 9-11
 - Abbildung, 9-2
 - Ändern der Darstellung von numerischen Werten, 9-4
 - Ändern von Format und Präzision, 9-8 bis 9-11
 - Anzeige von Integer-Werten in anderen Radizes, 9-4
 - Einstellung von Bereichsoptionen, 9-5 bis 9-8
 - numerische Bereichsprüfung, 9-6 bis 9-8
 - Popup-Menüoptionen, 9-3
 - Überblick, 9-2 bis 9-3
 - Einheiten, 9-26 bis 9-34
 - abgeleitete Einheiten (Tabelle), 9-28
 - Basiseinheiten (Tabelle), 9-28
 - CGS-Einheiten (Tabelle), 9-29
 - Eingabe von Einheiten, 9-30
 - polymorphe Einheiten, 9-33
 - Sonstige Einheiten (Tabelle), 9-30
 - strikte Typenprüfung, 9-31 bis 9-32
 - zusätzliche Einheiten, die mit SI-Einheiten verwendet werden (Tabelle), 9-29
- Farbfeld, 9-24 bis 9-25
- Farbrampe
 - Zweck und Verwendung, 9-25 bis 9-26
- Palette (Abbildung), 2-2, 9-1
- runde Numerik
 - Zweck und Verwendung, 9-22 bis 9-24
- Numerische Bedien- und Anzeigeelemente mit Schieberegler, 9-12 bis 9-21
 - Abbildung, 9-12
 - ausgefüllte Schieberegler, 9-20 bis 9-21
 - Popup-Menüoptionen (Abbildung), 9-14
 - Schieberegler mit Mehrfachwerten, 9-20 bis 9-21
 - Schieberegler-Skala, 9-14 bis 9-20
 - Ändern von Skalagrenzen, 9-15
 - linearer Skalenabstand, 9-15
 - Popup-Menü Skala, 9-14
 - Text-Labels für Skalen, 9-18 bis 9-20
 - Zweck und Verwendung, 9-12 bis 9-14
- numerische Bereichsprüfung
 - Option Format umwandeln, 9-6
 - Option Ignorieren, 9-6
 - Option Unterdrücken, 9-7
- numerische Datenspeicherformate
 - Byte-Integer-Zahl, A-3
 - Double, A-3
 - Extended, A-2
 - Long-Integer-Wert, A-3
 - Single, A-3
 - Word-Integer-Wert, A-3
- numerische Datentypen, 25-6

numerische Konstanten, 17-6
 universell, 17-8
 Numerische Option,
 Bedienelemente-Palette, 9-1
 Numerische Palette, 9-1

O

Objekte

ausrichten, 2-12 bis 2-13
 auswählen, 2-7 bis 2-9
 duplizieren, 2-14
 einteilen, 2-13
 erstellen, 2-4 bis 2-6
 Größenanpassung, 2-25 bis 2-26
 Labels, 2-15 bis 2-24
 erstellen, 2-17 bis 2-19
 Textattribute ändern,
 2-19 bis 2-24
 löschen, 2-15
 Popup-Menüs für, 2-1 bis 2-2, 2-6 bis 2-7
 verschieben
 Eins nach hinten, 2-12
 Ganz nach hinten, 2-12
 Ganz nach vorne, 2-11, 2-12
 mit dem Positionierwerkzeug, 2-10
 Objekte ausrichten, 2-12
 Objekte auswählen, 2-7
 Objekte duplizieren, 2-14
 Objekte klonen, 2-14
 Objekte kopieren, 2-14
 klonen, 2-14
 Verfahren, 2-14
 von anderen Anwendungen, 2-14
 Ziehen und Ablegen, 2-9 bis 2-10
 zwischen VIs, 2-14
 Objekte kopieren (duplizieren), 2-14
 Objekte löschen
 Methoden, 2-15

Verbindungen, 3-9 bis 3-10
 Objekte verschieben
 Arrays, 14-14
 Cluster, 14-24
 Eins nach hinten, 2-12
 Ganz nach hinten, 2-12
 Ganz nach vorne, 2-11, 2-12
 Methoden zum Verschieben,
 2-10 bis 2-11
 mit dem Positionierwerkzeug, 2-10
 Occurrence-RefNum, 12-4
 Online-Referenz, 1-10
 Erstellen Ihrer eigenen, 5-15
 Option <Return> Boolesch hervorheben, 6-4
 Option Ablaufinvariante Ausführung, 6-3
 Option Abstand zwischen Elementen
 einfügen, 14-15
 Option Achsenstil, 15-15
 Option Aktualisierung von Type Def., 24-22
 Option Aktualisierungsmodus, 15-25
 Option Alle Aufrufenden anzeigen, 3-20
 Option Alle Elemente, 22-5
 Option Alle entfernen, 19-12
 Option Alle SubVIs anzeigen, 3-20
 Option Alle VIs anzeigen, 3-18
 Option Allgemeine Plots, 15-20
 Option Am Punkt ausrichten, 15-33
 Option An Quelle angepaßt, 17-7
 Option Änderungen verwenden, 24-4
 Option Angepaßt, 15-17
 Option Anpassungsmodus, 24-6
 Option Anschluß anzeigen, 1-7
 Option Anwendername, 7-20
 Option Anwendungsverteilung, 27-7
 Option Anzeige indexieren, 14-11
 Option Anzeigeelement erzeugen, 17-13
 Option Anzeigen, Popup-Menü des
 Listenfelds, 13-4 bis 13-5
 Option AppleEvent, Menü

- Option Array transponieren
 - Intensitätsgraph, 15-38
 - Popup-Menü des Graphen, 15-6, 15-12
 - Popup-Menü des Kurvendiagramms, 15-24
- Option Attributknoten erstellen, 8-3, 22-1 bis 22-4
- Option Auf eine Zeile begrenzen, 11-6
- Option Auf Fenstergröße anpassen, 6-5
- Option Ausführen, 4-1
- Option Ausgang hinzufügen, 20-2
- Option Ausschneiden, 2-14
- Option Auswahl anzeigen, 14-15
- Option Auswahl beenden, 14-16 bis 14-17, 15-30
- Option Auswahl beginnen, 14-16, 15-30
- Option Auswahlmodus, Popup-Menü des Listenfelds, 13-5 bis 13-6
- Option Auto-Handling von Menüs beim Start, 6-5
- Option Autom. Skalierung X, 15-17
- Option Autom. Skalierung Y, 15-17
- Option Automatisch Zentrieren, 6-5
- Option Automatische Aktualisierung vom Verzeichnis, 7-36
- Option Automatisches Aktualisieren von Typdefinitionen, 24-22
- Option Balkendiagramme, 15-20
- Option Bearbeitungsmodus, 24-6
- Option Bedienelement bearbeiten..., 24-2, 24-16
- Option Bedienelement erzeugen
 - Dialogfeld ActiveX-Objekt auswählen, 16-4
 - Popup-Menü Terminal, 17-13
- Option Bedienelement suchen, 22-6
- Option Bedienelement, Dialogfeld Benutzerspezifische Druckeinstellungen, 5-5
- Option Beim Start verfügbare Datenträgerkapazität prüfen, 7-9
- Option Beschreibung, Menü Datenoperationen, 2-29
- Option Bildobjekt, 24-13
- Option Booleschen Text verbergen, 10-4
- Option Boolescher Text, 10-4
- Option Case anzeigen, 19-21
- Option Case duplizieren, 19-21
- Option Case entfernen, 19-22
- Option Case hinzufügen, 19-17
- Option Cases neu anordnen, 19-17, 19-22
- Option Cluster-Anordnung..., 4-15, 14-23
- Option Cluster-Größe..., 14-29
- Option CVI FP Datei konvertieren, 25-10
- Option Das Elemente-Fenster anzeigen, 24-8
- Option Daten ausschneiden, 11-9, 14-13
- Option Daten einfügen, 11-9
- Option Daten freigeben..., 4-6
- Option Daten kopieren
 - Menü Datenoperationen, 11-9
 - Probe-Werkzeug, 4-26
- Option Datenbankzugriff aktivieren, 4-7
- Option Datenbereich..., 4-18
 - Bedien- und Anzeigeelemente mit Schieberegler, 9-13, 9-15
 - numerische Bedien- und Anzeigeelemente, 9-6
- Option Dialogfeld, 6-4
- Option Dialogfeld, VI-Einstellungen, 6-4
- Option Dialogschriftart..., 2-20
- Option Diesen Parameter löschen, 25-5
- Option Dieses Terminal trennen, 3-10
- Option Digitale Anzeige, 2-15
- Option Digitale Anzeige, Menü Anzeigen, 2-15, 15-24
- Option Dimension entfernen, 14-8
- Option Direkte SubVIs anzeigen, 3-20
- Option Dokument erstellen, 16-3, 16-4

