

LabVIEW™ -Upgrade-Hinweise

Version 7.1

In dieser Broschüre wird beschrieben, wie LabVIEW unter Windows, Mac OS und UNIX auf die neueste Version aktualisiert wird, welche Probleme beim Upgrade auftreten können und um welche Funktionen die neue LabVIEW-Version erweitert wurde.

Sollten Sie ein Upgrade von LabVIEW 6.1 oder einer älteren Version auf LabVIEW 7.1 planen, finden Sie in den *LabVIEW 7.0 Upgrade Notes* Informationen über alle wesentlichen Neuerungen der LabVIEW-Umgebung und die Möglichkeiten zur individuellen Gestaltung der LabVIEW-Oberfläche. Generell wird empfohlen, dass alle Nutzer, die bislang mit einer LabVIEW-Version bis einschließlich 6.1 gearbeitet haben, zusätzlich zur vorliegenden Broschüre die *LabVIEW 7.0 Upgrade Notes* zu Rate ziehen. Zum Download der *LabVIEW 7.0 Upgrade Notes* besuchen Sie bitte unsere Website ni.com/info und geben den Info-Code rd70un ein.

Der beste Weg, um sich mit den Neuerungen der LabVIEW-Umgebung in Version 7.0 vertraut zu machen, sind die Übungen im Begleitheft *Erste Schritte mit LabVIEW*.

Weitere Informationen...

Die neuen Funktionen von LabVIEW 7.1 sind in der *LabVIEW-Hilfe* beschrieben. Zum Öffnen der *LabVIEW-Hilfe* wählen Sie **Hilfe»VI-, Funktionen- und Anwendungshilfe**. Alle Handbücher und Application Notes zu LabVIEW sind auch in der *LabVIEW-Bibliothek* zu finden. Zum Öffnen der *LabVIEW-Bibliothek* wählen Sie **Hilfe»Suchen im LabVIEW-Bücherregal**.

Zur Ansicht der PDF-Dateien benötigen Sie den Adobe Acrobat Reader mit Suchfunktion ab Version 5.0.5. (**Mac OS**) Zur Ansicht von PDF-Dateien muss mindestens Adobe Acrobat Reader 6.x mit Suchfunktion installiert sein. Der Acrobat Reader kann auf der Website www.adobe.de heruntergeladen werden.

DataSocket™, DIAdem™, FieldPoint™, HiQ™, IVI™, LabVIEW™, MATRIXx™, National Instruments™, NI™, ni.com™, NI-DAQ™, NI-VISA™ und Xmath™ sind Warenzeichen bzw. Handelsnamen der Firma National Instruments. Alle aufgeführten Produkt- oder Firmennamen sind Warenzeichen oder Handelsnamen der jeweiligen Firmen. Patentinformationen zu Produkten von National Instruments erhalten Sie unter **Hilfe»Patente** in der Software, der Datei `patents.txt` auf Ihrer CD oder über ni.com/patents. Alle urheberrechtlichen Angaben sowie Bedingungen und Haftungsausschlussklauseln zu Komponenten, die in US1 zum Einsatz kommen (Xerxes C++, ICU und HDF5), finden Sie unter `US1Copyrights.chm`.

Juni 2004
322786D-01

Inhaltsverzeichnis

Bekannte Probleme beim Upgrade	3
Konvertierung von VIs	3
Upgrade von Toolkits, Instrumententreibern und Erweiterungspaketen (Add-Ons)	4
Upgrade von früheren LabVIEW-Versionen	5
Upgrade von LabVIEW 7.0	6
Upgrade von LabVIEW 6.x	7
Upgrade von LabVIEW 5.x	12
Upgrade von LabVIEW 4.x	14
Upgrade von LabVIEW 3.x oder älteren Versionen	16
Eigenschaften von LabVIEW 7.1	16
Umgang mit Optionsfeld-Elementen	16
Neuerungen bei den Analyse-VIs	16
Polynom-VIs und VIs für rationale Polynome	17
Palette “Lineare Algebra”	17
Palette “Zeitbereich”	19
Palette “Frequenzbereich”	19
Datenspeicher-VIs	20
Express-VI “Signale anfügen”	20
VIs und Funktionen für Bluetooth	20
Speichern von Graphen, Diagrammen, Tabellen und digitalen Elementen als Grafiken	21
Umgang mit dem Navigationsfenster	21
Puffer-Zuweisungen anzeigen	21
Xmath-Skriptknoten	22
LabVIEW und Hyperthreading	22
Vorbereitung für Beispiel-VIs für die Beispielsuchmaschine	22
Neuerungen an der Beispielsuchmaschine	23
Zeitgesteuerte Schleifen	23
Neuerungen am VI-Server und der Frontpaneldarstellung im Netz	23
Neue Eigenschaften und Methoden	24
LabVIEW Companion Products CD	24
Neuerungen und Änderungen an der Dokumentation	24
Unveränderte Dokumentation	24
Speichern von VIs für ältere Versionen	25
Andere Änderungen und Neuerungen in LabVIEW 7.1	25
Neuerungen beim Application Builder	25
Änderungen an VIs und Funktionen	25
Neue, umbenannte und verschobene VIs	27
Verschiedenes	27

Bekannte Probleme beim Upgrade

Wenn Sie LabVIEW von Version 7.0 auf 7.1 aktualisieren möchten, lesen Sie bitte zuerst die Abschnitte *Konvertierung von VIs*, *Upgrade von Toolkits*, *Instrumententreibern und Erweiterungspaketen (Add-Ons)* und *Upgrade von LabVIEW 7.0*.

Für Upgrades von Version 6.x sollten Sie sich zunächst mit den Abschnitten *Konvertierung von VIs*, *Upgrade von Toolkits*, *Instrumententreibern und Erweiterungspaketen (Add-Ons)* und *Upgrade von LabVIEW 6.x* befassen.

Wenn Sie bisher mit LabVIEW 5.x gearbeitet haben, sollten Sie vor dem Fortfahren zu den Abschnitten *Konvertierung von VIs*, *Upgrade von Toolkits*, *Instrumententreibern und Erweiterungspaketen (Add-Ons)* und *Upgrade von LabVIEW 5.x* springen.

Für Upgrades von LabVIEW 4.x sind zunächst die Abschnitte *Konvertierung von VIs*, *Upgrade von Toolkits*, *Instrumententreibern und Erweiterungspaketen (Add-Ons)* und *Upgrade von LabVIEW 4.x* durchzulesen.

Für Anwender, die mit Version 3.x arbeiten, sind vor der weiteren Lektüre folgende Abschnitte von Bedeutung: *Konvertierung von VIs*, *Upgrade von Toolkits*, *Instrumententreibern und Erweiterungspaketen (Add-Ons)* und *Upgrade von LabVIEW 6.x*.

Konvertierung von VIs

VIs, die zuletzt in einer LabVIEW-Version ab 4.0 gespeichert wurden, werden beim Öffnen in LabVIEW 7.1 automatisch konvertiert und kompiliert. Um zu verhindern, dass das VI bei jedem Öffnen umgewandelt und neu kompiliert wird – wofür zusätzlicher Arbeitsspeicher benötigt wird – sollte das VI am besten in LabVIEW 7.1 gespeichert werden.

VIs, an denen ungespeicherte Änderungen vorgenommen wurden – worunter unter anderem auch eine Neukompilierung zählt – können sich merklich verlangsamen. Weitere Hinweise zu diesem Thema entnehmen Sie bitte auch dem Abschnitt *Memory Issues in Front Panels* der Application Note *LabVIEW Performance and Memory Management*.



Hinweis In LabVIEW 7.1 gespeicherte VIs lassen sich nicht in älteren Versionen öffnen. Wenn Sie also beispielsweise möchten, dass ein VI auch in LabVIEW 7.0 geöffnet werden kann, klicken Sie daher auf **Datei»Mit Optionen speichern** und wählen **Für vorige Version speichern**. Vor dem Speichern in LabVIEW 7.1 empfiehlt es sich, eine Sicherheitskopie der VIs anzulegen, die für die Verwendung in einer älteren Version vorgesehen sind.

Der Speicherbedarf, der zum Konvertieren eines VIs erforderlich ist, lässt sich leicht abschätzen, indem der Speicherplatz zusammengerechnet wird, den das VI und alle dazugehörigen SubVIs auf der Festplatte belegen. Bei VIs in VI-Bibliotheken sind ca. 30 Prozent der Größe der betreffenden Bibliothek hinzuzurechnen, da diese VIs komprimiert sind. Für den Konvertierungsvorgang wird mindestens so viel Speicher benötigt wie errechnet, plus 20 MB, die für die Ausführung von LabVIEW vonnöten sind.

