

Hinweise zum Upgrade von LabVIEW™

In diesem Dokument wird beschrieben, wie LabVIEW unter Windows, Mac OS und Linux auf LabVIEW 2009 aktualisiert wird, welche Probleme beim Upgrade auftreten können und um welche Funktionen die neue LabVIEW-Version erweitert wurde. Um eventuellen Kompatibilitätsproblemen vorzubeugen, sollten Sie vor dem Laden aller VIs, die mit älteren Versionen von LabVIEW erstellt wurden, diese Hinweise zum Upgrade von LabVIEW lesen. Sie sollten von alten Dateien Sicherheitskopien erstellen, bevor Sie diese in der neuen Version von LabVIEW laden.

Wenn Sie bisher mit einer LabVIEW-Version bis einschließlich 7.1 gearbeitet haben, empfehlen wir Ihnen, zusätzlich zum vorliegenden Dokument auch die folgenden Dokumente zu lesen, in denen die wichtigsten Änderungen und Neuerungen zwischen LabVIEW 7.1 und LabVIEW 2009 beschrieben sind:

- **Hinweise zum Upgrade von LabVIEW** (für Version 8.0)—Die Abschnitte *Hinweise zur Kompatibilität beim Upgrade* und *Änderungen und Neuerungen in LabVIEW 8.0* enthalten wichtige Informationen für Benutzer, die bisher mit einer älteren Version gearbeitet haben. Zum Herunterladen der *Hinweise zum Upgrade von LabVIEW* für LabVIEW 8.0 besuchen Sie die Website ni.com/info und geben Sie den Infocode `rdg8lv` ein.
- **Hinweise zum Upgrade von LabVIEW** (für Version 8.2)—Die Abschnitte *Hinweise zur Kompatibilität beim Upgrade* und *Änderungen und Neuerungen in LabVIEW 8.2* enthalten wichtige Informationen für Benutzer, die bisher mit einer älteren Version gearbeitet haben. Zum Herunterladen der *Hinweise zum Upgrade von LabVIEW* für LabVIEW 8.2 besuchen Sie die Website ni.com/info und geben Sie den Infocode `rdgup82` ein.
- **Hinweise zum Upgrade von LabVIEW** (für Version 8.5)—Die Abschnitte *Hinweise zur Kompatibilität beim Upgrade* und *Änderungen und Neuerungen in LabVIEW 8.5* enthalten wichtige Informationen für Benutzer, die bisher mit einer älteren Version gearbeitet haben. Zum Herunterladen der *Hinweise zum Upgrade von LabVIEW* für LabVIEW 8.5 besuchen Sie die Website ni.com/info und geben Sie den Infocode `rdgu85` ein.
- **Hinweise zum Upgrade von LabVIEW** (für Version 8.6)—Die Abschnitte *Hinweise zur Kompatibilität beim Upgrade* und *Änderungen und Neuerungen in LabVIEW 8.6* enthalten wichtige Informationen für Benutzer, die bisher mit einer älteren Version gearbeitet haben. Zum Herunterladen der *Hinweise zum Upgrade von LabVIEW* für LabVIEW 8.6 besuchen Sie die Website ni.com/info und geben Sie den Infocode `exdm66DE` ein.

In der *LabVIEW-Hilfe* finden Sie auch weitere Informationen zu den Funktionen von LabVIEW 2009, den Grundlagen der LabVIEW-Programmierung, schrittweise Anleitungen zur Verwendung von LabVIEW sowie Hinweise zu den VIs, Funktionen, Paletten, Menüs und Werkzeugen des Programms. In der *LabVIEW-Hilfe* wird auch auf weitere LabVIEW-Dokumentation von National Instruments verwiesen. Zum Öffnen der *LabVIEW-Hilfe* klicken Sie auf **Hilfe»LabVIEW-Hilfe durchsuchen**.

Inhalt

Upgrade auf LabVIEW 2009.....	2
Konvertieren von VIs.....	3
Upgrade von Modulen, Toolkits und Gerätetreibern.....	4
Upgrade zusätzlicher Software von National Instruments.....	5
Upgrade älterer LabVIEW-Versionen.....	6
Hinweise zur Kompatibilität beim Upgrade.....	7

Upgrade von LabVIEW 8.6.....	7
Upgrade von LabVIEW 8.5.....	11
Upgrade von LabVIEW 8.2.....	14
Upgrade von LabVIEW 8.0.....	20
Upgrade von LabVIEW 7.x.....	24
Upgrade von LabVIEW 6.x.....	39
Upgrade von LabVIEW 5.x oder älteren Versionen.....	43
Änderungen und Neuerungen in LabVIEW 2009.....	43
Installation von LabVIEW.....	43
Neue Beispiel-VIs.....	43
Verbesserungen am Blockdiagramm.....	43
Verbesserungen am Frontpanel.....	44
Optische Verbesserungen.....	45
Verbesserungen am LabVIEW Application Builder.....	50
Verbesserungen an LabVIEW-Projekten.....	51
Neue und verbesserte VIs, Funktionen und Knoten.....	51
Neue und geänderte Klassen, Eigenschaften, Methoden und Ereignisse.....	57
Verbesserungen an der objektorientierten Programmierung in LabVIEW.....	58
Verbesserungen an LabVIEW-Webdiensten (Windows, Nicht im Basispaket enthalten).....	58
SSL-Unterstützung für den LabVIEW-Webdienst.....	59
Vergleichen von VIs über die Kommandozeile oder mit Versionsverwaltungsprovidern.....	59
Fehlersuche mit dem Sondenüberwachungsfenster.....	59
Verbesserungen am Symbol-Editor.....	59
LabVIEW 2009 (64 Bit).....	60
Rekursion.....	60
Wiederverwenden von Blockdiagrammabschnitten.....	60
Verwenden des TDM-Excel-Zusatzpakets (Windows).....	61

Upgrade auf LabVIEW 2009

Im Folgenden wird beschrieben, wie das Upgrade auf die neue Version von LabVIEW durchzuführen ist und welche Dokumente gelesen werden sollten. National Instruments empfiehlt, dass Sie sich zunächst die *LabVIEW-Versionshinweise* und das vorliegende Dokument durchlesen.

- Um sicherzustellen, dass Ihnen alle Kompatibilitätsprobleme bekannt sind, sollten Sie vor der Installation der neuen LabVIEW-Version folgende Abschnitte lesen:
 - Upgrade auf LabVIEW 2009**—Dieser Abschnitt enthält Hinweise zum Upgrade von Toolkits und Modulen, zum Kopieren von Einstellungen der Programmoberfläche und von `user.lib`-Dateien aus vorhergehenden LabVIEW-Versionen sowie zum Umwandeln von VIs für LabVIEW 2009.
 - Hinweise zur Kompatibilität beim Upgrade**—Dieser Abschnitt enthält Hinweise bzgl. der Kompatibilität beim Upgrade von VIs auf die aktuelle Version von LabVIEW. Beachten Sie besonders den Abschnitt, der sich auf Ihre aktuelle LabVIEW-Version bezieht.



Hinweis Tests, mit denen sich VIs auf Kompatibilitätsprobleme prüfen lassen, können auf der Website von National Instruments unter ni.com/info nach Eingabe des Infocodes `ex36rv` heruntergeladen werden.

- Änderungen und Neuerungen in LabVIEW 2009**—Dieser Abschnitt beschreibt neue Funktionen in LabVIEW. Ausführliche Informationen zu diesen neuen Funktionen finden Sie

in der *LabVIEW-Hilfe*. Zum Öffnen der *LabVIEW-Hilfe* klicken Sie auf **Hilfe»LabVIEW-Hilfe durchsuchen**.