- Option Dokumentation drucken, 5-1
 - Benutzerspezifische Druckeinstellungen, 5-4 bis 5-5
 - Blockdiagramm, 5-7
 - Frontpanel, 5-6
 - Liste des Sub-VIs, 5-7
 - VI- Hierarchie, 5-7
 - VI-Anschluß und Icon, 5-6
 - VI-Historie, 5-7
- Option Drag, 15-33
- Option Drag zulassen, 15-33
- Option Druckereinstellungen, Menü Datei, 5-2
- Option Einfügen, 2-14, 17-12
- Option Eingang hinzufügen, 17-10, 20-2
- Option Einstellungen SubVI-Knoten, 6-7
 - Abbildung, 6-7
 - Optionen, 6-7
- Option Eintrag bei jedem Speichern hinzufügen, 6-6
- Option Element entfernen, 19-12
- Option Element hinzufügen, 19-12
- Option Entwicklungsvertrieb, 27-7
- Option Erneut Zeichnen, 3-18
- Option Ersetzen, 8-4, 17-12
- Option Farbe
 - Palette, 2-26
- Option Farbe,
 - Diagrammanzeigeelemente, 15-21
- Option Farbe/Graustufen drucken, 7-17
- Option Farben interpolieren, 9-25
- Option Farbzuzuweisung, Intensitätsdiagramm, 15-37 bis 15-38
- Option Fehlerliste zeigen,
 - Windows-Menü, 4-10
- Option Fenster drucken
 - Zweck, 5-1
- Option Fixiert zu Plot, 15-33
- Option Format & Genauigkeit...
 - Attributknoten, 22-5
 - in Popup-Menüs des Graphen, 15-15
- Option Format & Präzision...
 - digitale Anzeigen
 - absolute Zeit und Datum, 9-10
 - Beispiel, 9-9
 - Dialogfeld Format & Präzision (Abbildung), 9-9
 - Graph-Anzeigen
 - Attributknoten, 22-5
 - in Schieberegler-Popup-Menüs, 9-14
 - Zeit und Datum formatieren, 15-16
- Option Format umwandeln, 9-6
- Option Formatieren...,
 - Graphenanzeigeelemente, 15-14
- Option Frei, 15-33
- Option Frontpanel nach dem Laden anzeigen, 6-2
- Option Füllen zur Basislinie, 15-21
- Option Geänderte VIs, 27-7
- Option Gehe zu Cursor, 15-32
- Option Globale Variable aufnehmen, 3-18
- Option Graphik beim Zeichnen glätten, 7-10
- Option Graphik glätten, 15-13
- Option Groß/Kleinschreibung nicht beachten, Tastaturmodus, 13-6
- Option Größenanpassung an Text, 17-5
- Option Hex-Anzeige, 11-6
- Option Historienlänge..., 15-25, 15-37
- Option Horizontale Hierarchie, 3-18
- Option Icon bearbeiten..., 3-2
- Option Icon und Beschreibung, 5-6
 - VI-Anschluß und Icon, 5-6
 - VI-Beschreibung, 5-6
- Option Icon, Beschreibung, Panel und Diagramm, 5-4
- Option Ignorieren, numerische Bereichsprüfung, 9-6
- Option In Anzeigeelement ändern, 8-3, 14-22, 18-11
- Option In Ausgang ändern, 20-3
- Option In Bedienelement ändern, 8-3, 14-22, 18-12

- Option In die Mitte bringen, 15-32
- Option In Eingang ändern, 20-4
- Option In Globale lesen ändern, 23-3
- Option In Globale schreiben ändern, 23-3
- Option In Lesen ändern, 21-6, 22-2
- Option In Lokale lesen ändern, 23-5
 - mehrere lokale Variablen, 23-5
- Option In Lokale schreiben ändern, 23-5
- Option In Setzen ändern, 21-6, 22-2
- Option Indizierung aktivieren, 19-8
- Option Indizierung deaktivieren, 19-8
- Option Interpolation, 15-21
- Option Kein Pfad, 12-2
- Option Kommunikation,
 - Funktionenmenü, 5-1
- Option Konfigurieren, Funktion Aufruf von Bibliotheken, 25-4
- Option Konstante erzeugen, 17-4
- Option Kooperationsebene, 7-8
- Option Kopfzeile drucken, 5-5
- Option Kopie speichern unter..., 2-31
- Option Kopieren, 2-14
- Option Kopieren in die Zwischenablage,
 - Popup-Menü Gestalterische Elemente, 24-11
- Option Label, Untermenü Anzeigen, 2-3, 2-18, 2-19
- Option Letztes Element anzeigen, 14-14
- Option Linienbreite, 15-20
- Option Linienstil, 15-20
- Option Lokale Sequenz-Variable hinzufügen, 19-19
- Option Löschen, Menü Bearbeiten, 2-15
- Option Marker hinzufügen, 9-17
- Option Marker löschen, 9-17
- Option Markerabstand, 15-13
- Option Mehr..., Farbpalette, 2-27
- Option Menüeinträge einfügen, 6-16, 6-16 bis 6-18
- Option Menüeinträge löschen, 6-16, 6-18 bis 6-19
- Option Mit Optionen speichern..., 2-31, 27-3, 27-6
- Option Nach Abarbeitung schließen, 6-2
- Option Nach dem Schließen nach Kommentar fragen, 6-6
- Option Nach dem Speichern nach Kommentar fragen, 6-7
- Option Nach öffnen ausführen, 6-3
- Option Namen anzeigen, 15-31
- Option Neues Paßwort anwenden, 27-8
- Option Objekt aus Datei erstellen, 16-3, 16-4
- Option Objekt danach einfügen, 9-20, 13-9, 13-13
 - Bedienelement Enumerated Type, 13-13
 - Ring, Popup, 13-8
 - Textbeschriftungen, Popup, 9-18
- Option Objekt davor einfügen, 9-20, 13-9, 13-13
 - Bedienelement Enumerated Type, 13-13
 - Ring, Popup, 13-8
 - Textbeschriftungen, 9-18
- Option Objekt deaktivieren
 - Bedienelemente des Listenfeldes, 13-7
 - Bedienelemente des Rings, 13-10
- Option Originalgröße, 24-12
- Option Panelanordnung..., B-14
- Option Panelreihenfolge, 8-8
- Option Parameter danach einfügen, 25-5
- Option Parameter davor einfügen, 25-5
- Option Paßwort-Anzeige, 11-6
- Option Priorität, 6-3
- Option Probe anpassen, 4-26
- Option Protokoll aktivieren/Drucken nach Abschluß, 6-5
- Option Protokolldatei-Bindung löschen, 4-6
- Option Punktstil, 15-20
- Option Reihenfolge von Clustern, 4-15, 14-23
- Option Repräsentierung, 9-4
- Option Rücksetzen auf Standard, 14-13, 14-23
- Option Run-Time Popup-Menü zulassen, 6-5