Wenn Ihr Rechner nicht genug Speicherplatz hat, um alle VIs auf einmal zu konvertieren, gehen Sie bei der Umwandlung etappenweise vor. Werfen Sie unter Umständen auch einen Blick auf die Hierarchie der VIs. Laden und speichern Sie zunächst die SubVIs der unteren Hierarchiestufen und fahren Sie in der Hierarchie aufwärts fort. Um VIs verzeichnisweise umzuwandeln, wählen Sie **Werkzeuge»Fortgeschritten» Massenkompilierung**. Wird während des Umwandlungsprozesses zuerst ein Haupt-VI gefunden, wird für die Massenkompilierung in etwa genau so viel Arbeitsspeicher benötigt, als ob dieses VI zuerst geöffnet worden wäre.

Wie viel Arbeitsspeicher eine VI-Konvertierung in Anspruch genommen hat, wird unter dem Menüpunkt **Hilfe»About LabVIEW** über LabVIEW angezeigt.

Upgrade von Toolkits, Instrumententreibern und Erweiterungspaketen (Add-Ons)

Installieren Sie nach Abschluss der Installation von LabVIEW 7.1 alle zu dieser Version kompatiblen Toolkits und Erweiterungspakete in das Programmverzeichnis von LabVIEW 7.1.

Toolkits, Instrumententreiber und Zusatz-VIs aus älteren Versionen müssen für LabVIEW 7.1 ebenfalls massenkompiliert werden. Die Massenkompilierung von VIs ist im Abschnitt *Konvertierung von VIs* dieser Broschüre beschrieben. LabVIEW 7.1 ist zu allen Toolkits, Instrumententreibern und Zusatzpaketen kompatibel, die für eine LabVIEW-Version ab 4.0 erworben wurden.

(Full Development System) Für den LabVIEW Application Builder ist ein separates Upgrade auf den LabVIEW Application Builder 7.1 erforderlich. Im LabVIEW Professional Development System ist der Application Builder 7.1 bereits enthalten. Die Installation des Application Builders wird in den *LabVIEW Benutzerhandbuch Application-Builder* erläutert.

Upgrade von früheren LabVIEW-Versionen

Eine Aktualisierung einer vorhandenen LabVIEW-Version hat keine Auswirkungen auf die bereits vorhandene Version, da die neue Version in ein anderes Verzeichnis installiert wird. So lautet das Programmverzeichnis von LabVIEW-Versionen bis 5.x beispielsweise `labview`. Alle LabVIEW-Versionen ab 6.0 sind in einem Verzeichnis mit dem Namen `labview x.x` gespeichert (`x.x` steht für die jeweilige Versionsnummer).

Um alle Einstellungen zur LabVIEW-Oberfläche von der bisherigen Version zu übernehmen, kopieren Sie die im `labview`-Verzeichnis befindliche Datei mit den Einstellungen in das Programmverzeichnis von LabVIEW 7.1. Alle bis dato vorgenommenen Änderungen an den Grundeinstellungen werden damit in der neuen Version überschrieben.

(Windows) Die Einstellungen sind in der Datei `labview.ini` gespeichert.

(Mac OS) Die Datei mit den Einstellungen lautet `LabVIEW Preferences` und befindet sich im Ordner `Library:Preferences` im Stammverzeichnis.

(UNIX) LabVIEW speichert die Einstellungen unter dem Namen `.labviewrc` im Stammverzeichnis.

Um Dateien aus dem Verzeichnis `user.lib` von vorigen LabVIEW-Versionen zu übernehmen, kopieren Sie die Dateien aus dem `labview`-Verzeichnis in das Verzeichnis `user.lib` von LabVIEW 7.1.

(Windows) Die bisherige LabVIEW-Version kann auch in der Systemsteuerung unter “Software” deinstalliert und durch LabVIEW 7.1 ersetzt werden. Dateien, die im `labview`-Verzeichnis erstellt wurden, bleiben vom Deinstallationsprogramm unberührt.



Vorsicht! Da alle `.llb`-Dateien im Verzeichnis `vi.lib` bei einer Neu- bzw. Deinstallation von LabVIEW entfernt werden, gehen alle VIs und Elemente verloren, die in eine dieser `.llb`-Dateien gespeichert wurden. Speichern Sie Ihre VIs und Elemente daher im Verzeichnis `user.lib`. So werden die VIs auch den Paletten **Funktionen** und **Elemente** hinzugefügt.

Um die vorhandene LabVIEW-Version durch LabVIEW 7.1 zu ersetzen, stellen Sie als Standardverzeichnis für die Installation dasselbe `labview`-Verzeichnis ein, das auch für die vorhandene Version verwendet wurde.

Probleme beim Upgrade und bei der Kompatibilität verschiedener LabVIEW-Versionen werden in den folgenden Abschnitten besprochen.

Upgrade von LabVIEW 7.0

Bei einer Aktualisierung von LabVIEW 7.0 können folgende Probleme auftreten.

Unterstützte Plattformen

Im Hinblick auf die Plattformkompatibilität gibt es bei LabVIEW 7.1 folgende Änderungen:

- LabVIEW 7.1 läuft nicht mehr unter Windows Me/98.
- LabVIEW 7.1 funktioniert nicht mehr unter Mac OS 9.x (oder älteren Versionen) und unterstützt die PPC Toolbox nicht mehr.
- LabVIEW 7.1 läuft auf Solaris-Versionen ab 7.

Umgang mit der Methode “FP öffnen”

Die in LabVIEW 7.0 enthaltene Methode “FP öffnen” wurde auf “FP öffnen - alt” umbenannt, da es in LabVIEW 7.1 eine neue Version dieser Methode gibt. Diese gibt keinen Fehler mehr aus, wenn das Frontpanel bereits geöffnet ist. Wenn die alte Methode in Ihren VIs enthalten ist, ersetzen Sie sie durch die neue oder durch das VI “OpenFP” im Verzeichnis `labview\vi.lib_oldvers_oldvers.lib`.

Aus der Funktionen-Palette entfernte VIs

Folgende VIs werden mit LabVIEW 7.1 nicht mehr installiert:

- **HP34401A Find Range:** Anstelle dieses VIs ist nun “HP34401A Config Measurements” zu verwenden.
- **Zähler für Nullstellen eines reellen Polynoms:** Der Dateiname dieses VIs wurde von “Polynomial Zero Counter.vi” auf “Polynomial Zeros Counter.vi” geändert.
- **PPC-VIs:** Wurden durch die TCP-VIs ersetzt.

Änderungen am VI “Skalarprodukt”

Die VIs zur Berechnung des Skalarprodukts arbeiteten in LabVIEW 7.0 nach folgender Gleichung:

$$X*Y = \sum_{i=0}^{n-1} x_i y_i$$

In LabVIEW 7.1 wird das Skalarprodukt statt dessen durch die folgende Gleichung bestimmt:

$$X*Y = \sum_{i=0}^{n-1} x_i y_i^*$$

wobei y_i^* die Komplex-Konjugierte von y_i ist.

Upgrade von LabVIEW 6.x

Bei einem Upgrade von LabVIEW 6.x auf LabVIEW 7.1 können folgende Probleme auftreten: Informationen über Probleme, die beim Upgrade von LabVIEW auftreten können, finden Sie im Abschnitt [Upgrade von LabVIEW 7.0](#).

Alle Unterschiede, die es zwischen den einzelnen Versionen ab 6.x gab, sind in den *LabVIEW Upgrade Notes* zu jeder Version beschrieben, die Sie unter ni.com/manuals finden.

Unterstützung von Windows 95

LabVIEW 7.1 ist unter Windows 95 nicht lauffähig.

Speichern von Signalverlaufsdaten in einer Datei

Der Datentyp “Signalverlauf” verwendet seit der LabVIEW-Version 7.0 anstelle einer Fließkommazahl mit doppelter Genauigkeit für t_0 den Datentyp “Zeitstempel”. Wenn Sie Daten in diesem Datenformat in LabVIEW 6.x gespeichert haben, ohne die Änderung in LabVIEW 7.0 zu berücksichtigen, tritt beim Öffnen der Daten in LabVIEW 7.1 unter Umständen ein Fehler auf. Was es im Hinblick auf Signalverläufe bei der Umstellung von LabVIEW 6.x auf 7.1 zu beachten gilt, erfahren Sie, wenn Sie auf unserer Website ni.com/info die Zeichenkombination `exd9zq` eingeben.

Unterstützung von HiQ

HiQ gibt es ab einschließlich LabVIEW 7.1 nicht mehr für Mac OS und wird in den Nachfolgerversionen von LabVIEW 7.1 auch nicht mehr für andere Betriebssysteme enthalten sein. Wenn in einer Ihrer Anwendungen HiQ-VIs vorkommen, sollten diese nach den Empfehlungen in der *LabVIEW-Hilfe* durch Mathematik-VIs der Palette “Analyse” ersetzt werden.

Vis für serielle Kompatibilität

Die VIs “Serielle Kompatibilität” befinden sich nicht mehr auf der **Funktionen**-Palette. VIs zur Kommunikation mit VXI-Modulen sollten mit VISA-VIs und -Funktionen erstellt werden.