2. (Optional) Deinstallieren Sie ältere Versionen von LabVIEW.
3. Installieren und aktivieren Sie die aktuelle Version von LabVIEW. Alle Schritte für den Installationsvorgang von LabVIEW lesen Sie in folgenden Abschnitten der *LabVIEW-Versionshinweise*:
 - *Systemvoraussetzungen*
 - *Installation von LabVIEW 2009* und die entsprechenden Abschnitte für Ihre Plattform.
 - *Installation von LabVIEW-Zusatzpaketen*, wenn Sie LabVIEW-Toolkits oder -Module von einem anderen Speicherort als der LabVIEW-Plattform-DVD installieren möchten.
 - **(Windows)** *Aktivierung der LabVIEW-Lizenzen* und alle Unterabschnitte
 - (Optional) *Installation und Konfiguration von Geräten* und den entsprechenden Abschnitt für Ihre Plattform.
 - *Weiterführende Informationen*
4. In der *LabVIEW-Readme* finden Sie Hinweise zu Problemen, die in der neuen Version behoben wurden, sowie zu bekannten Problemen in der neuen Version und zusätzlicher Dokumentation, die nicht in der *LabVIEW-Hilfe* enthalten ist. Die Datei `readme.html` finden Sie im Verzeichnis `labview\readme`.
5. Kopieren Sie die Umgebungseinstellungen einer älteren LabVIEW-Version. Weitere Informationen zum Kopieren der Umgebungseinstellungen finden Sie unter [Kopieren der Einstellungen einer älteren LabVIEW-Version](#).
6. Kopieren Sie die `user.lib`-Dateien einer älteren LabVIEW-Version. Weitere Informationen zum Kopieren der `user.lib`-Dateien finden Sie unter [Kopieren der user.lib-Dateien einer älteren LabVIEW-Version](#).
7. Konvertieren Sie VIs für LabVIEW 2009. Informationen zum Konvertieren von VIs, die in einer älteren Version von LabVIEW gespeichert wurden, finden Sie unter *Konvertieren von VIs*.

Konvertieren von VIs

Zum Öffnen von VIs, die LabVIEW 3.x oder älteren LabVIEW-Versionen erstellt wurden, setzen Sie sich bitte mit National Instruments in Verbindung und lassen Sie sich von einem unserer Mitarbeiter die erforderlichen Änderungen am Blockdiagramm erklären. Mit LabVIEW 4.0 bis 5.x erstellte VIs müssen zunächst in LabVIEW 8.2 geöffnet und gespeichert werden, damit sie mit LabVIEW 2009 geöffnet werden können. Alle mit LabVIEW-Versionen ab einschließlich 6.0 gespeicherten VIs werden in LabVIEW 2009 automatisch umgewandelt und kompiliert. Um zu verhindern, dass das VI bei jedem Öffnen umgewandelt und neu kompiliert wird – was zusätzlichen Arbeitsspeicher kostet – sollte das VI in LabVIEW 2009 gespeichert werden. VIs mit ungespeicherten Änderungen (dazu zählt unter anderem auch eine Neukompilierung) können merklich langsamer sein als gespeicherte VIs.



Hinweis In LabVIEW 2009 gespeicherte VIs lassen sich nicht in älteren LabVIEW-Versionen öffnen. Vor dem Speichern umgewandelter VIs in LabVIEW 2009 empfiehlt es sich, eine Sicherheitskopie der VIs anzulegen, die für die Verwendung in LabVIEW 8.6, 8.5, 8.2 oder 8.0 vorgesehen sind. Zum Speichern von VIs zur Ausführung in LabVIEW 8.6, 8.5, 8.2 oder 8.0 wählen Sie im Menü **Datei»Für vorige Version speichern**.

Wenn Ihr Rechner nicht genug Arbeitsspeicher enthält, 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, die umgewandelt werden sollen. Laden und speichern Sie zunächst die SubVIs der unteren Hierarchiestufen. Fahren Sie dann in der Hierarchie aufwärts fort. Das Haupt-VI sollte zum Schluss geöffnet und konvertiert werden. Um VIs verzeichnisweise umzuwandeln, wählen Sie

Werkzeuge»Fortgeschritten»Massenkompilierung. VIs eines Verzeichnisses oder einer LLB werden in einer festgelegten Reihenfolge kompiliert. Eine Beschreibung der Reihenfolge, in der die Massenkompilierung erfolgt, finden Sie in der *LabVIEW-Hilfe* auf der Registerkarte **Inhalt** unter dem Thema **Grundlagen»Erstellen von VIs und SubVIs»Anleitung»Speichern von VIs»Massenkompilierung von VIs**. 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.

Die Größe des von LabVIEW belegten Speicherplatzes können Sie sich unter **Hilfe»Über LabVIEW** anzeigen lassen.

Upgrade von Modulen, Toolkits und Gerätetreibern

Wenn Sie ein Upgrade von einer älteren Version von LabVIEW durchführen, müssen Sie aktuelle Versionen von zuvor installierten Modulen, Toolkits oder Gerätetreibern installieren. Auf den LabVIEW Plattform-DVDs sind die meisten Module und Toolkits enthalten, die mit LabVIEW 2009 kompatibel sind. Bei Modulen und Toolkits, die nicht auf den LabVIEW-Plattform DVDs enthalten sind, geben Sie auf der Website von National Instruments ni.com/info den Infocode `compat` ein, um Informationen darüber zu erhalten, welche Module und Toolkits mit der aktuellen Version von LabVIEW kompatibel sind.

Module und Toolkits von National Instruments

In der folgenden Tabelle finden Sie Informationen zu den Installationsdatenträgern:

Betriebssystem	Datenträger	Wichtige Hinweise
Windows	DVD	Installieren Sie LabVIEW 2009 und mit dieser Version kompatible Module und Toolkits mithilfe der LabVIEW-Plattform-DVDs. Des Weiteren können Sie Module und Toolkits auswählen, die im Evaluierungsmodus ausgeführt werden. Mit den LabVIEW-Plattform-DVDs werden neue Versionen eines Toolkits mit LabVIEW 2009 installiert, ohne dass Sie ältere Versionen deinstallieren oder modifizieren müssen. Informationen zur Installation von LabVIEW, Modulen und Toolkits finden Sie in den <i>LabVIEW-Versionshinweisen</i> .
Mac/Linux; Windows, wenn die LabVIEW-Plattform-DVDs das Modul oder Toolkit nicht enthalten	CD	Verwenden Sie die Installations-CD, die mit dem Modul oder Toolkit geliefert wurde. Stellen Sie vor Beginn der Installation sicher, dass es sich um eine kompatible Version des Moduls oder Toolkits handelt. Weitere Informationen darüber, welche LabVIEW-Module und -Toolkits mit der aktuellen Version von LabVIEW kompatibel sind, erhalten Sie auf der Website von National Instruments ni.com/info unter Eingabe des Infocodes <code>compat</code> . Installieren Sie dann Module und Toolkits im Verzeichnis von LabVIEW 2009. Massenkompilieren Sie alle VIs, die in Vorgängerversionen von LabVIEW gespeichert wurden. Weitere Informationen dazu finden Sie im Abschnitt <i>Massenkompilierung in LabVIEW</i> .