- Option Schieber hinzufügen, 9-21
- Option Schieberegister hinzufügen, 19-10
- Option Seite einrichten, Menü Datei, 5-2
- Option Seitenumbruch zwischen Abschnitten, 5-5
- Option Sequenz-Variable hinzufügen, 19-19
- Option Skalierung, 9-15, 15-15
- Option Skalierungsfaktoren, 15-15
- Option Speicher schnellstmöglich freigeben, 7-7
- Option Speichern, 24-5, 24-21
- Option Speichern anpassen, 27-8
- Option Speichern unter..., 2-31
- Option Stapelplot, 15-27
- Option Startauswahl, 14-17
- Option Stil, 9-15
- Option Stopp, Menü Bedienen, 4-4
- Option SubVI erstellen, 3-10
- Option Terminal suchen, 8-3, 22-6
- Option Terminals anzeigen, 17-1, 17-10
- Option Text in Mitte fixiert, 10-4
- Option Textbeschriftungen, 9-18
- Option Typendefinition aufnehmen (Type Defs aufnehmen), 3-18
- Option Umkehren..., 24-12
- Option Unabhängige Größen, 24-15
- Option Ungültige Verbindungen entfernen, 4-9, 18-9, 18-12
- Option Unterdrücken, 9-7
- Option Verbindung suchen, 4-26
- Option Verbindungen hervorheben, 3-20
- Option Verdichten des Speichers während der Ausführung, 7-8
- Option Vertikale Hierarchie, 3-18
- Option VI-Hierarchie anzeigen
 - Popup-Menü von Hierarchieknoten, 3-19, 3-20
 - Popup-Menü von SubVIs, 3-15
 - Projekt-Menü, 3-15
- Option VI-Info anzeigen... (Hinweis), 2-29
- VI-Beschreibungen bearbeiten (Hinweis), 2-29
- VIs dokumentieren, 2-30
- Option VIs in vi.lib aufnehmen, 3-18
- Option Vollständigen VI-Pfad im Label (Full VI Path in Label), 3-18
- Option Von Typdefinition trennen, 24-23
- Option Voreinstellungen..., 7-1
- Option Warnungen zeigen, 4-10
- Option Werkzeugpalette anzeigen, 6-5
- Option While-Schleife entfernen, 2-15
- Option Wiedergewinnen...
 - Menü Bedienen, 4-6
 - Untermenü Datenprotokollierung, 4-6
- Option Zeiger hinzufügen, 9-22
- Option Z-Skala, 15-37
- Option Zurückkehren...", 2-32
- Optionen des Dialogfelds Umwandlung. *Siehe*
 - Option LabVIEW/CVI Funktionenpanel
- Optionen des Menü-Editors, 6-12 bis 6-13
- Optionen des Menüs Bearbeiten, 6-9
- Optionen Dokumentation, Dialogfeld VI-Einstellungen. *Siehe* auch
 - Beschreibungen., 6-5 bis 6-7
 - Abbildung, 6-6
 - Eintrag bei jedem Speichern hinzufügen, 6-6
 - Feld Hilfefeld, 6-7
 - Hilfe Tag, 6-7
 - Nach dem Schließen nach Kommentar fragen, 6-6
 - Nach dem Speichern nach Kommentar fragen, 6-7
 - Standard zur Historie verwenden (im Dialog "Voreinstellungen"), 6-6

P

- Palette Array & Cluster, 14-1, 14-5
- Palette Elemente
 - Bedien- und Anzeigeelemente auswählen, 1-4
 - Boolsche Palette, 10-1
 - Graphenpalette, xxv, 15-33
 - Hinzufügen von benutzerspezifischen Bedienelementen, 24-1
 - Option Ein Bedienelement auswählen..., 24-5
 - Palette Array & Cluster, xxv, 2, 14-1
 - Palette Graph, 2
- Palette Erweitert, 25-3
- Palette Farben
 - Abbildung, 2-26
- Palette Fortgeschritten, 25-3
- Palette Funktionen
 - Konstanten verfügbar in, 17-4
- Palette Gestaltungselemente, 5-9, 24-17
- Palette Graph, 2
- Palette Numerisch, 8-3
- Palette Probe anpassen, 4-27
- Palette Repräsentation, 13-12
- Palette Schaltverhalten, 10-4
- Palette Semaphore, 26-21, 26-23
- Parallele Diagramme, 28-10
- Parameter danach einfügen, 25-5
- Parameter davor einfügen, 25-5
- Pattern, 3-6
- Performance Profiler, 28-1 bis 28-7
- Pfad
 - Datenspeicherformate, A-5
 - gewandelte Daten, A-13
- Pfadkonstanten, universelle, 17-7
- Pfadvoreinstellungen, 7-2 bis 7-6
 - Bibliothekverzeichnis, 7-2
 - Dialogfeld für, 7-2
 - Standardverzeichnis, 7-4
 - temporäres Verzeichnis, 7-4
 - VI-Suchpfad, 7-4 bis 7-6
- Pipette-Werkzeug, Icon-Editor, 3-4
- Plattformthemen und Kompatibilitäten, Fragen zu, B-6 bis B-7
- Platz, zu Panel oder Diagramm hinzufügen, 2-26
- Plots, 15-1
- Polymorphe Einheiten, 9-33
- Popup-Fenster des Anschlußfeldes
 - Alle Terminals trennen, 3-10
 - Diese Verbindung ist, 3-9
 - Dieses Terminal trennen, 3-10
 - Horizontal kippen, 3-7
 - Um 90 Grad drehen, 3-7
 - Vertikal kippen, 3-7
- Popup-Menü des Graphen, 15-12
- Popup-Menü des Icon-Anschlußfeldes, 3-5
- Popup-Menü für Objekte, 2-5
- Popup-Menü Plotmuster, 15-20
- Popup-Menü Skala, 9-14
- Popup-Menü VI-Liste, 4-10
- Popup-Menüs
 - Anwendung, 2-1 bis 2-2, 2-6 bis 2-7
 - Definition, 2-6
 - Zugriff auf, 1-4
- Popup-Menüs des Indexarrays, 14-13
- Popup-Panels erstellen, 6-1 bis 6-2
- Portierbarkeitsthemen, 29-1 bis 29-3
 - Auflösungs- und Schriftartenunterschiede, 29-3 bis 29-5
 - Labels, 29-5
 - vordefinierte Schriftarten, 29-4 bis 29-5
 - bequemes Portieren zwischen Plattformen, 29-2
- Bildunterschiede, 29-5
- nicht-portierbare VIs, 29-1 bis 29-2
- Trennzeichenunterschiede, 29-3
- Überblick, 29-1

Positionierwerkzeug
 Abbildung, 2-5
 Objekte auswählen, 2-10
 Objekte verschieben, 2-10 bis 2-11

Präzision von Digitalanzeigen, Ändern,
 9-8 bis 9-11

Prioritäten, in Multitasking. *Siehe*
 Multitasking.