Zur Kommunikation mit seriellen Schnittstellen wird der `serpdrv`-Treiber nicht mehr benötigt. Die Funktionalität dieses Treibers wird nun durch VISA-VIs realisiert. Verwenden Sie in neuen Anwendungen zur Steuerung von Geräten mit serieller Schnittstelle die VIs und Funktionen der VISA-Palette. Alle VIs, die in Vorgängerversionen von LabVIEW enthalten waren und mit seriellen VIs gearbeitet haben, sind daher auch in LabVIEW 7.1 funktionstüchtig.

Wenn in der ursprünglichen LabVIEW-Anwendung Änderungen an der Zuweisung der Portnummern vorgenommen wurden, ist die geänderte Zuweisung in LabVIEW 7.1 anzugeben. Verwenden Sie dazu das VI “Set Serial Alias” im Verzeichnis `labview\vi.lib\Instr_sersup.lib`. Verbinden Sie ein String-Array mit dem **VISA Aliases**-Eingang des VIs und geben Sie die Portbezeichnungen ein, die im Eingangsarray verwendet werden. Jedes Element im Array sollte einen Port darstellen. Wenn Sie beispielsweise Port 0 das VISA-Alias “MySerialPort” zugewiesen haben, geben Sie in das Array **VISA Aliases** als erstes Element `MySerialPort` ein. Vor dem VI “Set Serial Alias Ports” muss immer zunächst das VI “VISA: Seriellen Port konfigurieren” aufgerufen werden.

Beispiele zur Steuerung von Instrumenten mit serieller Schnittstelle mit Hilfe von VISA-VIs finden Sie unter `examples\instr\smp1ser1.lib`.

Standarddaten in Schleifen

In LabVIEW 6.0 und vorherigen Versionen wurden bei einer For-Schleife undefinierte Werte ausgegeben, wenn die Schleife nicht ausgeführt wurde. In LabVIEW-Versionen ab 6.1 geben For-Schleifen Standardwerte aus, wenn für die Anzahl der Durchläufe 0 angegeben wird, oder wenn ein Eingang mit aktivierter Auto-Indizierung mit einem leeren Array verbunden wird. In diesem Fall wird die Schleife nicht ausgeführt, und für alle nicht auto-indizierten Ausgabetunnel werden die Standardwerte für den Tunnel-Datentyp ausgegeben.

Lizenzen für netzwerkgesteuerte Frontpanel

Das LabVIEW Full Development System und der Application Builder umfassen eine Lizenz, mit der es einem Client möglich ist, ein Frontpanel über ein Netzwerk fernzusteuern. Das LabVIEW Professional Development System wird mit einer solchen Lizenz für fünf Client-Rechner ausgeliefert.

Diese Lizenzen sind auf mehrere Benutzer erweiterbar.

Vergabe mehrerer Ausführungsthreads

In LabVIEW 7.1 werden zur Ausführung von VIs mehr Threads vergeben als in den Vorgängerversionen von LabVIEW 7.0. Daher kann es bei der Verarbeitung mehrerer Threads zu Fehlern kommen, wenn ein Funktionsknoten zum Aufruf externer Bibliotheken fälschlicherweise als ablaufinvariant gekennzeichnet wurde, während die DLL eigentlich nicht ablaufinvariant ist. Weitere Informationen zur Ablaufinvarianz finden Sie im Kapitel 2, *Shared Libraries (DLLs)*, des Handbuchs *Using External Code in LabVIEW*.

Das Verhalten der Zuweisung von Ausführungsthreads in LabVIEW wird mit Hilfe des VIs `threadconfig.vi` in der `vi.lib\Utility\sysinfo.llb` konfiguriert. Die Ablaufinvarianz von VIs lässt sich auch komplett deaktivieren. Entfernen Sie dazu im Dialogfeld **VI-Einstellungen** unter **Ausführung** das Häkchen aus dem Kontrollkästchen **Ablaufinvariante Ausführung**.

Für weitere Informationen zur Zuweisung mehrerer Ausführungsthreads lesen Sie bitte die Application Note *Using LabVIEW to Create Multithreaded VIs for Maximum Performance and Reliability*.

Instrumententreiber

Die LabVIEW-Instrumententreiber-CD ist nicht mehr im Lieferumfang von LabVIEW enthalten. Im National Instruments Instrument Driver Network unter ni.com/idnet stehen die Treiber jedoch weiterhin zum Download zur Verfügung. Auf der Gerätetreiber-CD von National Instruments sind unter anderem NI-DAQ und NI-VISA enthalten.

Einheiten und Umrechnungsfaktoren

Nach Verwendung der Funktion “Mehrfacharithmetik” ist nicht mehr, wie bisher, die Funktion “Einheit konvertieren” aufzurufen, um die überzählige Einheit zu entfernen.

Die Umrechnungsfaktoren für Einheiten wurden in LabVIEW 7.1 den Richtlinien des vom National Institute for Standards and Technology (NIST) veröffentlichten *Guide for the Use of the International System of Units (SI)* weiter angepasst. Außerdem wurde die Kalorie nun in Kalorie (thermisch) und die Pferdestärke in Leistung (elektrisch) umbenannt. Die Abkürzungen für diese Einheiten haben sich nicht geändert. Nachfolgend alle Änderungen an den Umrechnungsfaktoren für Einheiten zwischen LabVIEW 6.1 und 7.1.

Einheit	Umrechnungsfaktor or in 6.1	Umrechnungsfaktor in 7.1
Astronomische Einheit (AU)	149.498.845.000 m	149.597.900.000 m
British Thermal Unit (Mittelwert)	1055,79 J	1055,87 J
Elektronenvolt (eV)	1,602e-19 J	1,60217642e-19 J
Foot-Candle	10,764 lx	10,7639 lx
Pferdestärke – Leistung (elektrisch)	745,7 W	746 W Die neue Umrechnung ist exakt.
Gallone (imperial)	4,54596 l	4,54609 l
Lichtjahr	9,4605 Pm	9,46073 Pm
Pound Force	4,448 N	4,448222 N
Rod	502,92 cm	5,029210 m
Slug	14,594 kg	14,59390 kg
Unitäre atomare Masseneinheit (u)	1,66057e-27 kg	1,66053873e-27 kg

Eigenschaft “Panel-Aktualisierungen verzögern”

Wenn diese Eigenschaft den Wert TRUE hat, werden alle Frontpanel-Objekte mit noch ausstehenden Änderungen neu gezeichnet, und die nächste Aktualisierung der Objekte wird um eine bestimmte Zeit verschoben. Bei LabVIEW 6.1 und Vorgängerversionen werden Elemente, die sich inzwischen geändert haben, erst wieder aktualisiert, wenn an die Eigenschaft der Wert FALSE übermittelt wird.

In einigen Fällen nimmt die Neudarstellung der Elemente zusätzliche Zeit in Anspruch.

Datenbereiche numerischer Bedienelemente

In LabVIEW 6.1 und Vorgängerversionen wurden einige numerische Elemente standardmäßig auf einen Mindestwert von 0,00 und einen Höchstwert von 0,00 gesetzt, der Anzeigewert erhöhte sich bei einem Klick auf die Pfeilschaltfläche des Elements um 0,00, und die Standardeinstellung bei einer Bereichsüberschreitung lautete **Ignorieren**.

In LabVIEW 7.1 werden für diese Elemente die Werte des Standardwertebereichs für den jeweiligen Datentyp verwendet.

Typumwandlungspunkte und -definitionen

In den LabVIEW-Versionen ab 6.1 sind in den Verbindungen Angaben über Typdefinitionen enthalten, so dass im Blockdiagramm mehr Typumwandlungspunkte vorliegen können. Wenn Sie ein Typ-Definitionselement mit einem VI oder einem Knoten verbinden, der kein Typ-Definitionsknoten ist, zeigt LabVIEW einen Typumwandlungspunkt an. Typumwandlungspunkte werden auch angezeigt, wenn ein Ausgangsanschluss, der eine Typdefinition hat, mit einer Anzeige verbunden wird, die keine Typdefinition hat. Die Typumwandlungspunkte zeigen an, wenn die Typdefinitionen in einem VI nicht übereinstimmen.

Die Leistung des VIs wird davon in diesem Fall nicht beeinträchtigt.

Um Typdefinitionen in Strings zu konvertieren, dient das VI "Daten in String konvertieren", über das Sie in der *LabVIEW-Hilfe* nähere Einzelheiten finden.

Schaltflächenbeschriftungen im Dialogfeld "Datei"

In LabVIEW 6.1 und Vorgängerversionen wechselt die Beschriftung der Schaltfläche **Öffnen** auf **Speichern**, wenn der Benutzer einen neuen Dateinamen eingeben kann. In LabVIEW 7.1 lautet die Beschriftung für beide Fälle standardmäßig **OK**. Die Beschriftung lässt sich mit Hilfe des Eingangs **Tastenbeschriftung** der Funktion "Dateidialog" verändern. Wenn die Funktion in einem vorhandenen VI vorkommt, überprüfen Sie das VI, um sicherzustellen, dass die neue Standardbeschriftung **OK** dem Einsatzzweck Ihres VIs gerecht wird.