Hinweis Einige Versionen von Toolkits können nicht mit LabVIEW 2009 verwendet werden. Die Installation eines inkompatiblen Toolkits kann zu Fehlern bei einigen Funktionen des Toolkits oder von LabVIEW führen. National Instruments empfiehlt daher, die Kompatibilität von Toolkits vor deren Installation sicherzustellen. Weitere Informationen darüber, welche LabVIEW-Module und -Toolkits mit der aktuellen Version von LabVIEW kompatibel sind, erhalten Sie auf der Website von National Instruments ni.com/info unter Eingabe des Infocodes `compat`. Wenn die installierte LabVIEW-Version nach der Installation eines inkompatiblen Toolkits nicht mehr ordnungsgemäß funktioniert, deinstallieren Sie zunächst das Toolkit über die Option "Programme ändern oder entfernen" in der Systemsteuerung und reparieren Sie dann die installierte LabVIEW-Version.

Instrumententreiber

Für die Steuerung von und Kommunikation mit jeglicher Hardware sind aktuelle Gerätetreiber erforderlich. Wenn ein Instrumententreiber mit einer älteren Version von LabVIEW installiert wurde, muss dieser mit LabVIEW 2009 erneut installiert werden. Folgende Methoden stehen dafür zur Auswahl:

- **Treiber für modulare Instrumente von National Instruments**—Die Treiber für modulare Instrumente von National Instruments werden mit Hilfe der CD oder DVD für NI-Gerätetreiber installiert.
- **Treiber für Plug-and-Play-Geräte**—LabVIEW-Treibern für Plug-and-Play-Geräte können mit Hilfe der Suchmaschine für NI-Gerätetreiber gesucht werden. Zur Installation dieser Treiber müssen Sie LabVIEW nicht verlassen.
- **IVI-Treiber oder nicht geprüfte Gerätetreiber**—IVI-Treiber und nicht von National Instruments geprüfte Treiber können über das [Netzwerk für Gerätetreiber](#) auf der Website von National Instruments gesucht werden.



Hinweis Wenn Sie Treiber mithilfe der Suchmaschine für NI-Gerätetreiber neu installieren, wird eine Massenkompilierung des Verzeichnisses `labview\instr.lib` empfohlen.

Zusatzpakete von Drittanbietern

Um festzustellen, ob ein LabVIEW-Zusatzpaket mit LabVIEW 2009 und Ihrem Betriebssystem kompatibel ist, wenden Sie sich an den Hersteller des Produkts. Stellen Sie sicher, dass alle mit dem Zusatzpaket arbeitenden VIs massenkompiliert werden.

Weitere Informationen dazu finden Sie im Abschnitt *Massenkompilierung in LabVIEW*.

Massenkompilierung in LabVIEW

VIs, die zuletzt in einer älteren LabVIEW-Version gespeichert wurden, werden beim Öffnen 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 der aktuellen LabVIEW-Version gespeichert werden. Für alle VIs, die mit Hilfe von Zusatzpaketen von Drittanbietern oder LabVIEW-Modulen und -Toolkits installiert werden, die nicht auf den LabVIEW-Plattform-DVDs enthalten sind, wird eine Massenkompilierung empfohlen.

Weitere Informationen dazu finden Sie unter dem Thema **Grundlagen»Erstellen von VIs und SubVIs»Anleitung»Speichern von VIs»Massenkompilierung von VIs**.

Upgrade zusätzlicher Software von National Instruments

Für LabVIEW 2009 wird eine NI-Teststand-Version ab einschließlich 3.5 benötigt. Hinweise zum Upgrade und zum Bestellen dieses Programms finden Sie auf der Website ni.com/info nach Eingabe des Infocodes `exd8yy`.

NI TestStand 3.5 und neuere Versionen geben bei der Konfiguration folgender Express-VIs von LabVIEW 2009 einen Fehler aus:

- "Digitalsignale gruppieren"
- "Attribute für dynamische Daten ermitteln"
- "Attribute für dynamische Daten festlegen"

Weitere Informationen zu dem Fehler finden Sie auf der Website von National Instruments ni.com/info nach Eingabe des Infocodes `rdtf10`.



Hinweis In NI TestStand ab Version 4.1 ist dieses Problem behoben.

Weitere Informationen zu Problemen mit LabVIEW und NI TestStand finden Sie in der Datei `Readme.html` auf der Installations-CD von NI TestStand oder im Verzeichnis `<TestStand>\Doc`.

Für LabVIEW 2009 wird NI Spy 2.3 oder eine neuere Version empfohlen. NI Spy 2.5 ist auf der Gerätetreiber-CD von National Instruments enthalten.

LabVIEW 2009 arbeitet mit Measurement Studio ab Version 8.0. Zum Bestellen der benötigten Version von Measurement Studio besuchen Sie die Website ni.com/info und geben Sie den Infocode `exd8yy` ein.

Upgrade älterer LabVIEW-Versionen

LabVIEW 2009 kann ohne vorherige Deinstallation älterer Versionen installiert werden. Versionen von LabVIEW haben möglicherweise gemeinsame Komponenten. Ein Upgrade von LabVIEW hat jedoch keine Auswirkungen auf die 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 werden im Verzeichnis `labview x` gespeichert, wobei `x` für die Version steht.

Ersetzen einer bestehenden LabVIEW-Version

Deinstallieren Sie die bestehende Version von LabVIEW, starten Sie das Installationsprogramm von LabVIEW 2009 und wählen Sie das `labview`-Verzeichnis der letzten Version als Installationsverzeichnis.

(Windows) Die bisherige LabVIEW-Version kann auch in der Systemsteuerung unter "Software" deinstalliert und durch LabVIEW 2009 ersetzt werden. Dateien, die im `labview`-Verzeichnis erstellt wurden, bleiben vom Deinstallationsprogramm unberührt.



Hinweis Bei der Deinstallation oder Neuinstallation von LabVIEW werden alle LLB-Dateien im Verzeichnis `vi.lib` einschließlich der darin befindlichen VIs und Elemente entfernt. Speichern Sie Ihre VIs und Elemente daher im Verzeichnis `user.lib`. So werden die VIs auch den Paletten **Funktionen** und **Elemente** hinzugefügt.

Kopieren der Einstellungen einer älteren LabVIEW-Version

Um die Einstellungen zur LabVIEW-Oberfläche von der letzten Version zu übernehmen, kopieren Sie die Datei mit den LabVIEW-Voreinstellungen (`*.ini`) aus dem `labview`-Verzeichnis.



Vorsicht Alle bis dato vorgenommenen Änderungen an den Grundeinstellungen werden damit in der neuen Version überschrieben.

Kopieren Sie die Datei mit den Einstellungen nach der Installation in das Programmverzeichnis von LabVIEW 2009.

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

(Mac OS) Die Einstellungen werden in einer Textdatei unter `~/Library/Preferences/LabVIEW 9.0 Preferences` gespeichert.

(Linux) Die Einstellungen werden unter

`/home/<Benutzername>/natinst/.config/LabVIEW-2009/labview.conf` gespeichert, wobei `<Benutzername>` der Name des Benutzers ist, der die aktuelle LabVIEW-Instanz ausführt.



Hinweis (Linux) Das Format der Einstellungen hat sich in LabVIEW 2009 von `myapp.preferences_name: value in preference_name = value` geändert. Wenn Sie die Datei mit den LabVIEW-Einstellungen in das Verzeichnis von LabVIEW 2009 kopieren, müssen Sie die Einstellungen manuell an das neue Format anpassen.

Kopieren der user.lib-Dateien einer älteren LabVIEW-Version

Um Dateien aus dem Ordner `user.lib` von der bisherigen Version zu übernehmen, kopieren Sie die Dateien aus dem `labview`-Verzeichnis der vorigen LabVIEW-Version. Kopieren Sie die Dateien nach der Installation von LabVIEW 2009 in das Verzeichnis `user.lib` des neuen LabVIEW-Verzeichnisses.