Probe
 Auto-Probe während Highlight-Option,
 auswählen, 7-13
 Optionen, 4-26 bis 4-27
 Zweck und Verwendung, 4-24 bis 4-26

Probewerkzeug, 2-5

Profilfenster, 28-1 bis 28-7
 Abbildung, 28-2
 Ergebnisse einsehen, 28-3
 Gesamtzeit, 28-4
 Kontrollkästchen Speicherauslastung,
 28-6
 Kontrollkästchen
 Speicherverwendungsprofil, 28-2
 Kontrollkästchen Timing-Statistik, 28-4
 Speicherinformationen, 28-6
 SubVI-Zeit, 28-4
 Taste Schnappschuß, 28-2
 Taste Speichern, 28-2
 Taste Start, 28-2
 Timing-Informationen, 28-4 bis 28-5
 VI-Zeit, 28-4

Programmatische Datenabfrage, 4-7 bis 4-9

Programmatisches Drucken
 aktivieren/deaktivieren, 6-5
 Seitenlayout einrichten, 5-9 bis 5-10
 Steuern des Ausdruckzeitpunkts, 5-8
 Verbessern der Drucke, 5-9

Protokolldatei-Bindung ändern... Option, 4-6

Pulldown-Menüs, 2-6

Punkte, 14-20, 15-4

Q

Quellterminals
 Definition, 17-2

Queue RefNum, 12-6

Queue-VIs, 26-24

R

Radix von Integer-Werten, auswählen, 9-4

Rahmen, Panel in R. drucken, 5-7

Rechte Maustaste für Zugriff auf
 Pop-up-Menüs, 1-4

Rechteck-Werkzeug, Icon-Editor, 3-4

Refnum-Bedienelemente
 Abbildung, 12-3
 Anwendungs- oder VI-Refnum, 12-5
 Arten, 12-3 bis 12-6
 Arten von Renums (Abbildung), 12-3
 Automation-RefNum
 Beispiele, 12-6 bis 12-7
 Byte-Stream-Datei-RefNum, 12-4
 Datenprotokoll-Datei-RefNum, 12-4
 Definition, 12-3
 Geräte-RefNum, 12-4
 Netzwerkverbindung-RefNum, 12-4
 Notifier-RefNum, 12-5
 Occurrence-RefNum, 12-4
 Queue-RefNum, 12-6
 Rendezvous-RefNum, 12-6
 Semaphore-RefNum, 12-6
 strikte Typen-VI-RefNum, 12-5 bis 12-6
 VISA-RefNum, 12-5
 wann sie verwendet werden (Hinweise),
 12-4 bis 12-7
 Zweck und Verwendung, 12-2 bis 12-7

Rendezvous-RefNum, 12-6

Resizing-Cursor, 2-25

Resizing-Handles. *Siehe*
 Größenanpassungs-Ziehpunkte.

Return-Taste
 Texteingabe beenden, 7-10
Revisionsnummern für VIs, 27-12 bis 27-13
Ring-Konstanten, 17-6
Ring-Steuerungen
 Attributknoten Beispiel, 22-10 bis 22-11
Rollbalken
 Farbenvoreinstellungen, 7-14
 Verwendung mit Bedien- und
 Anzeigeelementen vom Typ
 String, 11-3
Rollbalkenoption, 11-3
Rücksetzen der Historie-Informationen, 27-13
Runde Numerik, 9-22 bis 9-24
 Abbildung, 9-22
 Zweck und Verwendung, 9-22 bis 9-23
Run-Time Popup-Menü zulassen, 6-4
Run-Time-VIs, 27-3 bis 27-4

S

Schieberegister
 Definition, 19-10
 linke und rechte Terminals, 19-10
 Zweck und Verwendung, 19-10
Schieberegler mit Mehrfachwerten,
 9-20 bis 9-21
Schleifen. *Siehe* For-Schleife; While-Schleife.
Schriftarten
 Labels, 2-19 bis 2-24
 Unterschiede beim Portieren von VIs
 zwischen Plattformen, 29-3 bis 29-5
 vordefinierte Schriftarten in LabVIEW,
 29-4 bis 29-5
 Voreinstellungen für Schriftarten
 angeben, 7-15
Schriftarttring, 2-21
Schrittasten, 4-19, 4-21
Schwarzweiß-Icons gegenüber
 Farb-Icons, 3-3

Scrolloptionen, Graphenanzeigeelemente,
 15-17 bis 15-19
Semaphore erfassen VI, 26-21
Semaphore zerstören VI, 26-22
Semaphore. *Siehe* auch
 Abstimmungsfunktionen., 26-21 bis 26-23
 Beispiel, 26-22
 Definition, 26-21
 erfassen, 26-21
 erstellen, 26-21
 freigeben, 26-22
 verwenden, 26-21
 zerstören, 26-22
Semaphore-RefNum, 12-6
Sendet AE Dokument-drucken VI
Sequenzstrukturen
 Anzeigefenster für Subdiagramme, 19-13
 Bearbeitung von Subdiagrammen, 19-20
 Beispiele in examples.llb.vi, 19-18
 Hinzufügen von Subdiagrammen,
 19-21 bis 19-22
 Icon, 19-2, 19-13
 lokale Sequenzvariablen, 19-19
 Löschen von Subdiagrammen, 19-22
 neu ordnen, 19-20
 Überblick, 17-11, 19-18
 Verbindungsprobleme
 keine Verbind. zu einem Tunnel in
 allen Cases einer
 Case-Struktur, 19-23
 überlappende Tunnel, 19-23
 Verbindung unterhalb statt durch
 Strukturen, 19-25 bis 19-26
 Verbindung von mehreren Rahmen
 einer Sequenzstruktur, 19-25
 Zuweisen mehrerer Werte zu lokalen
 Sequenz-Variablen, 19-22
 Verschiebungen zwischen
 Subdiagrammen, 19-21
 Zweck und Verwendung, 17-11,
 19-18 bis 19-20

- Serielle Schnittstellen-VIs
 - zum Senden von Text an Drucker, 5-1
- Server: Exportierte VIs
 - Beispiele für Listeneinträge für exportierte VIs (Tabelle), 7-29
 - Liste mit Platzhaltern, 7-29
- Sichern von Dateien, 27-2
- Single- (SGL) Fließkommazahlen, Auswahl von, 9-4
- Singlethread-Ausführung. *Siehe* auch Multithread-Ausführung.
 - Benutzerschnittstelle bei einer Singlethread-Anwendung, 26-4
 - Festlegen von Taskprioritäten, 26-8 bis 26-11
 - Singlethread- gegenüber Multithread-Anwendungen, 26-3
- Skalargrenzen für numerische Bedienelemente, Ändern, 9-15 bis 9-18
- Skalar-Datentypen
 - gewandelte Daten, A-13
 - numerische und nicht numerische (Tabellen), A-8 bis A-9
- Skalierungsoptionen,
 - Graphenanzeigeelemente, 15-12 bis 15-17
- Sonderzeichen. *Siehe* Bedien- und Anzeigeelemente vom Typ String.
- Sonstige Voreinstellungen, 7-22
- Spalten
 - Ändern der Größe, 11-7, 11-7 bis 11-8
 - Überschriften, 11-7
- Speicher
 - Voreinstellungen angeben, 7-7
- Speicherauslastung, 28-14 bis 28-33
 - allgemeine Regeln zur Verbesserung, 28-21 bis 28-22
 - Datenfluß-Programmierung und Datenpuffer, 28-17 bis 28-19
 - dynamisches Laden von VIs, B-11
 - Freigabe durch LabVIEW, 28-25 bis 28-26
 - Frontpanels, 28-22 bis 28-23
 - globale Variablen, 28-24
 - grundlegende Konzepte, 28-16 bis 28-19
 - lokale Variablen, 28-24
 - Überblick, 28-14
 - Überwachung, 28-20
 - VI-Komponenten, 28-16 bis 28-17
 - virtueller Speicher, 28-15
 - Wiederverwendung des Datenspeichers durch SubVIs, 28-24
 - Wiederverwendung von Eingangspuffer durch Ausgang, 28-26
- Speicherformate für Boolesche Daten, A-1
- Speichern benutzerspezifischer Bedienelemente, 24-5
- Speichern von Typendefinitionen, 24-21
- Speichern von VIs
 - Speichern zur Verteilung, 27-6 bis 27-8
- Speicheroption, 2-31
- Standardmäßig Warnungen im Fehlerfenster anzeigen (Option), 7-13
- Standard-Timer, Verwenden, 7-7 bis 7-8
- Standardverzeichnis, Pfad angeben für, 7-4
- Standardverzeichnisse, Einstellen, 7-3 bis 7-4
- Steuerungen
 - Steuerungen in Dialogfeldern, 8-9 bis 8-10
- Steuerungen in Dialogfeldern
 - Anpassen, 8-9
- Steuerungsflußausführung
 - in Sequenzstrukturen, 19-18
- Stoppen der Ausführung von VIs, 4-4
- <Strg-b>, unterbrochene Verbindungen entfernen, 4-9
- <Strg-Eingabe>, neue Zeilen einbetten, 7-10
- <Strg-f>, Dialogfeld Suchen aufrufen, 3-23
- <Strg-h>, Hilfetaste, 1-7
- <Strg-klicken>
 - Aktion Hierarchie anzeigen ausführen, 3-21
 - Arbeitsbereich hinzufügen, 2-26