Die Funktion "Online-Hilfe steuern"

Der Eingang **Pfad zur Hilfedatei** der Funktion "Online-Hilfe steuern" darf nicht frei bleiben. An den Anschluss kann entweder ein Dateiname mit der Erweiterung `.chm` oder `.hlp` übergeben werden oder der entsprechende vollständige Pfad. Wenn Sie nur den Namen einer kompilierten Hilfedatei angeben haben, sucht LabVIEW im Verzeichnis `labview\help` nach der Datei.

Methode "VI ausführen"

Wenn Sie den Parameter **Ref automatisch schließen** der Methode "VI ausführen" auf TRUE setzen, wird die Referenz auch dann geschlossen, wenn die Methode einen Fehler ausgibt. Dadurch kann ein VI, das auf die Referenz angewiesen ist, nicht ausführbar werden.

Anzeige des Frontpanels nach dem Laden

Wenn Sie in LabVIEW 7.1 ein VI so einstellen, dass das Frontpanel beim Laden des VIs sofort sichtbar ist, wird das Frontpanel beim Laden durch den VI-Server nicht angezeigt. In diesem Fall muss das Frontpanel mit Hilfe der Methode “FP öffnen” programmatisch geöffnet werden.

Funktion “VI-Referenz öffnen”

Wenn Sie mit Hilfe der Funktion “VI-Referenz öffnen” eine Referenz auf eine Vorlage erstellen, die sich bereits im Arbeitsspeicher befindet, gibt die Funktion einen Fehler aus.

Exponentialdarstellung

In LabVIEW-Versionen vor 6.1 wurde zur Exponentialschreibweise in einem Formelknoten das Symbol \wedge verwendet. In den LabVIEW-Versionen ab 6.1 ist eine Potenzierung mit dem Symbol $**$ gekennzeichnet (zum Beispiel $x**y$). Das Symbol \wedge steht nun für das bitweise Exklusiv-ODER (XOR).

IVI-Konfigurationsdateiformat

Bei Dateien zur Speicherung von IVI-Konfigurationen wird nun zwischen Groß- und Kleinschreibung unterschieden. Wenn in Ihrem Programm mit logischen, virtuellen oder Treiber-Session-Namen gearbeitet wird, sollten Sie sich vergewissern, dass der in der IVI-Konfigurationsdatei verwendete Name mit dem angegebenen Namen übereinstimmt.

Formular für technische Unterstützung

Die `techsup.llb` wird bei der Installation von LabVIEW nicht installiert. Bei Problemen oder Fragen bezüglich der Installation, Konfiguration oder Ausführung von Anwendungen besuchen Sie bitte unsere Website ni.com/support.

Upgrade von LabVIEW 5.x

Bei einem Upgrade von LabVIEW 5.x auf LabVIEW 7.1 können folgende Probleme auftreten: Alle anderen Probleme, die beim Upgrade von LabVIEW auftreten können, sind in den Abschnitten [Upgrade von LabVIEW 6.x](#) und [Upgrade von LabVIEW 7.0](#) beschrieben.

Alle Unterschiede zwischen LabVIEW 5.x und 7.1 sowie alle Unterschiede zur *Beilage zum Handbuch für LabVIEW 5.1* sind in den *Upgrade Notes* zu jeder Version aufgeführt, die unter ni.com/manuals zu finden sind.

Konvertierung von Datenprotokolldateien

In LabVIEW 7.1 werden die Typdefinitionen von Datenprotokolldateien überprüft, um festzustellen, ob eine Konvertierung der jeweiligen Datei erforderlich ist. Wenn die Datei mit einer älteren Version als LabVIEW 6.0 erstellt wurde oder der Datentyp "Signalverlauf" darin enthalten ist, wird die Datei vor dem Auslesen oder Anhängen von Daten umgewandelt. In allen anderen Fällen werden die Dateien beim Auslesen oder Anfügen von Daten nicht konvertiert.

Beim Öffnen von Datenprotokolldateien in LabVIEW 7.1, die in älteren Versionen des Programms erstellt wurden, werden Sie gefragt, ob die jeweilige Datei in das LabVIEW-7.1-Format umgewandelt werden soll. Wenn Sie mit der Umwandlung einverstanden sind, wird die vorhandene Datei mit einer neuen Datei im richtigen Format ersetzt. Wenn die Datei nicht umformatiert werden soll, gibt LabVIEW einen Fehler aus und die Datei wird nicht geöffnet.



Hinweis Wenn Sie Datenprotokolldateien aus LabVIEW-Versionen vor 7.0 weiter in diesen Versionen nutzen möchten, sollten Sie am besten eine Sicherheitskopie der Dateien anlegen. Eine einmal vorgenommene Umformatierung kann nicht rückgängig gemacht werden, so dass die betreffende Datei für LabVIEW 6.1 oder eine Vorgängerversion davon verloren wäre.

Sollen Datenprotokolldateien beim Öffnen automatisch neu formatiert werden, fügen Sie in die Datei mit den LabVIEW-Einstellungen folgende Zeile ein:

```
silentDatalogConvert=True
```

(Mac OS) Fügen Sie folgende Zeile ein:

```
silentDatalogConvert:True
```

(UNIX) Fügen Sie folgende Zeile ein:

```
labview.silentDatalogConvert:True
```

Wenn die Dateien nicht automatisch umformatiert werden sollen, ersetzen Sie das `True` durch `False`.

Inkompatibilitäten zwischen dem LabVIEW-5.x-VI-Server und LabVIEW-7.1-Clients

Vom einem LabVIEW-7.1-Client lässt sich keine Verbindung zum VI-Server von LabVIEW 5.x herstellen, da die in LabVIEW 5.x erstellte Applikation bestimmte Aspekte des LabVIEW-7.1-Serverprotokolls nicht erkennt.

Es ist jedoch problemlos möglich, eine Verbindung zwischen einem LabVIEW-5.x-Client und dem VI Server einer LabVIEW-7.1-Anwendung herzustellen.

UDP-Funktionen

Zur Kommunikation über Netzwerke gibt es nun in LabVIEW UDP-Funktionen. Die früher statt dessen verwendeten UDP-VIs sind allerdings aus Kompatibilitätsgründen weiter in LabVIEW enthalten. Zu finden sind die VIs unter `vi.lib\oldvers\oldvers.llb`.

Upgrade von LabVIEW 4.x

Bei einem Upgrade von LabVIEW 4.x auf LabVIEW 7.1 können die in den nachfolgenden Abschnitten beschriebenen Probleme auftreten. Weitere Probleme, die beim Upgrade von LabVIEW auftreten können, sind in den Abschnitten [Upgrade von LabVIEW 5.x](#), [Upgrade von LabVIEW 6.x](#) und [Upgrade von LabVIEW 7.0](#) beschrieben.

Alle Unterschiede zwischen LabVIEW 4.x und 7.1 sowie alle Unterschiede zur *Beilage zum Handbuch für LabVIEW 5.1* sind in den *Aktualisierungshinweisen* zu jeder Version aufgeführt, die unter ni.com/manuals zu finden sind.

Umwandlung boolescher Werte für LabVIEW 4.x

Das Format zur Speicherung boolescher Werte hat sich von LabVIEW 4.x auf LabVIEW 5.x geändert. In LabVIEW 4.x werden boolesche Werte in zwei Bytes gespeichert, sofern sie nicht zu einem Array gehören. In diesem Fall speichert LabVIEW 4.x jeden booleschen Wert in einem einzelnen Bit. Von LabVIEW-Version 7.1 an aufwärts wird ein boolescher Wert in einem einzigen Byte gespeichert, und zwar unabhängig davon, ob dieser Wert Teil eines Arrays ist oder nicht. Durch diese Änderung können mehr Funktionen mit booleschen Werten arbeiten und Arrays boolescher Werte sind in ihrem Verhalten Arrays numerischer Werte ähnlicher. Die Änderung im Speicherformat boolescher Werte wirkt sich in LabVIEW 7.1 auf die Datenmanipulation in Code-Interface-Knoten aus, verhält sich allerdings in LabVIEW 7.1 bei existierenden CINs kompatibel.