Hinweise zur Kompatibilität beim Upgrade

In den nachfolgenden Abschnitten werden Probleme beim Upgrade bezüglich der Kompatibilität verschiedener LabVIEW-Versionen besprochen. Allgemeine Informationen zum Upgrade auf die neueste LabVIEW-Version finden Sie auf der Website ni.com/info nach Eingabe des Infocodes `lvupgradeDE`.

Bekannte Funktionsprobleme, Kompatibilitätsprobleme sowie kurz vor Veröffentlichung hinzugefügte Funktionen von LabVIEW 2009, die nicht anderweitig dokumentiert sind, werden in der Datei `readme.html` im `labview`-Verzeichnis beschrieben. Allgemeine Informationen zur neuesten LabVIEW-Version finden Sie auch in der [Developer Zone](#) auf ni.com.

Upgrade von LabVIEW 8.6

Beim Upgrade von LabVIEW 8.6 auf LabVIEW 2009 können die nachfolgend aufgeführten Kompatibilitätsprobleme auftreten.



Hinweis Zusätzliche Informationen zu weiteren Problemen, die beim Upgrade von LabVIEW 8.6.x auftreten können, erhalten Sie auf der Website von National Instruments ni.com/info nach Eingabe des Infocodes `exdm66DE`.

Systemvoraussetzungen

(Windows) Für LabVIEW 2009 ist ein Speicherplatz von mindestens 1,6 GB erforderlich.

(Mac OS) Für die Minimalversion von LabVIEW 2009 sind mindestens 563 MB Festplattenspeicher erforderlich. Für die vollständige Installation benötigt die Version 1,2 GB.

(Linux) Für die Minimalversion von LabVIEW 2009 sind mindestens 630 MB Festplattenspeicher erforderlich und für die vollständige Installation 835 MB.

Änderungen an der Funktionsweise von VIs und Funktionen

Die Funktionsweise der nachfolgenden VIs und Funktionen hat sich in LabVIEW 2009 geändert.

VIs und Funktionen für Bluetooth

Zur Nutzung der **Bluetooth**-VIs und -Funktionen muss mindestens Windows XP mit Service Pack 2 installiert sein.

Signalerzeugungs-VIs

Die nachfolgenden VIs der Palette **Signalerzeugung** wurden für LabVIEW 2009 umgeschrieben. Um die neuen Funktionen verwenden zu können, ersetzen Sie die folgenden VIs durch die neuen VIs auf der **Funktionen**-Palette:

- "Bernoulli-Rauschen"
- "Binäre MLS"
- "Binomisches Rauschen"
- "Gamma-Rauschen"
- "Gaußsches weißes Rauschen"
- "Poisson-Rauschen"

- "Gleichverteiltes weißes Rauschen"

Allgemeine Änderungen an der Funktionsweise von VIs und Funktionen

In LabVIEW 2009 gibt es folgende allgemeine Änderungen an der Funktionsweise von VIs und Funktionen:

- Wenn Sie einen Wert mit einer Einheit mit ungeradem Exponenten mit der Quadratwurzelfunktion verbinden, wird die Verbindung ungültig, da LabVIEW keine gebrochenen Exponenten unterstützt.
- Das VI "Bessel-Koeffizienten" wurde zur genaueren Implementierung von Grenzfrequenzen geändert. Das VI wird daher möglicherweise langsamer als in älteren LabVIEW-Versionen ausgeführt. Das gilt auch für VIs, die dieses VI aufrufen.
- LabVIEW verteilt Webdienste in versionsspezifische Verzeichnisse. Ein typisches Stammverzeichnis für in LabVIEW 2009 verteilte Webdienste lautet zum Beispiel `C:\Dokumente und Einstellungen\All Users\Anwendungsdaten\National Instruments\Web Services 2009 32-bit`. Alle mit vorherigen LabVIEW-Versionen verteilten Webdienste müssen erneut verteilt werden. Zum Löschen eines Webdienstes, der mit einer vorherigen LabVIEW-Version verteilt wurde, muss dieser manuell aus seinem Speicherort gelöscht werden.
- Das VI "Integral x(t)" wurde für LabVIEW 2009 umgeschrieben. Um die neuen Funktionen verwenden zu können, ersetzen Sie dieses VI durch das neue VI "Integral x(t)" der **Funktionen-Palette**.

Veraltete VIs und Funktionen

Folgende VIs und Funktionen werden von LabVIEW 2009 nicht unterstützt:

- **LToCStr**—Verwenden Sie stattdessen die Funktion "LToCStrN". Die Funktion "LToCStrN" unterscheidet sich von der Funktion "LToCStr", da ein Parameter zum Festlegen der Größe des C-String-Puffers benötigt wird, in den der String kopiert wird. Bei den genannten Funktionen handelt es sich um Funktionen des Code-Interface-Knotens (CIN).
- **Sound VIs (Mac OS)**—Verwenden Sie stattdessen die **Sound**-VIs. Die API von LabVIEW 2009 ist unter Windows, Mac OS und Linux identisch.

Veraltete Eigenschaften, Methoden und Ereignisse

Die folgenden Eigenschaften, Methoden und Ereignisse werden von LabVIEW 2009 nicht mehr unterstützt:

- Die Eigenschaft "Busname" der Klasse "Digitalgraph". Verwenden Sie stattdessen die Eigenschaft "Plotname".
- Die Eigenschaft "Namen aufgerufener Objekte" der Klasse "VI". Verwenden Sie stattdessen die Methode "VI-Abhängigkeiten ermitteln (Namen und Pfade)". Die Methode "VI-Abhängigkeiten ermitteln (Namen und Pfade)" hat den gleichen Funktionsumfang wie die Eigenschaft "Namen aufgerufener Objekte" bei Verwendung der Standardwerte für alle Eingangsparameter.
- Die Eigenschaft "Callees" der Klasse "VI Properties" (ActiveX).

Umbenannte Eigenschaften, Methoden und Ereignisse

- In LabVIEW 2009 sind die XML-Parser-Klassen nicht mehr durch "XML" gekennzeichnet. So wurde z. B. "XML_Attribute" in "Attribute" geändert.
- Folgende Eigenschaften, Methoden und Ereignisse wurden in LabVIEW 2009 umbenannt:

Klasse	Name in LabVIEW 8.6	Name in LabVIEW 2009	Typ
Dokument	Namensraum?	"Namensräume verarbeiten"	Eigenschaft
Dokument	Schema?	"Schema verarbeiten"	Eigenschaft