- Farben kopieren und übertragen, 2-28
- letzten Heftpunkt löschen (Hinweis), 18-3
- <Strg-M> Zugriffstaste für Bearbeiten/Ausführen (Hinweis), 6-5
- <Strg-m>, zwischen Ausführungs- und Bearbeitungsmodus wechseln, 6-5
- <Strg>-Taste
 - Attributknoten klonen, 22-4
 - Blockdiagramm eines SubVIs in den Vordergrund holen, 4-22
 - Kurzbefehl Herausspringen, 4-21
 - Kurzbefehl Hineinspringen, 4-21
 - Kurzbefehl Überspringen, 4-21
 - Neuen Skalenmarker erstellen, 9-18
 - Objekte klonen, 2-14
 - Werkzeuge des Icon-Editors in Pipette umwandeln, 3-5
 - zwischen Arrayelementen wechseln, 14-14
 - zwischen Cluster-Elementen wechseln, 14-22
- <Strg-Umschalt>, temporäre Werkzeugpalette aufrufen, 2-4
- <Strg-Umschalt-Z>-Taste, Aktion Wiederherstellen, 7-32
- <Strg-Z>-Taste, Aktion Rückgängig machen, 7-32
- Strikte Typendefinitionen
 - Bearbeiten (Hinweis), 24-21
 - Speichern, 24-21
 - Zweck und Verwendung, 24-20
- String (Palette), 8-3
- String-Bedien- und Anzeigeelemente
 - Option Hex-Anzeige, 11-6
- String-Datentypen, 25-7
- String-Konstanten, 17-5
- String-Popup-Menü, 11-3
- Strings
 - Datenspeicherformate, A-5
 - gewandelte Daten, A-13

Strukturen

- Definition, 1-6, 17-11
- Effizienzerwägungen, 28-34 bis 28-42
 - globale Tabelle gemischter Datentypen, 28-38 bis 28-41
 - statische globale Tabelle mit Strings, 28-41
 - Vermeiden komplizierter Datentypen, 28-36 bis 28-37
- Plazierung und Größenanpassung in Blockdiagrammen, 19-5
- Tunnel, 17-12
- Typen von, 17-11
- Überblick, 19-18
- Verbindungsprobleme
 - Entfernen von Strukturen, ohne Inhalte zu löschen, 19-26
 - keine Verbind. zu einem Tunnel in allen Cases einer Case-Struktur, 19-23
 - überlappende Tunnel, 19-23
 - Verbindung von mehreren Rahmen einer Sequenzstruktur, 19-25
 - Zuweisen mehrerer Werte zu lokalen Sequenz-Variablen, 19-22
- Strukturunabhängige Datentypen, A-12
- Subdiagramme
 - Definition, 19-2
 - Hinzufügen von, 19-21 bis 19-22
 - Löschen von, 19-22
 - neu anordnen, 19-21
 - Verschachtelung, 17-11
 - Verschiebungen zwischen, 19-21
- SubVI-Knoten Einstellungsoptionen, 4-30
- SubVIs
 - anhand einer Auswahl bestätigen, 3-10 bis 3-14
 - aus einem VI erstellen, 3-2 bis 3-10
 - bei Plazierung auf dem Blockdiagramm Name d. SubVIs anzeigen, 7-12

Beschreibungen bearbeiten
 (Hinweis), 2-29
 Breakpoints, 4-27 bis 4-31
 darstellende Icons, 1-6
 Druckliste des SubVIs, 5-7
 Entsprechung von Unterprogrammen
 in C, 3-1
 erstellen. *Siehe* Erstellen von SubVIs.
 Labels auf dem Blockdiagramm
 (Hinweis), 2-19
 Leistungserwägungen, 28-11
 nicht editierbare Labels (Hinweis), 2-19
 Speicherauslastung, 28-23
 SubVI durch ein anderes VI mit selbem
 Namen ersetzen, B-11
 Unmöglichkeit, Beschreibungen vom
 Diagramm d. aufruf. VIs aus zu
 bearbeiten (Hinweis), 2-29
 Verbindungen mit dem aufrufenden VI,
 im Hierarchiefenster, 3-15 bis 3-16
 Wiederverwendung des
 Datenspeichers, 28-24
 zwei verschiedene SubVIs mit selbem
 Namen aufrufen, B-12
 Suchen (Dienstprogramm), 3-23 bis 3-25
 Symbolleiste
 Optionen des Menü-Editors, 6-13
 verborgen, 6-5
 Systemabstürze, Fragen zu, B-4 bis B-5

T

Tabellen, 11-7 bis 11-10
 Abbildung, 11-7
 Ändern der Größe von Tabellen, Zeilen
 und Spalten, 11-7 bis 11-8
 Definition, 11-7
 Eingabe und Auswahl von Daten,
 11-8 bis 11-10
 Tags von Menüpunkten der Anwendung
 (Tabelle), 6-20 bis 6-23

Taste Anhalten, 4-19
 Taste Durchsuchen, 16-5
 Taste Gestörte Ausführung, 4-9, 18-9
 Taste Herauspringen, 4-21
 Taste Highlight-Funktion, 4-22
 Taste Hineinspringen, 4-21
 Taste Layout wiederherstellen, 3-19
 Taste Löschen, 4-6
 Taste Löschen,
 Frontpanel-Datenprotokollierung, 4-6
 Taste nicht OK, 24-4
 Taste Stopp
 Anhalten von VIs, 4-4
 verbergen, 4-4, 6-5
 Zweck und Verwendung, 4-2, 4-4
 Taste Überspringen, 4-21
 Taste Unterbrochene Ausführung. *Siehe* auch
 Taste Gestörte Ausführung, 4-9
 Taste Unterbrochene Ausführung. *Siehe* auch
 Taste Gestörte Ausführung., 18-9
 Taste Wiederholt Ausführen, 4-4
 Taste Zum Löschen markierter Datensatz, 4-6
 temporäres Verzeichnis, Pfad angeben für, 7-4
 Terminal-Muster für Anschluß
 Bedien- und Anzeigeelementen Terminals
 zuweisen, 3-7 bis 3-8
 definieren, 3-5 bis 3-6
 Hierarchiefenster, 3-15 bis 3-22
 Muster ändern, 3-6
 Muster auswählen, 3-6
 Verbindungen bestätigen, 3-10
 Verbindungen löschen, 3-9 bis 3-10
 Terminals
 als Eingangs- und
 Ausgangsanschlüsse, 1-5
 Anzeige, 17-1
 Arten von, 17-2
 Ausgangsterminals, 1-5
 Bedien- und Anzeigeelemente, 17-2
 Bedingungsterminal, 19-4
 Definition, 17-2