Wenn in LabVIEW 4.x Binärdaten in eine Datei gespeichert werden, unter denen sich auch boolesche Werte befinden, ist das Format nicht dasselbe wie beim Speichern der Datei in LabVIEW 7.1. In LabVIEW 7.1 gibt es allerdings einen Mechanismus, mit dem in LabVIEW 4.x gespeicherte Binärdaten ausgelesen bzw. so gespeichert werden können, dass sie mit LabVIEW 4.x kompatibel sind. Bei den Funktionen "Datei schreiben", "Datei lesen", "Typenformung", "In String konvertieren" und "String in Daten konvertieren" gibt es dazu im Kontextmenü einen Menüpunkt mit der Bezeichnung **4.x-Daten konvertieren**. Wenn Sie die Funktion also mit

der rechten Maustaste anklicken und diesen Menüpunkt auswählen, wird mit den Daten so verfahren, als ob in LabVIEW 4.x gearbeitet würde. Verwenden Sie zum Erzeugen von Daten im LabVIEW 4.x-Format die Funktionen "Datei schreiben", "In String formatieren" und Typenformung, und zum Auslesen von Daten im LabVIEW 4.x-Format die Funktionen "Datei lesen", "String in Daten konvertieren" und "Typenformung". Wenn Sie die Menüoption **4.x-Daten konvertieren** auswählen, wird die Funktion mit einem roten 4.x gekennzeichnet, um anzuzeigen, dass die Daten nicht im üblichen LabVIEW-7.1-Format vorliegen. Sollen die Daten nicht (mehr) umgewandelt werden, klicken Sie die Menüoption **4.x-Daten konvertieren** erneut an, so dass das Häkchen daneben verschwindet.

Bei mehreren Dateien mit booleschen Werten kann auch ein VI erstellt werden, mit dem die Dateien in einem Format geöffnet und gespeichert werden, das LabVIEW 7.1 erkennt.

Wenn in LabVIEW 7.1 ein VI geladen wird, das zuletzt in LabVIEW 4.x oder einer älteren Version gespeichert wurde, wird bei den Funktionen "Datei schreiben", "Datei lesen", "Typenformung", "In String konvertieren" und "String in Daten konvertieren" die Einstellung **4.x-Daten konvertieren** automatisch aktiviert. Die Funktionen arbeiten genau wie zuvor. Wenn ein VI, das mit einer der genannten Funktionen verbunden ist, mit dem LabVIEW-7.1-Format für boolesche Werte arbeiten soll, deaktivieren Sie im Kontextmenü den Menüpunkt **4.x-Daten konvertieren**. Wenn VIs weder mit Dateien arbeiten, die boolesche Werte im LabVIEW-4.x- oder einem älteren Format enthalten, noch boolesche Werte mit VIs austauschen, die in LabVIEW 4.x oder einer Vorgängerversion davon ausgeführt werden, empfiehlt sich in der Regel die Verwendung des LabVIEW-7.1-Formats. Das alte Format wird unter Umständen in einer der nächsten Versionen abgeschafft.

VI-Steuerungs-VIs

Die VIs zur VI-Steuerung befinden sich nicht auf der **Funktionen**-Palette, sind jedoch aus Kompatibilitätsgründen weiterhin in der `vi.lib\utility\victl.lib` enthalten. Verwenden Sie anstelle der VIs zur VI-Steuerung bitte jetzt die VI-Server-Funktionen "VI-Referenz öffnen", "Aufruf über Referenz" sowie die Eigenschaftsknoten und Methodenknotten.

Auch einige Fehlercodes, die die VIs zur VI-Steuerung in LabVIEW 7.1 ausgeben, haben sich geändert. Während die VIs bislang einen Fehlercode zwischen 7 und 1000 ausgaben, arbeiten sie nun mit den Codes 1004 und 1003. Wenn ein mit LabVIEW 4.x erstelltes VI auf einen Fehlercode zwischen 7 und 1000 wartet, muss das VI für LabVIEW 7.1 umgeschrieben werden.

DDE-VIs

(**Windows**) Die DDE-VIs werden nicht auf der **Funktionen**-Palette angezeigt, sind jedoch aus Kompatibilitätsgründen weiterhin in der `vi.lib\platform\dde.llb` enthalten.

Upgrade von LabVIEW 3.x oder älteren Versionen

Zum Upgrade von LabVIEW 3.x oder einer älteren Version auf die neueste Version von LabVIEW gibt es ein VI Conversion Kit, über das Sie auf unsere Website ni.com/support nähere Auskunft erhalten. Weitere Probleme, die beim Upgrade von LabVIEW auftreten können, sind in den Abschnitten [Upgrade von LabVIEW 4.x](#), [Upgrade von LabVIEW 5.x](#), [Upgrade von LabVIEW 6.x](#) und [Upgrade von LabVIEW 7.0](#) beschrieben.

Alle Unterschiede zwischen LabVIEW 4.x und 7.1 sowie alle Unterschiede zur *Beilage zum Handbuch für LabVIEW 5.1* sind in den *Aktualisierungshinweisen* zu jeder Version aufgeführt, die unter ni.com/manuals zu finden sind.

Eigenschaften von LabVIEW 7.1

Die neuen Funktionen von LabVIEW 7.1 sind in der *LabVIEW-Hilfe* beschrieben.

Umgang mit Optionsfeld-Elementen

Mit den Optionsfeld-Elementen können Sie ein Dialogfeld mit mehreren Optionen erstellen, aus denen der Benutzer eine auswählen kann. Wenn keine Option vorausgewählt sein soll, klicken Sie das Optionsfeld mit der rechten Maustaste an und aktivieren Sie den Menüpunkt **Keine Auswahl zulassen**. Da Optionsfeld-Elemente mit Enums arbeiten, lassen sich mit solchen Elementen beispielsweise verschiedene Cases einer Case-Struktur ansteuern.

Beispiele für die Anwendung des Optionsfeld-Elements sind die VIs “Radio Buttons Control” und “Radio Buttons with Event Structure” unter `labview\examples\general\controls\booleans.llb`.

Neuerungen bei den Analyse-VIs

Die Analyse-VIs in LabVIEW wurden in folgender Weise verbessert:

- Einsatz des BLAS/LAPAK-Algorithmus', mit dem die Rechenzeit für Gleichungen der linearen Algebra vermindert wurde
- Zwei neue Paletten zur Berechnung von Polynomfunktionen

- Polymorphe VIs zur linearen Algebra und zum Frequenzbereich
- Neue VIs zur linearen Algebra und zum Zeitbereich

Polynom-VIs und VIs für rationale Polynome

Verwenden Sie die Polynom-VIs, um Berechnungen und Entwicklungen mit Polynomen durchzuführen. Verwenden Sie die VIs für rationale Polynome, um Berechnungen und Entwicklungen mit rationalen Polynomen durchzuführen.

Palette “Lineare Algebra”

Die Unterpalletten **Fortgeschrittene lineare Algebra**, **Komplexe lineare Algebra** und **Fortgeschrittene komplexe lineare Algebra** sind nicht mehr in der Palette **Linear Algebra** enthalten. Die VIs, die sich vorher auf diesen Paletten befanden, sind nun polymorphe Instanzen der VIs auf der Palette **Lineare Algebra**. In der folgenden Tabelle sehen Sie, wo sich alle VIs aus den Unterpalletten jetzt befinden.

LabVIEW-7.0-VI	LabVIEW-7.1-VI	Polymorphe Instanz
A x Vektor	A x B	A x Vektor
A x B (komplex)	A x B	A x B (komplex)
A x Vektor (komplex)	A x B	A x Vektor (komplex)
Komplexe Cholesky-Faktorisierung	Cholesky-Faktorisierung	Komplexe Cholesky-Faktorisierung
Transponierte und konjugiert-komplexe Matrix	Transponierte Matrix	Transponierte und konjugiert-komplexe Matrix
Komplexe Determinante	Determinante	Komplexe Determinante
Komplexes skalares Produkt	Skalarprodukt	Komplexes skalares Produkt
Komplexe Eigenwerte und -vektoren	Eigenwerte und -vektoren	Komplexe Eigenwerte und -vektoren
Komplexe inverse Matrix	Inverse Matrix	Komplexe inverse Matrix
Komplexe LU-Faktorisierung	LU-Faktorisierung	Komplexe LU-Faktorisierung
Zustandswert der komplexen Matrix	Zustandswert der Matrix	Zustandswert der komplexen Matrix

LabVIEW-7.0-VI	LabVIEW-7.1-VI	Polymorphe Instanz
Norm der komplexen Matrix	Norm der Matrix	Norm der komplexen Matrix
Rang der komplexen Matrix	Rang der Matrix	Rang der komplexen Matrix
Hauptdiagonale komplexer Matrizen	Hauptdiagonale der Matrix	Hauptdiagonale komplexer Matrizen
Komplexe pseudoinverse Matrix	Pseudoinverse Matrix	Komplexe pseudoinverse Matrix
Komplexes äußeres Produkt	Äußeres Produkt	Komplexes äußeres Produkt
Komplexe QR-Faktorisierung	QR-Faktorisierung	Komplexe QR-Faktorisierung
Spezielle komplexe Matrix definieren	Spezielle komplexe Matrix definieren	Spezielle komplexe Matrix definieren
Komplexe lineare Gleichungen lösen	Lineare Gleichungen lösen	Komplexe lineare Gleichungen lösen (eine rechte Seite), Komplexe lineare Gleichungen lösen (mehrere rechte Seiten)
SVD-Faktorisierung	Singulärwertzerlegung	Singulärwertzerlegung (reell)
Testen auf komplex positiv definit	Testen auf positiv definit	Testen auf komplex positiv definit

Folgende VIs sind zur Palette **Lineare Algebra** hinzugekommen:

- “Eigenvektoren für Rücktransformation”: Transformiert Eigenvektoren einer abgeglichenen reellen oder komplexen Matrix in die Eigenvektoren der ursprünglichen Matrix zurück.
- “Reelle Matrix aus Eigenwerten erzeugen”: Erzeugt aus einer Reihe von Eigenwerten eine reelle Matrix.
- “Reelle verallgemeinerte Eigenwerte und Vektoren”: Berechnet die verallgemeinerten rechten Eigenwerte und -vektoren eines Matrizenpaars mit reellen oder komplexen Werten.
- “Hessenberg-Zerlegung”: Führt die Hessenberg-Zerlegung einer reellen oder komplexen Matrix durch.
- “Matrix abgleichen”: Gleicht zur Verbesserung der Genauigkeit der berechneten Eigenwerte und Eigenvektoren eine allgemeine reelle oder komplexe Matrix ab.
- “Charakteristisches Polynom der reellen Matrix”: Berechnet das charakteristische Polynom einer reellen Matrix.