Klasse	Name in LabVIEW 8.6	Name in LabVIEW 2009	Typ
Variable	Alarm:UngültigerStatus:ProtokollMöglich	"Alarm:UngültigerStatus:Protokoll möglich"	Eigenschaft
Variable	Alarm:Boolesch:AlarmEin	"Alarm:Boolesch:Alarm ein"	Eigenschaft
Variable	Alarm:Boolesch:ProtokollMöglich	"Alarm:Boolesch:Protokoll möglich"	Eigenschaft
Variable	Alarm:High:ProtokollMöglich	"Alarm:High:Protokoll möglich"	Eigenschaft
Variable	Alarm:HighHigh:ProtokollMöglich	"Alarm:HighHigh:Protokoll möglich"	Eigenschaft
Variable	Alarm:Low:ProtokollMöglich	"Alarm:Low:Protokoll möglich"	Eigenschaft
Variable	Alarm:LowLow:ProtokollMöglich	"Alarm:LowLow:Protokoll möglich"	Eigenschaft
Variable	Alarm:Änderungsrate:ProtokollMöglich	"Alarm:Änderungsrate:Protokoll möglich"	Eigenschaft
Variable	Alarm:U32Bitfeld:AlarmEin	"Alarm:U32Bitfeld:Alarm ein"	Eigenschaft
Variable	Alarm:U32Bitfeld:ProtokollMöglich	"Alarm:U32Bitfeld:Protokoll möglich"	Eigenschaft
Variable	Alarm:U32Bitfeld:MaskeAuswählen	"Alarm:U32Bitfeld:Maske auswählen"	Eigenschaft
Variable	Protokollierung:Datenprotokoll	"Protokollierung:Daten protokollieren"	Eigenschaft
Variable	Protokollierung:Ereignisprotokoll	"Protokollierung:Ereignisse protokollieren"	Eigenschaft
Variable	Netzwerk:ElemGröße	"Netzwerk:Elementgröße"	Eigenschaft
Variable	Netzwerk:PunkteProSignalverlauf	"Netzwerk:Werte pro Signalverlauf"	Eigenschaft
Variable	Netzwerk:BindungVerwenden	"Netzwerk:Bindung verwenden"	Eigenschaft
Variable	Netzwerk:PufferVerwenden	"Netzwerk:Puffern"	Eigenschaft
Variable	Echtzeit:DatenpunkteInSignalverlauf	"Echtzeit:Werte im Signalverlauf"	Eigenschaft
Variable	Echtzeit:PufferVerwenden	"Echtzeit:Puffern"	Eigenschaft
Variable	Skalierung:MaxSkaliert	"Skalierung:Skaliertes Max"	Eigenschaft
Variable	Skalierung:MinSkaliert	"Skalierung:Skaliertes Min"	Eigenschaft
Variable	Skalierung:MaskeInvertieren	"Skalierung:Maskeninvertierung"	Eigenschaft
Variable	Skalierung:MaxGemessen	"Skalierung:Unskaliertes Max"	Eigenschaft
Variable	Skalierung:MinGemessen	"Skalierung:Unskaliertes Min"	Eigenschaft
Variable	Skalierung:MaskeAuswählen	"Skalierung:Maske auswählen"	Eigenschaft

Änderungen am Application Builder

In LabVIEW 8.6 wurden VIs und Bibliotheken in Form einer einfachen Liste zusammen mit der Applikation gespeichert und alle VIs mit identischen Namen in unterschiedlichen Ordnern gespeichert. In LabVIEW 2009 speichert der Application Builder alle Quelldateien in der Applikation, wobei die

Dateien in einer ähnlichen Verzeichnisstruktur wie die Originaldateien angeordnet sind. Auf diese Weise wird die aufgestellte Dateihierarchie in der Applikation bewahrt.

Beim dynamischen Aufrufen von VIs sollten Sie relative Pfade verwenden, weil nur so das ordnungsgemäße Laden der VIs während der Ausführung gewährleistet werden kann.

Änderungen an den Ausgangstunneln von Case-Strukturen

In LabVIEW 2009 wird der Datentyp eines Ausgangstunnels einer Case-Struktur durch Verwendung eines Datentyps festgelegt, den alle Cases in der Struktur verarbeiten können (einschließlich der Cases, die nie ausgeführt werden). Stellen Sie sich z. B. eine Case-Struktur mit den Cases TRUE und FALSE vor. Im TRUE-Case ist der Datentyp U8 mit einem Ausgangstunnel verbunden. Im FALSE-Case ist der Datentyp U32 mit dem Ausgangstunnel verbunden. Wenn Sie in LabVIEW 8.6.x eine Konstante zur Auswahl des TRUE-Cases verbinden, ist der Datentyp am Ausgangstunnel U8, da die Konstante die Ausführung des FALSE-Cases verhindert. Wenn Sie in LabVIEW 2009 eine Konstante zur Auswahl des TRUE-Cases verbinden, lautet der Datentyp des Ausgangstunnels dagegen U32.

Aufgrund dieser Änderung sind VIs, die mit LabVIEW 8.6.x erstellt wurden, möglicherweise in LabVIEW 2009 funktionsuntüchtig, wenn der Ausgangsdantentyp eine Festkommazahl oder ein Array fester Größe ist.

Änderungen an den VIs für benutzerdefinierte Symbol-Editoren

In bisherigen Versionen von LabVIEW wurde beim Aufrufen eines VIs, das als benutzerdefinierter Symbol-Editor dient, automatisch das Frontpanel des VIs geöffnet. In LabVIEW 2009 müssen Sie ein als benutzerdefinierter Symbol-Editor genutztes VI so konfigurieren, dass es beim Aufruf sein Frontpanel öffnet. Wenn Sie mit einfachen VIs arbeiten, deren Frontpanel-Elemente vor dem Öffnen nicht neu angeordnet werden müssen, können Sie die Eigenschaft "Ausführung:Frontpanel bei Aufruf anzeigen" verwenden. Bei komplizierteren VIs, deren Frontpanel-Inhalt vor dem Öffnen neu angeordnet werden muss, sollten Sie mit der Methode "Frontpanel:Öffnen" arbeiten.

Änderungen an benutzerdefinierten Sonden (Linux)

Benutzerdefinierte Sonden, die in LabVIEW-Versionen bis einschließlich 8.6 gespeichert wurden, können nicht in LabVIEW 2009 geöffnet werden. Sie müssen die Sonden manuell aus dem Verzeichnis `LabVIEW Data` der Vorgängerversion in das gleichnamige Verzeichnis von LabVIEW 2009 kopieren. Das Verzeichnis `LabVIEW Data` von LabVIEW 2009 befindet sich unter `/home/<username>/LabVIEW Data`.

Änderungen in LabVIEW MathScript

LabVIEW MathScript ist nicht mehr im Full oder Professional Development System enthalten. Beginnend mit LabVIEW 2009 wird LabVIEW MathScript als LabVIEW MathScript RT Module installiert. VIs mit MathScript-Knoten, die in älteren LabVIEW-Versionen erstellt wurden, können daher erst nach der Installation und Aktivierung des MathScript RT Modules oder nach dem Entfernen der MathScript-Knoten gestartet werden. Wenn Sie das MathScript RT Module bereits erworben haben, können Sie zur Aktivierung des Produkts in LabVIEW **Hilfe»LabVIEW-Komponenten aktivieren** auswählen.

Änderungen an .NET

Zum Erstellen von und Kommunizieren mit .NET-Objekten wird mindestens .NET Framework 2.0 benötigt.

Upgrade von LabVIEW 8.5

Beim Upgrade von LabVIEW 8.5 auf LabVIEW 2009 können folgende Kompatibilitätsprobleme auftreten. Sonstige potentielle Probleme beim Upgrade von LabVIEW sind im Abschnitt [Upgrade von LabVIEW 8.6](#) beschrieben.



Hinweis Zusätzliche Informationen zu weiteren Problemen, die beim Upgrade von LabVIEW 8.5.x auftreten können, erhalten Sie auf der Website von National Instruments ni.com/info nach Eingabe des Infocodes `rdgu85`.

Unterstützte Plattformen

Die LabVIEW-Versionen ab einschließlich 8.6 unterstützen keine Macintosh-Computer mit PowerPC-Prozessoren.

Systemvoraussetzungen

(Windows) Zur Installation von LabVIEW 8.6 oder LabVIEW 2009 wird ein Festplattenspeicher von mindestens 1,6 GB benötigt.