Eingangsterminals, 1-5, 17-2
Iterationsterminal, 19-3
Knoten-Eingabeterminals, 17-2
löschen, 1-5
Plazierung in For- und
 While-Schleifen, 19-6
Quellterminals, 17-2
Schieberegisterterminals, 19-10
Terminals für Bedien- und
 Anzeigeelemente, 17-2
Zählerterminal, 19-3
Terminals für Bedien- und
 Anzeigeelemente, 17-2
Text- & Bild-Ring, Ändern von Größe und
 Text, 13-11 bis 13-12
Text suchen, 3-25 bis 3-27
Textbeschriftungen
 numerische Skalen, 9-18 bis 9-20
Textelemente, 24-15
Textlabels
 Schriftart, Stil, Größe und Farbe ändern,
 2-19 bis 2-24
Text-Menü
 Option Text in Mitte fixiert, 10-4
Text-Werkzeug, Icon-Editor, 3-4
Trennlinie, für Bedienelemente des
 Listenfeldes, 13-7
Trennzeichen, Punkt und Komma, 29-14
Trennzeichenunterschiede beim Portieren von
 VIs zwischen Plattformen, 29-3
Tunnel
 Definition, 17-12, 18-2
 Verbindungsprobleme, 19-23
 verschieben, 18-7
Typ-Deskriptoren
 allgemeines Format, A-7
 Arrays, A-10
 Cluster, A-11
 Datentypen
 Beispiele, A-10
 nicht num. (Tabelle), A-9

numerische Skalar-Datentypen
 (Tabelle), A-8 bis A-9
Speicherung physikalischer Mengen
 (Hinweis), A-10
 Definition, A-7
 Überblick, A-7
Typendefinitionen, 24-19 bis 24-24
 Aktualisierung, 24-22 bis 24-23
 automatisches Aktualisieren, 24-22
 Cluster-Typendefinitionen, 24-23
 Datentypenanforderungen, 24-19
 Definition, 24-19
 Erstellen, 24-21
 im Hierarchiefenster aufnehmen, 3-18
 Speichern, 24-21
 strikte Typendefinitionen, 24-20
 Suchen, 24-23
 Trennen, 24-23
 übereinstimmendes Datentypen, 24-19
 Zweck und Verwendung, 24-21 bis 24-22

U

Überlagerte Plots, 15-27
Um 90 Grad drehen, 3-7
umgekehrte Schrägstrich-(\) Codes
 Beispiele, 11-4 bis 11-6
 Eingabe und Ansicht von nicht
 anzeigbaren Zeichen, 11-4 bis 11-6
 Hinweis, 11-2
 Tabelle mit Codes, 11-4
<Umschalttaste> drücken und auf Objekte
 klicken, 2-8
<Umschalt-Eingabe>-Taste
 entsprechenden Knoten suchen, 3-22
 Textbeschriftungen durchgehen, 9-20
 Textbeschriftungen erstellen, 9-18
<Umschalt>-Klicken
 Aktion Alle SubVIs anzeigen
 ausführen, 3-21
 Aus- und Abwählen, 2-7

- Auswählen
 - mehrere Knoten, 3-21
 - Tabellenzeilen und -spalten, 11-8
- Auswahlrechteck verwenden, 2-8
- <Umschalt-Return>-Taste
 - entsprechenden Knoten suchen, 3-22
 - Textbeschriftungen durchgehen, 9-20
 - Textbeschriftungen erstellen, 9-18
- <Umschalt>-Taste
 - Positionieren von Objekten, 2-10
 - unsichtbare Kurzbefehle unter
 - Macintosh-OS 7 (Hinweis), 6-10
- unabhängige Bestandteile, 24-7 bis 24-8
- undefinierte Daten, Erkennen, 4-16 bis 4-17
- unterdrückte VIs
 - Breakpoint-Techniken, 4-30 bis 4-31
 - numerische Bereichsprüfung, 9-7
- Untermenü Anzeigen
 - Elemente-Palette, 8-3
 - Option Boolescher Text, 10-4
 - Option Label, 2-18, 2-19
 - Rollbalkenoption, 11-3
- Untermenü Datenoperationen
 - Bearbeitungsmodus, 4-2 bis 4-4, 8-3
 - Befehl Aktuellen Wert als Standard, 14-23
 - Befehl Leeres Array, 14-13
 - Option Aktualisierungsmodus, 15-25
 - Option Auswahl beenden, 14-15, 15-30
 - Option Auswahl beginnen, 15-30
 - Option Auswahl beginnen, 14-15
 - Option Beschreibung..., 2-29
 - Option Daten ausschneiden, 11-9, 14-13
 - Option Daten einfügen, 11-9
 - Option Daten kopieren, 11-9
 - Option Graphik glätten, 15-13
 - Option Letztes Element anzeigen, 14-14
 - Rücksetzen auf Standard, 14-23
 - Zweck und Verwendung, 4-2 bis 4-4

- Untermenü Datenprotokollierung
 - Protokoll, 4-5
 - Protokolldateibindung ändern, 4-6
 - Protokolldatei-Bindung löschen, 4-6
 - Wiedergewinnen, 4-6
- Untermenü X-Achse, 15-13
- Untermenü X-Skala, 15-36
- Untermenü Y-Achse, 15-13
- Untermenü Y-Skala, 15-36
- Untermenü zeigen
 - Option Boolescher Text, 10-4
 - Option Digitale Anzeige, 9-14

V

- Verbesserung der VI-Leistung. *Siehe* Leistung.
- Verbinden von Blockdiagrammen, 19-22 bis 19-26
 - Einfügen von Objekten, 17-12
 - Ersetzen von Objekten, 17-12
 - grundlegende Verbindungstechniken, 18-1 bis 18-5
 - Heftpunkte, 18-3
 - komplizierte VIs, 18-4 bis 18-5
 - Maussymbol für Verbindungswerkzeug, 18-1
 - Strukturverbindungsprobleme
 - keine Verbindung zu Tunneln, 19-23
 - überlappende Tunnel, 19-23
 - Verbindung unterhalb statt durch Strukturen, 19-25 bis 19-26
 - Verbindung von mehreren Rahmen von Sequenzstrukturen, 19-25
 - Zuweisen mehrerer Werte zu, 19-22
- Verbindungen
 - Biegungen, 18-6
 - Kreuzungen, 18-6
 - Segmente, 18-6
 - Zweige, 18-6