- “e hoch Matrix”: Berechnet die Exponentialfunktion einer reellen oder komplexen Matrix.
- “Logarithmus einer Matrix”: Berechnet den natürlichen Logarithmus einer reellen oder komplexen Matrix.
- “Matrix-Potenzierung”: Berechnet die n -te Potenz einer reellen oder komplexen Matrix.
- “Quadratwurzel aus Matrix”: Berechnet die Quadratwurzel einer reellen oder komplexen Matrix.
- “QZ-Zerlegung”: Führt die QZ-Zerlegung eines quadratischen Matrizenpaars mit reellen oder komplexen Werten durch.
- “Schur-Zerlegung”: Führt die Schur-Zerlegung einer reellen oder komplexen quadratischen Matrix durch.
- “Singularwertzerlegung”: Führt die Singularwertzerlegung einer $[m,n]$ -Matrix mit reellen oder komplexen Werten durch.
- “Sylvester-Gleichungen”: Dient zur Lösung der Sylvester-Gleichung für eine reelle oder komplexe Matrix.
- “Transponierte Matrix”: Dient zum Transponieren einer reellen oder komplexen Matrix.

Palette “Zeitbereich”

Auf der Palette **Zeitbereich** gibt es folgende neue VIs:

- “Neuaufnahme (Konstante nach Konstante)”: Tastet ein Eingangssignal oder mehrere Eingangssignale entsprechend des eingestellten Abtastintervalls und Zeitstempels ab.
- “Neuaufnahme (Konstante nach Variable)”: Tastet ein Eingangssignal oder mehrere Eingangssignale entsprechend der angegebenen Zeitinstanz ab.

Palette “Frequenzbereich”

Während die Palette **Frequenzbereich** in den Versionen vor LabVIEW 7.1 die VIs “Reelle FFT”, “Inverse reelle FFT”, “Komplexe FFT” und “Inverse komplexe FFT” enthielt, befinden sich auf der Palette nun die polymorphen VIs “FFT” und “Inverse FFT”, die jeweils komplexe oder reelle Werte verarbeiten können.

Datenspeicher-VIs

(Windows) Die Datenspeicher-VIs dienen dazu, Signalverlaufsdaten in eine Datei im NI-Format zum Testdatenaustausch (.tdm-Datei) zu speichern bzw. aus einer solchen Datei zu laden. Mit .tdm-Dateien können Daten zwischen NI-Programmen wie LabVIEW und DIAdem ausgetauscht werden.

Bei den Datenspeicher-VIs werden Kanäle für Signalverläufe und deren Eigenschaften erzeugt. Eine Kanalgruppe besteht aus einer Reihe von Kanälen. In einer Datei ist eine Reihe von Kanalgruppen enthalten. Neben numerischen Werten können die Datenspeicher-VIs auch Arrays aus Strings bzw. aus Zeitstempeln verarbeiten. Im Blockdiagramm sind die Dateien, Kanalgruppen und Kanäle durch eine Referenznummer dargestellt.

Des Weiteren lassen sich mit den Datenspeicher-VIs Kanalgruppen oder Kanäle herausfiltern, die speziellen Bedingungen entsprechen.

Beispiele zur Verwendung der Datenspeicher-VIs sind in der Bibliothek `labview\examples\file\storage.llb` zu finden.

Express-VI “Signale anfügen”

Mit dem Express-VI “Signale anfügen” werden Signale zusammengesetzt. Ein Beispiel zur Verwendung des Express-VIs “Signale anfügen” befindet sich im Verzeichnis `examples\express`.

VIs und Funktionen für Bluetooth

(Windows) Die VIs und Funktionen für Bluetooth dienen zur Kommunikation mit Geräten, die das Bluetooth-Kommunikationsprotokoll verwenden. In LabVIEW können nur solche Bluetooth-Geräte genutzt werden, die unter Windows XP (ab Service Pack 1) mit dem Microsoft Bluetooth-Treiber arbeiten. Für welche Geräte sich der Microsoft Bluetooth-Treiber eignet, erfahren Sie im Windows-Katalog unter www.microsoft.com. Die meisten Bluetooth-Geräte werden bereits mit einem passenden Treiber ausgeliefert. Für die Verwendung in LabVIEW wird allerdings die Microsoft-Bluetooth-Treiber-CD benötigt, die Sie beim Hersteller Ihres Gerätes erhalten.

In der Application Note *Using LabVIEW with Wireless Devices* finden Sie weitere Informationen zur drahtlosen Kommunikation mit Hilfe der Bluetooth-Technologie in LabVIEW. Einen ausführlichen Einblick in die Bluetooth-Technologie bietet die Website www.bluetooth.com. Anwendungsbeispiele für die Bluetooth-VIs und -Funktionen in LabVIEW finden Sie unter `labview\examples\comm\Bluetooth.llb`.

Speichern von Graphen, Diagrammen, Tabellen und digitalen Elementen als Grafiken

Zum Abspeichern eines Graphen, Diagramms, einer Tabelle oder eines digitalen Bedien- oder Anzeigeelements als .emf oder .bmp-Datei klicken Sie das Objekt mit der rechten Maustaste an und wählen Sie den Menüpunkt **Datenoperationen»Vereinfachtes Bild exportieren** aus.

Umgang mit dem Navigationsfenster

Das **Navigationsfenster** ermöglicht einen Überblick über das aktive Frontpanel im Bearbeitungsmodus oder das aktive Blockdiagramm. Es dient dazu, große Frontpanel bzw. Blockdiagramme besser zu überblicken und darin zu navigieren. Um zu einem Bereich des Frontpanels oder Blockdiagramms zu springen, klicken Sie einfach auf den entsprechenden Bereich im Navigationsfenster. Die Ansicht im Navigationsfenster kann verändert werden, indem das Bild im Navigationsfenster angeklickt und der Mauszeiger in die gewünschte Richtung gezogen wird. Die Bereiche des Frontpanels oder Blockdiagramms, die aktuell nicht sichtbar sind, werden im Navigationsfenster grau dargestellt.

Klicken Sie zum Öffnen des Navigationsfensters auf **Fenster» Navigationsfenster anzeigen** oder drücken Sie die Tastenkombination <Strg-Shift-N>. (**Mac OS**) Drücken Sie <Befehl-Shift-N>. (**Sun**) Drücken Sie <Meta-Shift-N>. (**Linux**) Drücken Sie <Alt-Shift-N>.



Hinweis Das **Navigationsfenster** gibt es nur im LabVIEW Full bzw. Professional Development System.

Puffer-Zuweisungen anzeigen

Um dieses Fenster anzuzeigen, wählen Sie **Werkzeuge»Fortgeschritten» Puffer-Zuweisungen anzeigen**. Aktivieren Sie die Datentypen, deren Puffer angezeigt werden sollen, und klicken Sie auf die Schaltfläche **Aktualisieren**. An den Stellen, an denen LabVIEW Puffer zur Speicherung von Daten erzeugt, erscheinen auf dem Blockdiagramm schwarze Quadrate.



Hinweis Das Dialogfeld **Puffer-Zuweisungen anzeigen** gibt es nur im LabVIEW Full bzw. Professional Development System.

Zu wissen, wo LabVIEW Puffer erzeugt, ermöglicht Ihnen eventuell, die Ausführungsgeschwindigkeit des VIs zu steigern, indem Sie das VI so bearbeiten, dass es weniger Speicher benötigt. Im Abschnitt *VI Memory Usage* der Application Note *LabVIEW Performance and Memory Management* finden Sie Informationen darüber, wie LabVIEW Speicher verwendet, sowie Tipps für eine effiziente Speichernutzung.