(Mac OS) Für LabVIEW 8.6 wird ein Festplattenspeicher von mindestens 262 MB benötigt. Für die Minimalversion von LabVIEW 2009 sind mindestens 563 MB Festplattenspeicher erforderlich. Für die vollständige Installation benötigt die Version 1,2 GB.

(Linux) Für die Minimalversion von LabVIEW 8.6 sind mindestens 365 MB Festplattenspeicher und für die vollständige Installation 651 MB erforderlich. Für die Minimalversion von LabVIEW 2009 sind mindestens 630 MB Festplattenspeicher erforderlich und für die vollständige Installation 835 MB.

Änderungen an der Funktionsweise von VIs und Funktionen

Die Funktionsweise der nachfolgenden VIs und Funktionen hat sich in LabVIEW-Versionen ab 8.6 geändert.

Protokoll-VIs

Die **Protokoll-VIs** wurden mit Hilfe der LabVIEW-Klassen neu geschrieben. Die Anschlüsse **Protokoll (Eingang)** und **Protokoll (Ausgang)** wurden vom Datentyp "Referenz" in den Datentyp "LabVIEW-Klasse" geändert. Wenn Sie Konstanten sowie Bedien- und Anzeigeelemente nicht durch Rechtsklick auf "Typ-Def. (Referenz)" erstellt haben, können die VIs möglicherweise nicht ordnungsgemäß ausgeführt werden, da LabVIEW diese Objekte nicht aktualisieren kann. Darüber hinaus funktionieren alle "Knoten zum Aufruf über Referenz", die den vorherigen Datentyp (Referenz) der Parameter **Protokoll (Eingang)** und **Protokoll (Ausgang)** aufrufen, nicht wie erwartet.

Wenn Sie mit Hilfe der **Protokoll-VIs** ein HTML-Protokoll erstellen, das auf einem Zielsystem ausgeführt werden soll, muss das Zielsystem bei der Protokollerstellung über eine Referenz angegeben werden. Wenn Sie das HTML-Protokoll auf einem Host-Rechner erstellen und dann ohne Referenz auf ein Zielsystem weitergeben, sind die VIs nicht ausführbar.

Der Eingang **Ausrichtung** des VIs "Protokollausrichtung festlegen" wurde von einem vorzeichenlosen Word-Integer (U16) in einen Long-Integer (I32) geändert.

Der Standardwert für den Eingang **Konfigurationsdaten der Express-VIs hinzufügen** des VIs "Liste der SubVIs an Protokoll anfügen" wurde von TRUE in FALSE geändert.

Externer Code (DLLs und CINs)

Die Speichermanager-Funktionen enthalten nur eine Speicherzone, DS (data space, Datenraum). Bei der Arbeit mit C und C++ ersetzen CINs und DLLs, die den LabVIEW-Arbeitsspeicher verwalten, alle Referenzen auf Speicherfunktionen der Applikationszone (AZ) durch die entsprechende DS-Funktion.

Allgemeine Änderungen an der Funktionsweise von VIs und Funktionen

In LabVIEW-Versionen ab 8.6 gibt es folgende allgemeine Änderungen an der Funktionsweise von VIs und Funktionen:

- Das VI "STFT-Spektrogramm" wurde umgeschrieben und hat in LabVIEW-Versionen ab 8.6 zwei neue Eingänge. Ersetzen Sie dieses VI aus älteren LabVIEW-Versionen durch das VI "STFT-Spektrogramm" auf der Palette **Funktionen**.
- Viele der Mathematik- und Signalverarbeitungs-VIs wurden von nicht ablaufinvariant in ablaufinvariant geändert. Aufgrund dieser Änderungen sollten Sie nicht zu viele dieser VIs von einem ablaufinvarianten VI aus aufrufen, das auf Kopien zwischen Instanzen austauschen eingestellt ist. Weitere Informationen dazu, welche VIs nicht von einem VI aus aufgerufen werden können, das auf "Kopien zwischen Instanzen austauschen" eingestellt ist, finden Sie auf der Website von National Instruments ni.com/info nach Eingabe des Infocodes `exrehi`.
- In LabVIEW-Versionen ab 8.6 werden Einzelprozess-Umgebungsvariablen automatisch auf systemvariabel eingestellt. Einzelprozess-Variablen können nicht als "absolut" konfiguriert werden.
- Wenn Sie einen leeren Pfad mit dem Eingang **Pfad (Eingang)** des "Knotens zum Aufruf externer Bibliotheken" verbinden, gibt LabVIEW keinen Fehler mehr aus.
- Der Ausgang **Ausgabeelement** des VIs "Protokolltyp ermitteln" wurde in **Protokolltyp** geändert. Blockdiagrammabschnitte, die Sie aus älteren LabVIEW-Versionen übernehmen und in LabVIEW-Versionen ab 8.6 einfügen, werden automatisch umbenannt und verbunden. Das VI kann jedoch nicht ausgeführt werden, wenn Sie das VI "Protokolltyp ermitteln" mit Hilfe der Funktion "Aufruf über Referenz" aufrufen.
- Der Eingang **Protokolltyp** des VIs "Neues Protokoll" ist erforderlich. Mit diesem Eingang müssen Daten verbunden werden. Sie können das VI mit der rechten Maustaste anklicken und eine Konstante oder ein Bedienelement mit dem Eingang **Protokolltyp** verbinden.

Veraltete VIs und Funktionen

Folgende VIs und Funktionen werden von LabVIEW ab Version 8.6 nicht mehr unterstützt:

- **Einzelne Lösung für nichtlineares System**—Verwenden Sie stattdessen das VI "**nD Einzelne Lösung für nichtlineares System**". Das VI "nD Einzelne Lösung für nichtlineares System" ist ablaufinvariant.
- **Nichtlineares System lösen**—Verwenden Sie stattdessen das VI "**nD Nichtlineares System lösen**". Das VI "nD Nichtlineares System lösen" ist ablaufinvariant.
- **Semaphor erzeugen**—Verwenden Sie stattdessen das VI "**Semaphor-Referenz anfordern**". Das VI "Semaphor-Referenz anfordern" unterscheidet sich vom VI "Semaphor erzeugen", da bei mehrfacher Verwendung des VIs "Semaphor erzeugen" zum Erzeugen von mehreren Semaphoren mit dem gleichen Namen mehrere Kopien einer Referenz auf dieses Semaphor erstellt werden. Wenn das VI "Semaphor-Referenz anfordern" für mehrere Referenzen auf dasselbe benannte Semaphor genutzt wird, ist jede Referenz anders. Da vorhandene VIs zur Verwendung des VIs "Semaphor-Referenz anfordern" nicht automatisch umgewandelt werden, müssen alle VIs manuell aktualisiert werden, die in einer älteren Version von LabVIEW gespeichert wurden.
- **Semaphor auflösen**—Verwenden Sie stattdessen das VI "**Semaphor-Referenz freigeben**". Das VI "Semaphor-Referenz freigeben" unterscheidet sich vom VI "Semaphor auflösen", da bei Verwendung des VIs "Semaphor auflösen" zum Auflösen eines Semaphors auch alle anderen Kopien der Referenz auf dieses Semaphor aufgelöst werden. Wenn Sie mit Hilfe des VIs "Semaphor-Referenz freigeben" eine Referenz auf ein Semaphor freigeben, bleiben andere Referenzen auf dieses Semaphor weiterhin gültig. Das Semaphor wird nur aufgelöst, wenn keine weiteren Referenzen darauf bestehen. Da vorhandene VIs zur Verwendung des VIs

"Semaphor-Referenz freigeben" nicht automatisch umgewandelt werden, müssen VIs, die in einer älteren Version von LabVIEW gespeichert wurden, manuell aktualisiert werden.