- Verbindungsblockdiagramme
 - grundlegende Verbindungsverfahren, 1-6
- Verbindungswerkzeug
 - Abbildung, 2-5
 - Bedien- und Anzeigeelementen Terminals zuweisen, 3-7 bis 3-8
 - deaktiviert im
 - Bedienelement-Editor, 24-8
 - Hot Spot, 18-1
- Verhalten Latch bis zum Loslassen, 10-6
- Verhalten Latch während gedrückt, 10-5
- Verhalten Latch wenn losgelassen, 10-6
- Verhalten Schaltet bis losgelassen, 10-5
- Verhalten Schaltet wenn gedrückt, 10-5
- Verhalten Schaltet wenn losgelassen, 10-5
- Verschieben von Objekten
 - Cursor des Graphen, 15-30
 - Schieber, 9-13
- Verteilung von VIs
 - paßwortgeschützte VIs, 27-3
 - Überlegungen, 27-3 bis 27-8
 - Verwendung von Mit Optionen speichern, 27-7
- Vertikal kippen, 3-7
- Verzeichnis vi.lib, 2-33
- Verzeichnisse
 - einstellen, 7-3 bis 7-4
 - Pfad angeben für, 7-3 bis 7-4
 - Bibliothekverzeichnis, 7-3
 - Standardverzeichnis, 7-4
 - temporäres Verzeichnis, 7-4
 - VI-Suchpfad, 7-4 bis 7-6
- VI System Exec
 - Ausführen anderer Anwendungen von Ihren VIs aus, 25-1
- VI-Bibliothek bearbeiten
 - Dateien löschen, 2-35
- VI-Bibliotheken, 2-33 bis 2-35
 - Anordnung von Dateien in, 27-1 bis 27-2
 - Dateien in VI-Bibliotheken speichern, 2-33
 - erstellen, 2-34 bis 2-35
 - VIs entfernen, 2-35
 - VIs speichern im Verzeichnis vi.lib (Vorsicht), 2-33
- VI-Einstellungen
 - Ausführungsoptionen, 6-2 bis 6-3
 - Dokumentationsoptionen, 6-5 bis 6-7
 - Fensteroptionen, 6-3 bis 6-5
 - Popup-Panels erstellen, 6-1 bis 6-2
- VI-Einstellungsoptionen
 - Einstellung von Breakpoints, 4-27
 - verwandt mit Hist.-Informationen, 27-14
- VI-Hierarchie anzeigen, Projekt, 3-15
- VIs
 - automatisches Starten, B-14
 - Bedienung, 4-2 bis 4-9
 - Anhalten von VIs, 4-4
 - Ausführen von VIs, 4-1
 - Frontpanel-Datenprotokollierung, 4-5 bis 4-6
 - programmatische Datenabfrage, 4-7 bis 4-9
 - wiederholte Ausführung von VIs, 4-4
 - Benutzer am Schließen hindern, 6-4
 - Betrieb
 - Benutzer am Schließen von Ausführungs-VIs hindern, 6-4
 - Debugging, 4-9 bis 4-32
 - Ausschließen von Diagrammabschnitten, 4-32
 - Einstellung von Breakpoints, 4-27 bis 4-31
 - Einzelschrittausführung eines VI, 4-19 bis 4-21
 - Erkennen undefinierter Daten, 4-16 bis 4-17
 - Gründe für gestörte VIs, 4-10 bis 4-11
 - Highlight-Funktion, 4-22 bis 4-24
 - Korrektur des VI-Bereichsfehlers, 4-18

- LabVIEW-Funktionen, 4-19 bis 4-31
- mögliche Fehler, 4-11 bis 4-14
- Probe-Werkzeug, 4-24 bis 4-27
- Techniken, 4-14 bis 4-16
- Warnungen für überlappende Objekte, 4-16
- Definition, 1-1
- Laden
 - Fragen zu, B-12
- Merkmale, 1-2
- Portierbarkeitsthemen, 29-1 bis 29-3
 - Auflösungs- und Schriftartenunterschiede, 29-3 bis 29-5
 - bequemes Portieren zwischen Plattformen, 29-2
 - Bildunterschiede, 29-5
 - nicht portierbare VIs, 29-1 bis 29-2
 - Trennzeichenunterschiede, 29-3
 - Überblick, 29-1
- speichern, 2-31 bis 2-33
 - in Verzeichnissen außer vi.lib speichern (Vorsicht), 2-33
 - Speicheroptionen, 2-31
 - VI-Bibliotheken, 2-33 bis 2-34
 - VIs nach dem Speichern bearbeiten (Vorsicht), 2-32
 - Zurückkehren..., 2-32
- unterdrückte VIs
 - Breakpoint-Techniken, 4-30 bis 4-31
 - numerische Bereichsprüfung, 9-6 bis 9-8
- verglichen mit Funktionen in anderen Sprachen, 1-1 bis 1-2
- VIs bearbeiten
 - Anschlußterminal-Muster definieren, 3-5 bis 3-6
 - bearbeiten während VIs ablaufen, 4-2
 - Beschreibungen erstellen, 2-29 bis 2-31
 - Einteilen von Objekten, 2-13
 - Größenanpassung von Objekten, 2-25 bis 2-26
 - Arbeitsbereich auf dem Frontpanel oder Blockdiagramm, 2-26
 - Bilder, 2-25
 - Frontpanel, 2-25
 - Labels, 2-26
 - Icons erstellen, 3-2 bis 3-5
 - Objekte ausrichten, 2-12 bis 2-13
 - Objekte auswählen, 2-7 bis 2-9
 - Objekte duplizieren, 2-14
 - Objekte einteilen, 2-13
 - Objekte farbig kennzeichnen, 2-26 bis 2-28
 - Objekte löschen, 2-15
 - Objekte positionieren, 2-10
 - Objekte verschieben, 2-10 bis 2-11
 - Eins nach hinten, 2-12
 - Eins nach vorne, 2-12
 - Ganz nach hinten, 2-12
 - Ganz nach vorne, 2-11
 - Objektlabels
 - erstellen, 2-15 bis 2-24
 - freie Labels, 2-17 bis 2-19
 - Textattribute, 2-19 bis 2-24
 - während VIs arbeiten
 - Optionen des Popup-Menüs von Bedienelementen, 4-2
 - Optionen des Popup-Menüs von Objekten, 4-3
 - Ziehen und Ablegen von VIs, Bildern und Text, 2-9
 - zwischen Ausführ- und Bearbeitungsmodus wechseln, 6-5
- VIs dokumentieren, 2-30
- VIs drucken, 5-1 bis 5-14
 - aktives Fenster (Option Fenster drucken...), 5-2
 - aktivieren/deaktivieren, 6-5
 - alternative Druckmethoden, 5-10
 - Ausdrucken konfigurieren, 5-1 bis 5-3

- Auswahl der Druckoptionen, 5-3 bis 5-7
 - Dialogfeld Dokumentation drucken..., 5-3
 - Drucken der Kopfzeilen der Abschnitte, 5-5
 - Einstellen von Ausdruckformaten, 5-4
 - Erstellen von benutzerspezif. Druckeinstellungen, 5-5
 - Optionen des Seitenlayouts festlegen, 5-5
- PostScript drucken, 5-2
- programmatisches Drucken, 5-8 bis 5-10
 - Einstellen des Seitenlayouts, 5-9 bis 5-10
- Sendet AE Dokument-drucken, VI
- Steuern des Ausdruckzeitpunkts, 5-8
- Überblick, 5-1
- Verbesserung der Ausdrücke, 5-9
- VI's speichern, 2-31 bis 2-33
 - Speicheroptionen, 2-31
 - VI-Bibliotheken, 2-33 bis 2-34
 - VI's nach dem Speichern bearbeiten (Vorsicht), 2-32
 - Zurückkehren..., 2-32
- VI's und andere Objekte suchen, 3-24 bis 3-25
- VISA-RefNum, 12-5
- VI-Suchpfad, 7-4 bis 7-6
- Vordergrund/Hintergrund-Werkzeug, Icon-Editor, 3-4
- Voreinstellungen
 - Blockdiagramm, 7-11 bis 7-12
 - Abbildung, 7-11
 - Bei Platzierung Namen der SubVI's anzeigen, 7-12
 - Fenstertitel in Funktionspalette verwenden, 7-12
 - maximale Anzahl Rückgängig pro VI, 7-12, 7-32
 - Tip-Strips über Terminals anzeigen, 7-12
 - Transparente Namen-Label verwenden, 7-12
 - Verbindungshilfen anzeigen, 7-12
 - Verbindungspunkte an Kreuzungen anzeigen, 7-12
- Debugging, 7-13
 - Abbildung, 7-13
 - Auto-Probe während Highlight-Option, 7-13
 - Daten-Bubbles während der Highlight-Funktion anzeigen, 7-13
 - Standardmäßig Warnungen im Fehlerfenster anzeigen, 7-13
 - Warnen bei Objekten, die in der Studentenversion nicht verfügbar sind, 7-13
- Dialogfelder, 7-1 bis 7-23
- Drucker, 7-16 bis 7-18
 - Abbildung, 7-16
 - Bitmap drucken, 7-17
 - Drucker-Dithering zulassen, 7-17
 - Farbe/Graustufen, 7-17
 - PostScript drucken, 7-17
 - Seitenränder, 7-17
 - Standarddruck, 7-16
- Farbe, 7-14
 - Abbildung, 7-14
 - Blinken Hintergrund, 7-15
 - Blinken Vordergrund, 7-15
 - Blockdiagramm, 7-14
 - Formatumwandlungspunkt, 7-14
 - Frontpanel, 7-14
 - Menütext, 7-14
 - Rollbalken, 7-14
 - Standardfarben, 7-15
 - Use default color, 7-14
 - weitere Farben, 7-15