Bei Änderungen an einem VI, die eine Neukompilierung erforderlich machen, werden die schwarzen Quadrate ausgeblendet, da die Angaben zum Puffer unter Umständen nicht mehr stimmen. Mit Hilfe der Schaltfläche **Aktualisieren** im Dialogfeld **Puffer-Zuweisung anzeigen** wird das VI neu kompiliert und die Symbole werden wieder angezeigt. Nach dem Schließen des Dialogfelds sollten keine Quadrate mehr angezeigt werden.

Xmath-Skriptknoten

Xmath-Skriptknoten dienen zur Ausführung externer Xmath-Skripte. Sie sind jedoch nur verfügbar, wenn Xmath installiert ist.

Das Programm Xmath gehört zur Produktgruppe von NI MATRIXx, mit der skriptgesteuerte Anwendungen zur mathematischen Analyse und Visualisierung erstellt werden können. Xmath ist eine Arbeitsumgebung zum Erstellen von Skripten, zur spezialisierten Datenanalyse und zur Darstellung von Simulationsdaten. Es enthält eine Bibliothek von Funktionen zur Mathematik, Systemmodellierung und Analyse. Xmath und NI MATRIXx sind unter ni.com/matrixx näher beschrieben.

Zur Verwendung des Xmath-Skriptknotens in LabVIEW finden Sie in der Bibliothek `labview\examples\scriptnode\Differential Equation.llb` das Beispiel-VI "Xmath Script - Lorenz Diff Eq" und unter `labview\examples\scriptnode\Fractal.llb` das Beispiel "Xmath Script - Fractal".

LabVIEW und Hyperthreading

Hyperthreading ist eine erweiterte Funktion von Intel-Pentium-4-Ausführungen der oberen Leistungsklasse. Ein Hyperthreading-fähiger Computer hat nur einen Prozessor, arbeitet aber wie ein Mehrprozessorcomputer. In textbasierten Programmiersprachen werden Multithreading-Anwendungen erstellt, indem mehrere Threads erzeugt und mit Hilfe des Programmcodes koordiniert werden. LabVIEW erkennt dagegen automatisch, wenn in einem VI die Möglichkeit zum Multithreading besteht, und übernimmt alle Funktionen, die zur Koordination der einzelnen Threads erforderlich sind. Wie LabVIEW Programmcode auf einem hyperthreadingfähigen Computer ausführt, wird in der Application Note *LabVIEW and Hyper-Threading* erläutert.

Vorbereitung für Beispiel-VIs für die Beispielsuchmaschine

Wählen Sie **Werkzeuge»Beispiel-VIs für die Suchmaschine für Beispiele vorbereiten**, wenn Sie möchten, dass ein selbst erstelltes Beispiel-VI mit der Suchmaschine für Beispiele gesucht werden kann. Hier kann eine Beschreibung zum VI eingegeben und eine `.bin3`-Datei mit den

Suchkategorien, -stichwörtern und anderen Angaben zum VI erstellt werden. Die `.bin3`-Datei wird benötigt, um Beispiel-VIs in der Suchmaschine anzuzeigen.



Hinweis Die von Ihnen erstellten Beispiel-VIs und die `.bin3`-Datei müssen sich im Verzeichnis `examples`, `instr.lib` oder `user.lib` befinden. Die Verzeichnisstruktur der Beispiel-VIs muss mit der übereinstimmen, in der sie auf dem Computer des Benutzers gespeichert werden sollen. Wenn sich die `.bin3`-Datei mehr als eine Verzeichnisebene unter `examples`, `instr.lib` oder `user.lib` befindet, erkennt die Suchmaschine die Datei nicht.

Neuerungen an der Beispielsuchmaschine

In der Beispielsuchmaschine gibt es nun auf der Registerkarte **Suchen** einen Ordner mit dem Titel **Aktuell** und einen **Favoriten**-Ordner, mit deren Hilfe Sie die am häufigsten genutzten VIs leichter finden. Außerdem können VIs gesucht werden, die nur mit ausgewählten Geräten arbeiten. Weitere Informationen zur Suche nach VIs, die mit bestimmten Geräten arbeiten, finden Sie in der *Hilfe zur NI-Suchmaschine für Beispiele*. Denken Sie bitte daran, bei jedem Start der Suchmaschine oder bei jedem Hinzufügen und Entfernen von Geräten im Pulldown-Menü **Hardware** die Option **Hardware-Liste aktualisieren** auszuwählen.

Zeitgesteuerte Schleifen

(Windows 2000/XP) Eine zeitgesteuerte Schleife wird mit einer angegebenen Periode ausgeführt. Getaktete Schleifen sind bei VIs mit unterschiedlichen Taktraten, bei sich dynamisch ändernden Taktraten, bei mehreren Prioritätsstufen von Vorteil oder wenn eine hohe Genauigkeit bei der Einhaltung eines Ausführungszeitrahmens gefragt ist. Alles Weitere zu getakteten Schleifen entnehmen Sie bitte der Application Note *Using the Timed Loop to Write Multirate Applications in LabVIEW*. Mit Hilfe einer getakteten Schleife und den DAQmx-VIs und -Funktionen lassen sich VIs zur Steuerung von Taktgebern erstellen.

Anwendungsbeispiele für getaktete Schleifen in VIs finden Sie unter `labview\examples\general\timedloop.llb`.

Neuerungen am VI-Server und der Frontpaneldarstellung im Netz

Die Plug-ins für Browser verschiedener LabVIEW-Versionen können gleichzeitig ausgeführt werden. VIs, die in derselben LabVIEW-Version erstellt wurden, können mit jedem Browser-Plug-in dargestellt werden. Welches Plug-in der Browser lädt, wird mit dem HTML-Tag `OBJECT/EMBED` bestimmt. Für das ActiveX-Element von LabVIEW bestimmt die `CLASSID`, die Sie im Tag `OBJECT` angeben, welches Plug-in geladen wird. Die `CLASSID` ist bei jeder Version des Plug-ins anders. Welches Plug-in für Netscape

geladen wird, bestimmt der MIME-Typ, den Sie im Tag `EMBED` angeben. Welche Syntax in einer bestimmten LabVIEW-Version zu beachten ist, wird in der jeweiligen *LabVIEW-Hilfe* beschrieben.

Neue Eigenschaften und Methoden

Im Folgenden sehen Sie eine Übersicht über alle zu LabVIEW 7.1 hinzugekommenen Eigenschaften und Methoden zum VI-Server:

- Methode “Keine Auswahl zulassen”
- Methode “Skalierte Diagramm-Grafik lesen”
- Methode “Skalierte Panel-Grafik lesen”
- Eigenschaft “Ist-Sonde”

LabVIEW Companion Products CD

Für die Entwicklung spezieller Applikationen sind verschiedene Toolkits für LabVIEW erhältlich. Weitere Hinweise dazu finden auf der LabVIEW Companion Products CD, die mit LabVIEW ausgeliefert wird bzw. auf unserer Website unter ni.com/toolkits.

Neuerungen und Änderungen an der Dokumentation

Bei bestimmten Beschreibungen von VIs und Funktionen gibt es in der *LabVIEW-Hilfe* Schaltflächen mit der Beschriftung **Beispiel öffnen** und **Nach ähnlichen Beispielen suchen**. Wenn Sie auf die Schaltfläche **Beispiel öffnen** klicken, wird zum beschriebenen Thema ein Beispiel-VI gestartet. Beim Anklicken der Schaltfläche **Nach ähnlichen Beispielen suchen** öffnet sich die Beispielsuchmaschine und es werden alle Beispiel-VIs angezeigt, die zum beschriebenen Thema passen.

Es wird empfohlen, zur Anzeige der *LabVIEW-Hilfe* einen der folgenden Browser zu verwenden:

- **(Mac OS)** Safari 1.0 oder eine aktuellere Version
- **(UNIX)** Mindestens Netscape 6.0 oder Mozilla 1.2

Unveränderte Dokumentation

Folgende Beschreibungen sind seit LabVIEW 7.0 gleich geblieben:

- *LabVIEW Development Guidelines*
- *LabVIEW Measurements Manual*
- *LabVIEW-Schnellübersicht*
- *LabVIEW-Benutzerhandbuch*
- *Using External Code in LabVIEW*

Folgende Application Notes und wissenschaftliche Ausarbeitungen sind ebenfalls LabVIEW 7.0 übernommen worden:

- *Integrating the Internet into Your Measurement System—DataSocket Technical Overview*
- *Portieren und Lokalisieren von LabVIEW-VIs*
- *Using Apple Events and the PPC Toolbox to Communicate with LabVIEW Applications on the Macintosh*
- *Using LabVIEW to Create Multithreaded VIs for Maximum Performance and Reliability*

Speichern von VIs für ältere Versionen

Zum Speichern eines in LabVIEW 7.1 erstellten VIs in LabVIEW 7.0 klicken Sie im Menü auf **Datei»Mit Optionen speichern** und aktivieren die Option **Für vorige Version speichern**. Um ein VI für LabVIEW 6.1 zu speichern, öffnen Sie es in LabVIEW 7.0 und verfahren analog.