Hinweis Um unerwartete Ergebnisse zu vermeiden, leiten Sie keine Referenzen, die mit den VIs "Semaphor erzeugen" oder "Semaphor auflösen" verwendet werden, an die VIs "Semaphor-Referenz anfordern" oder "Semaphor-Referenz freigeben" und umgekehrt.

- **Xmath-Skriptknoten**—Verwenden Sie stattdessen den MathScript-Knoten. Da sich die MathScript-Syntax von der Xmath-Syntax unterscheidet, müssen Sie möglicherweise vorhandene Skripte bearbeiten, damit diese im MathScript-Knoten funktionieren.

Änderungen der Funktionsweise von Eigenschaften, Methoden und Ereignissen

Die Funktionsweise folgender Eigenschaften, Methoden und Ereignisse hat sich in LabVIEW-Versionen ab 8.6 geändert:

- Die Eigenschaft "Kamera-Controller:Typ" der Klasse "Szenengraph-Anzeige" enthält den Wert **Orientiert**.
- Die Eigenschaft "Kamera-Controller:Typ" der Klasse "Szenenfenster" enthält den Wert **Orientiert**.
- Mit Hilfe der Eigenschaft "Skalierung:Aktiviert" der Klasse "Variable" können Sie Skalierungen nur für Netzwerk-Umgebungsvariablen, I/O-Variablen oder I/O-Aliase verwenden.

Veraltete Eigenschaften, Methoden und Ereignisse

Die folgenden Eigenschaften, Methoden und Ereignisse werden von LabVIEW-Versionen ab 8.5 nicht mehr unterstützt:

- Die Methode "Elementwert:Festlegen [Serialisiert]" der Klasse "VI". Verwenden Sie stattdessen die Methode "Elementwert:Festlegen".
- Die Methode "Elementwert:Alle lesen [Serialisiert]" der Klasse "VI". Verwenden Sie stattdessen die Methode "Elementwert:Alle ermitteln".
- Die Methode "Elementwert:Lesen [Serialisiert]" der Klasse "VI". Verwenden Sie stattdessen die Methode "Elementwert:Ermitteln".
- Die Eigenschaft "VIModificationBitSet" der Klasse "VI Properties" (ActiveX). Verwenden Sie stattdessen die Eigenschaft "VIModificationBitSet2".
- Die Eigenschaft "Modifikationen:VI-Modifikationen" der Klasse "VI". Verwenden Sie stattdessen die neue Eigenschaft "Modifikationen:VI-Modifikationen". In LabVIEW 8.5 und Vorgängerversionen gibt die Eigenschaft "Modifikationen:VI-Modifikationen" einen 32-Bit-Wert aus. In LabVIEW-Versionen ab 8.6 gibt die neue Eigenschaft "Modifikationen:VI-Modifikationen" einen 64-Bit-Wert aus.

Umbenannte Eigenschaften, Methoden und Ereignisse

Folgende Eigenschaften, Methoden und Ereignisse wurden in LabVIEW-Versionen ab 8.6 umbenannt:

Klasse	Name in LabVIEW 8.5	Name in LabVIEW 8.6	Typ
GObject	Maße:Höhe	"Maße:Bereichshöhe"	Eigenschaft
GObject	Maße:Breite	"Maße:Bereichsbreite"	Eigenschaft
Projektobjekt	Verknüpfung mit Festplatte aufheben	"Autofüllfunktion anhalten"	Methode
Baumstruktur	Symbol für Erweitern/Reduzieren:Auf höchster Ebene anzeigen	"Symbol für Erweitern/Reduzieren:Symbol auf höchster Ebene anzeigen"	Eigenschaft
VI	Elementwert:Festlegen [Variant]	"Elementwert:Festlegen"	Methode
VI	Elementwert:Abfragen [Variant]	"Elementwert:Ermitteln"	Methode

Klasse	Name in LabVIEW 8.5	Name in LabVIEW 8.6	Typ
VI	Elementwert:Alle ermitteln [Variant]	"Elementwert:Alle ermitteln"	Methode

Änderungen an Umgebungsvariablen

Wenn ein VI, das eine Umgebungsvariable enthält, ausgeführt wird oder zur Ausführung reserviert wurde, können folgende Eigenschaften der Umgebungsvariablen nicht bearbeitet werden:

- Alle Eigenschaften auf der Seite **Variable** des Dialogfelds **Eigenschaften für Umgebungsvariable**.
- Die Eigenschaften des Typs **Puffern** auf der Seite **Netzwerk** des Dialogfelds **Eigenschaften für Umgebungsvariable**.
- **(RT Module)** Alle Eigenschaften auf der Seite **Echtzeit-FIFO** des Dialogfelds **Eigenschaften für Umgebungsvariable**.

Das Gleiche gilt für das Entfernen oder Umbenennen von Umgebungsvariablen oder den Objekten, die im **Projekt-Explorer** mit der Variable in Zusammenhang stehen. Beides ist erst möglich, wenn das VI nicht mehr zur Ausführung reserviert ist.

Erweitern der Lizenzen für Netzwerk-Frontpanel (Windows)

Applikationen oder Frontpanels können über ein Netzwerk auf einem anderen Computer angezeigt werden. LabVIEW unterstützt Lizenzen für 5, 20, 50 oder unbegrenzt viele Clients, die gleichzeitig mit einem Netzwerk-Frontpanel verbunden sind. Auf dem Serverrechner kann nur eine Lizenz vorhanden sein. Sie sollten eine Lizenz erwerben, die ausreichend Verbindungen für die Anzahl von möglichen Clients bietet. Wenn Sie in LabVIEW 8.5.1 oder älteren Versionen Ihre Lizenz für Netzwerk-Frontpanel erweitert haben, müssen Sie Ihre Seriennummer zum Aktivieren einer neuen gleichwertigen Lizenz im NI-Lizenzmanager verwenden.

Gemeinsame Komponenten des Application Builders

LabVIEW installiert eine Komponente zum Erstellen von DLLs, die von allen Versionen von LabVIEW auf dem lokalen Rechner gemeinsam verwendet wird. Wenn Sie nach der Installation einer LabVIEW-Version ab 8.6 eine ältere LabVIEW-Version installieren, wird diese gemeinsame Komponente durch die ältere Version ersetzt. Wenn Sie dann eine DLL in der aktuellen Version von LabVIEW erstellen, erhalten Sie eine Fehlermeldung, da der gemeinsamen Komponente der Funktionsumfang fehlt, auf den LabVIEW-Versionen ab 8.6 angewiesen sind. Zum Beheben dieses Problems installieren Sie erneut eine LabVIEW-Version ab 8.6.

Speichern von passwortgeschützten VIs für vorherige Versionen

Zum Speichern eines passwortgeschützten VIs für eine vorige Version von LabVIEW müssen Sie in LabVIEW-Versionen ab 8.6 ein Passwort eingeben. Sie können das Passwort auch programmatisch als Eingang der Funktion "VI-Referenz öffnen" eingeben.

Upgrade von LabVIEW 8.2

Beim Upgrade von LabVIEW 8.2 auf LabVIEW 2009 können die nachfolgend beschriebenen Kompatibilitätsprobleme auftreten. Alle weiteren potenziellen Probleme beim Upgrade von LabVIEW sind in den Abschnitten [Upgrade von LabVIEW 8.5](#) und [Upgrade von LabVIEW 8.6](#) beschrieben.