- Frontpanel, 7-9 bis 7-11
 - Abbildung, 7-9
 - Blinkgeschwindigkeit, 7-11
 - Die Funktions-Tasteneinstellungen des Systems übergehen, 7-10
 - Editor durch Doppelklick öffnen, 7-10
 - Graphik beim Zeichnen glätten, 7-10
 - Lokalen Dezimalpunkt verwenden, 7-10
 - Texteingabe mit Return-Schaltfläche (oder Eingabe-Taste), 7-10
 - Transparente Namen-Label verwenden, 7-11
 - Unterstützt den Zahlenblock von Sun-Tastaturen, 7-11
- Historie, 7-18 bis 7-20
 - Abbildung, 7-18
 - Aufzeichnen der von LabVIEW erzeugten Kommentare, 7-19
 - Einloggen mit dem Namen der LabVIEW-Registrierung, 7-20
 - Eintrag bei jedem Speichern hinzufügen, 7-19
 - Login-Abfrage beim Start von LabVIEW anzeigen, 7-20
 - Nach dem Schließen nach Kommentar fragen, 7-19
 - Nach dem Speichern nach Kommentar hinzufügen, 7-19
 - Revisionsnummer in der Titelbar anzeigen, 7-19
- Leistung und Speichermedium, 7-6
 - Beim Start verfügbare Datenträgerkapazität prüfen, 7-9
 - Dialogfeld (Abbildung), 7-6
 - Kooperationsebene, 7-8
 - Speicher schnellstmöglich freigeben, 7-7
 - Standard-Timer verwenden, 7-7
 - Verdichten des Speichers während der Ausführung, 7-8
- Pfade, 7-2 bis 7-6
 - Abbildung, 7-2
 - Bibliothekverzeichnis, 7-2
 - Dialogfeld für, 7-2
 - Format von Pfadnamen (Hinweis), 7-2
 - Standardverzeichnis, 7-4
 - Taste Entfernen, 7-6
 - temporäres Verzeichnis, 7-4
 - VI-Suchpfad, 7-4 bis 7-6
- Schriftarten, 7-15 bis 7-16
 - Abbildung, 7-15
 - Schriftart, 7-16
 - Schriftart anpassen..., 7-16
 - Standardschriftart, 7-16
- Server: Exportierte VIs, 7-27 bis 7-30
 - Abbildung, 7-28
 - Beispiele für Listeneinträge für exportierte VIs (Tabelle), 7-29
 - Liste mit Platzhaltern, 7-29
 - Taste Entfernen, 7-28
 - Taste Hinzufügen, 7-28
 - Taste Zugriff erlauben, 7-28
 - Zugriff verweigern, 7-28
- Server: Konfiguration, 7-23 bis 7-24
 - Abbildung, 7-23
 - Anwendungen und Einstellungen, 7-24
 - Taste Hinzufügen, 7-25
 - VI-Aufrufe, 7-24
 - VI-Methoden und Einstellungen, 7-24
- Server: TCP/IP-Zugriff, 7-24 bis 7-27
 - Abbildung, 7-24
 - fehlender Zugriff auf den DNS-Server (Hinweis), 7-27
 - Strikte Überprüfung, 7-26

- Taste Entfernen, 7-25
- Taste Hinzufügen, 7-25
- Taste Zugriff erlauben, 7-25
- Taste Zugriff verweigern, 7-25
- Zugriffsliste, 7-27
- sonstige, 7-22
 - Abbildung, 7-22
 - Direktklick zulassen, 7-22
 - Hot-Menüs verwenden, 7-22
 - Tip-Strips anzeigen, 7-22
 - VI im Ausführungsmodus öffnen, 7-23
- Speicherstelle, 7-30
- Uhrzeit und Datum, 7-21
 - Abbildung, 7-21
 - Datumtrennzeichen, 7-21
 - Standard Datumformat, 7-21
 - Standard Zeitformat, 7-21
 - Zeittrennzeichen, 7-21
- Voreinstellungen für Leistung, angeben, 7-6
- Voreinstellungen für Speichermedium, angeben, 7-6
- vorzeichenbehaftete und vorzeichenlose Integer-Werte
 - byte (8-Bit), 9-4
 - long (32-Bit), 9-4
 - Word (17-Bit), 9-4

W

- Wagenrücklauf
 - G '\'-Codes (Tabelle), 11-4
 - in String eingeben (Hinweis), 11-2
 - Option Auf eine Zeile begrenzen, 11-6
- Warnungen
 - im Fehlerfenster anzeigen, Voreinstellungen festlegen, 7-13
- Warnungstaste, 4-10
- Werkzeug Neuordnen Neuordnen. *Siehe* Neu anordnen.

- Werkzeuge
 - Definition, 2-4
 - Werkzeugpalette, 2-4 bis 2-6
 - Abbildung, 2-4
 - Ausführung von VIs bearbeiten, 4-2
 - temporäre Versionen, 2-4
 - verborgen (Hinweis), 6-5
 - verfügbare Werkzeuge, 2-5
 - Wechseln von Werkzeugen, 2-6
 - Zweck und Verwendung, 2-4 bis 2-6
- Werte außerhalb des Bereichs (Hinweis), 4-18
- Wettbewerbsbedingungen, 26-18
- While-Schleife
 - Auto-Indizierung, 19-7 bis 19-8
 - Icon für, 19-4
 - Plazieren von Objekten in Strukturen, 19-4
 - Schieberegister, 19-10 bis 19-13
 - Terminals in Schleifen, 19-6 bis 19-7
 - Vermeiden unnötiger Berechnungen, 28-12 bis 28-14
 - Zweck und Verwendung, 17-11, 19-4
- Wiedergewinnen von Datensätzen
 - Zum Löschen markierte Datensätze, 4-6
- Windows-Menü
 - Option Fehlerliste zeigen, 4-10

X

- X-Taste, 14-24
- XY-Graphen
 - Siehe* Kurven- und XY-Graphen.

Z

- Zählerterminal, 19-3
- Zeiger Hüge (Hinweis), 25-7
- Zeilen, Ändern der Größe, 11-7 bis 11-8
- Zeilenrücklauf, 11-2
- Ziehen und Ablegen, 2-9 bis 2-10

Zielterminals (Eingangsterminals), 17-2
Zoomoptionen, Graphenanzeigeelemente,
15-17 bis 15-19
Zweidimensionale Arrays, 14-3, 14-9, 14-11
Zweig (von Verbindungen), 18-6
Zwischenablage, 2-14
Zyklen, In SubVIs vermeiden, 3-13 bis 3-14
 Attributknoten in Case-Struktur,
 3-13 bis 3-14
 Attributknoten mit Schleifen, 3-12
 Frontpanel-Terminal in Case-Struktur,
 3-13
 lokale Variable in Case-Struktur,
 3-13 bis 3-14
 lokale Variable und Frontpanel-Terminals
 in Schleifen, 3-13
 unlogische Auswahlätze, 3-13