Andere Änderungen und Neuerungen in LabVIEW 7.1

Nachfolgend alle weiteren Änderungen an LabVIEW 7.1.

Neuerungen beim Application Builder

Alle Neuerungen am Application Builder sind im *LabVIEW Benutzerhandbuch Application-Builder* beschrieben.

Änderungen an VIs und Funktionen

Hinsichtlich der vorhandenen VIs und Funktionen gibt es in LabVIEW 7.1 folgende Änderungen:

- Die Funktion “Ereignisregistrierung aufheben” befindet sich jetzt auf der **ActiveX**-Palette.
- Der Eingang **Fehlermeldung** des VIs “Fehlercluster aus Fehlercode” ist jetzt optional. An diesen Eingang können Beschreibungen für selbst erstellte Fehlercodes übergeben werden.
- Der Eingang **Vorherige ignorieren** der Funktionen “Auf Meldung warten” und “Auf Meldung von mehreren warten” arbeitet nun wieder auf dieselbe Weise wie in LabVIEW 6.1 und den Versionen davor. Wird an den Eingang **Vorherige ignorieren** für den ersten Schleifendurchlauf der Wert TRUE übergeben und für alle nachfolgenden Durchläufe der Wert FALSE, werden alle Meldungen vor der erstmaligen Ausführung der Schleife ignoriert.
- Im Formelknoten kann jetzt die `atan2`-Funktion verwendet werden.

- Es gibt zwei neue Instanzen des VIs “Digital: Vergleich”, und zwar “Signalverlauf nach Signalverlauf” und “Tabelle nach Tabelle”. Mit Hilfe des VIs lässt sich ein digitaler Signalverlauf mit einem anderen digitalen Signalverlauf bzw. einem bestimmten Wert oder ein Satz digitaler Daten mit einem anderen Satz digitaler Daten oder einem bestimmten Wert vergleichen.
- Zum Express “LabVIEW-Messdaten in Datei schreiben” ist ein neuer Eingang, **Reset**, hinzugekommen, über den der Speichervorgang der `.lvmm`-Datei neu gestartet werden kann.
- Das Express-VI “In dynamische Daten konvertieren” hat zwei neue Optionen erhalten, mit denen die **Startzeit** des Eingangssignals auf **Null** oder **Jetzt**, also die aktuelle Systemzeit gesetzt werden kann.
- Für den Eingang **VI-Referenz** des Knotens “Aufruf über Referenz” sind strikt typisierte VI-Referenzen und statische VI-Referenzen zulässig. Mit Hilfe des Knotens zum Aufruf über Referenz kann ein VI innerhalb einer statischen VI-Referenz ausgeführt werden.
- Das VI “Savitzky-Golai-Filter (Punkt für Punkt)” heißt jetzt “Savitzky-Golay-Filter (Punkt für Punkt)”.
- In LabVIEW 7.0 führte das VI “(Unvollständige) Beta-Funktion” die Beta-Funktion aus, wenn $a = 1$ war. In LabVIEW 7.1 gilt: Für alle nicht-positiven reellen Werte der Obergrenze $a \leq 1$ ist die Funktion für alle reellen nicht-negativen Werte für x und y definiert.
- Folgende VIs wurden aus der Bibliothek `vi.lib\analysis\8numeric.llb` in die Bibliothek `vi.lib\analysis\baseonly.llb`:
 - Vektor normieren
 - Matrix normieren
 - Schnellskalierung (1D)
 - Schnellskalierung (2D)
 - 1D skalieren
 - 2D skalieren
 - Einheitsvektor
- Das VI “FIR-gefensterte Koeffizienten” hat einen neuen Eingang, **Option**, mit dem bestimmt wird, ob die Werte an **FIR-gefensterte Koeffizienten** skaliert werden sollen. Die Voreinstellung lautet **nicht skaliert**.
- Das VI “MSE” arbeitet in LabVIEW 7.1 anders als bisher. Wenn die Arrays **X-Werte** und **Y-Werte** eine unterschiedliche Länge haben, berechnet das VI den Parameter **mse** auf Grundlage der Sequenz mit den wenigsten Elementen und gibt eine Warnung aus.

- Zum VI “HP34401A Initialize” ist ein neuer Parameter, **Access Mode**, hinzugekommen, in dem spezielle Zugriffsoptionen für das VI “VISA öffnen” auswählbar sind. Die Voreinstellung lautet 0. Der Eingang **Instr Descriptor** wurde in **VISA Resource** umbenannt.
- Der Parameter **Source** des VIs “HP34401A App. Example”, “HP34401A Config Trigger” und “HP34401A Getting Started” wurde in **Trigger Source** umbenannt.
- Der Parameter **Instr Descriptor** des VIs “HP34401A Find Meter” heißt nun **VISA Resource**.
- Zum VI “Exponentialanpassung” ist ein neuer Parameter, **Standardabweichung**, hinzugekommen. An diesem Eingang wird die Standardabweichung $\sigma[i]$ für die Punkte $(x[i], y[i])$ angegeben.
- Der Eingang **Name** des VIs “Semaphor auflösen” wurde entfernt.

Neue, umbenannte und verschobene VIs

Alle nachfolgend genannten Beispiel-VIs befinden sich in LabVIEW 7.1 in einem anderen Verzeichnis oder wurden umbenannt.

- Die `sndExample.llb` und die `sndExample adv.llb` im Verzeichnis `examples\sound` wurden in `sound.llb` und `sound adv.llb` umbenannt.
- Das VI “TreeView” befindet sich nun unter `examples\comm\axevent.llb`.

Im Ordner **Neue Beispiele für LabVIEW 7.1** auf der Registerkarte **Suchen** der NI-Beispielsuchmaschine finden Sie neue Beispiel-VIs und Beschreibungen dazu. Die Beispiele können direkt gestartet werden.

Verschiedenes

Nachfolgend eine lose Auflistung sonstiger Änderungen an LabVIEW 7.1:

- Die digitalen Zustände X und Z werden im digitalen Signalverlaufsgraphen in einer neuen Linienart dargestellt.
- Wenn Sie mit einer Sonde 32-Bit-Integer oder Fließkommawerte mit doppelter Genauigkeit aufnehmen, können Sie auf der Seite **Bedingung** der Sonde den Radix auswählen. Von dieser Änderung sind die bedingte DBL-Sonde, die bedingte DBL-Array-Sonde, die bedingte Signed32-Sonde und die bedingte Unsigned32-Array-Sonde betroffen.
- Damit Fieldpoint-I/O-Steuerknoten mit FieldPoint-Namen arbeiten, die weder im Measurement & Automation Explorer (MAX) noch einem anderen Konfigurationsprogramm erstellt wurden, klicken Sie das Element mit der rechten Maustaste an und wählen aus dem Kontextmenü die Option **Undefinierte Namen zulassen**.

- Der Menüpunkt **Neu und geändert in 7.0** des Dialogfelds **Optionen** wurde in **Neu und geändert in 7.x** umbenannt.
- **(Linux)** Wenn Sie die LabVIEW Runtime-Engine auf einem Zielsystem installieren, auf dem eine Anwendung mit Analyse-VIs ausgeführt werden soll, geben Sie auf die Aufforderung hin, das LabVIEW Run-Time Engine Analysis Support Package (labview71-rte-aal-7.1-1.i386.rpm) zu installieren, Ja ein.
- Wenn Sie ein Unterpanel-Bedienelement im LabVIEW Full Development System mit der rechten Maustaste anklicken, lässt sich das Element über die Menüpunkte des Kontextmenüs konfigurieren. In LabVIEW 7.0 war dazu das LabVIEW Professional Development System erforderlich.
- Operationen mit 2D-Arrays sind unter Windows und Linux nun leistungsfähiger.
- Für die Eigenschaft “Zielbetriebssystem” gibt es einen neuen möglichen Wert, nämlich “RTX”.
- Die Eigenschaft “VI-Typ” arbeitet ebenfalls mit einem neuen möglichen Wert, nämlich “SubSystem”. Der Wert wird für SubVIs des LabVIEW Simulation Module ausgegeben, die nur in einem Simulationsdiagramm enthalten sein können.
- Mit dem Web-Dokumentationswerkzeug wird nun zusätzlicher HTML-Code erzeugt, mit dem bestimmt wird, mit welcher Sprachversion der Runtime-Engine ein integriertes VI angezeigt werden soll.
- Der Mauszeiger ändert sich nicht mehr, wenn beim Herstellen einer Verbindung mit einem bestimmten Anschluss eine fehlerhafte Verbindung entstehen würde.
- Die Software-Lizenzvereinbarung von National Instruments hat sich geändert. Lesen Sie bitte die neue *Software-Lizenzvereinbarung von National Instruments* auf der Installations-CD von LabVIEW 7.1 nach, welche Lizenzbedingungen nun gültig sind.