Hinweis Zusätzliche Informationen zu weiteren Problemen, die beim Upgrade von LabVIEW 8.2.x auftreten können, erhalten Sie auf der Website von National Instruments ni.com/info nach Eingabe des Infocodes *rdgup82*.

Unterstützte Plattformen

Im Hinblick auf die Plattformkompatibilität gibt es bei LabVIEW-Versionen ab 8.5 folgende Änderungen:

- LabVIEW-Versionen ab einschließlich 8.5 arbeiten mit der 32- und 64-Bit-Version von Windows Vista.
- LabVIEW 8.5 und 8.5.1 unterstützen Macintosh-Rechner mit Prozessoren von Intel und PowerPC. Die LabVIEW-Versionen ab einschließlich 8.6 unterstützen keine Macintosh-Computer mit PowerPC-Prozessoren.

Systemvoraussetzungen

(Windows) Für LabVIEW 8.5 ist ein Festplattenspeicher von mindestens 1,2 GB erforderlich. Für LabVIEW 2009 ist ein Festplattenspeicher von mindestens 1,6 GB erforderlich.

(Mac OS) Für die Minimalversion von LabVIEW 8.5 sind mindestens 502 MB Festplattenspeicher und für die vollständige Installation 734 MB erforderlich. Für die Minimalversion von LabVIEW 2009 sind mindestens 563 MB Festplattenspeicher erforderlich. Für die vollständige Installation benötigt die Version 1,2 GB.

(Linux) Für die Minimalversion von LabVIEW 8.5 sind mindestens 450 MB Festplattenspeicher und für die vollständige Installation 640 MB erforderlich. Für die Minimalversion von LabVIEW 2009 sind mindestens 630 MB Festplattenspeicher erforderlich und für die vollständige Installation 835 MB.

Kompatibilitätsprobleme bei Windows Vista

LabVIEW-Versionen ab 8.5 können mit der 32- und 64-Bit-Version von Windows Vista verwendet werden, jedoch mit folgenden Einschränkungen:

Die VIs "Eingangs-Port" und "Ausgangs-Port" werden nicht auf der Palette **Funktionen** angezeigt, da sie mit jeder Schnittstelle des Systems arbeiten, was aus Sicherheitsgründen unter Windows Vista nicht empfohlen wird.

- **(Windows Vista)** Die VI-Komponenten werden zwar ordnungsgemäß installiert, erscheinen aber im Protokoll des Windows Defenders als nicht signiert. Die VIs werden ordnungsgemäß ausgeführt.
- **(Windows Vista 64 Bit)** Diese VIs geben den Fehler -4850 aus.

Änderungen an der Funktionsweise von VIs und Funktionen

Die Funktionsweise der nachfolgenden VIs und Funktionen hat sich in LabVIEW-Versionen ab 8.5 geändert.

Verbesserungen an den Analyse-VIs und -Funktionen

In jeder Version von LabVIEW verbessert National Instruments viele der Algorithmen in LabVIEW- und C-Funktionen. National Instruments aktualisiert LabVIEW für die Verwendung der neuesten Compiler. Diese Verbesserungen und andere Änderungen an Hard- und Software können möglicherweise zu unterschiedlichen Ergebnissen zwischen LabVIEW 8.2 oder älteren Versionen und LabVIEW-Versionen ab 8.5 führen. Beim Vergleich von Fließkommazahlen doppelter Genauigkeit stellen Sie möglicherweise geringe Unterschiede in der Ordnung $1E-16$ fest. Weitere Informationen zum Vergleich von Fließkommazahlen finden Sie auf der Website von National Instruments ni.com/info nach Eingabe des Infocodes `exiigr`.

VIs zur Signalverarbeitung

Im VI "Übergangsmessungen" wurde der Ausgang **Vorschwingung** in **Vor Übergang** umbenannt. Außerdem arbeitet der Ausgang jetzt mit Clustern und nicht mehr wie bisher mit 64-Bit-Fließkommazahlen mit doppelter Genauigkeit. Der Ausgang **Überschwingen** wurde in **Nach Übergang** umbenannt. Außerdem arbeitet der Ausgang jetzt mit Clustern und nicht mehr wie bisher mit 64-Bit-Fließkommazahlen mit doppelter Genauigkeit.

Hyperbelfunktionen

Folgende Hyperbelfunktionen haben sich in LabVIEW-Versionen ab 8.5 geändert:

- Die Funktion "Areacossinus Hyperbolicus" gibt NaN aus, wenn der Eingangswert eine reelle Zahl ist, die sich außerhalb des Bereichs der Funktion befindet.
- Die Funktion "Areasecans" gibt NaN aus, wenn der Eingangswert eine reelle Zahl ist, die sich außerhalb des Bereichs der Funktion befindet.

VI's und Funktionen für Bibliotheken und ausführbare Anwendungen

Beim Knoten zum Aufruf externer Bibliotheken muss beim Erstellen eines Pascal-String-Zeigers im Blockdiagramm der String-Eingang verbunden werden. Zum Erstellen eines C-String-Zeigers müssen Sie entweder den String-Eingang verbinden oder im Pulldown-Menü **Mindestgröße** auf der Registerkarte **Parameter** des Dialogfelds **Aufruf externer Bibliotheken** eine String-Größe angeben. Das VI kann nicht ohne Angabe von Werten für die Strings ausgeführt werden.

Polymorphe VI-Anschlüsse, die 64-Bit- und numerische Datentypen doppelter Genauigkeit unterstützen

LabVIEW wandelt numerische Daten erweiterter Genauigkeit in Daten doppelter Genauigkeit um, wenn Sie die Daten mit einem polymorphen VI verbinden, das sowohl Daten doppelter Genauigkeit als auch 64-Bit-Integer unterstützt. Dadurch wird ein Teil der gebrochenen Komponente der ursprünglichen Daten beibehalten.

Allgemeine Änderungen an der Funktionsweise von VI's und Funktionen

In LabVIEW-Versionen ab 8.5 gibt es folgende allgemeine Änderungen an der Funktionsweise von VI's und Funktionen:

- Die VI's "Geräteattribut abrufen" und "Geräteattribut setzen" sind nicht mehr in LabVIEW enthalten. Wenn Sie eines dieser VI's in einer Applikation verwenden, ersetzen Sie es durch einen Eigenschaftsknoten der Palette **VISA: Fortgeschritten**, um den Funktionsumfang beizubehalten.
- Der Parameter **Alle Ordner** des VI's "Rekursive Dateiliste" kann Verknüpfungen zu Ordnern enthalten, aber das VI greift nicht auf diese zurück.

Änderungen an den Ausgangstunneln von Case-Strukturen

Die Funktionsweise der Ausgangstunnel von Case-Strukturen wurde zwischen LabVIEW 8.2 und LabVIEW 8.5 geändert. In LabVIEW 8.2 wird der Datentyp eines Ausgangstunnels einer Case-Struktur durch Verwendung eines Datentyps festgelegt, den alle Cases in der Struktur verarbeiten können (einschließlich der Cases, die nie ausgeführt werden). In LabVIEW 8.5 bis 8.6.x wird der Datentyp des Cases verwendet, der über einen Ausgangstunnel mit einer Konstante verbunden ist.

In LabVIEW 2009 funktioniert der Ausgangstunnel von Case-Strukturen wieder genauso wie in LabVIEW 8.2. Der Datentyp eines Ausgangstunnels wird also so gewählt, dass er von allen Cases in der Struktur verarbeitet werden kann (einschließlich der Cases, die nie ausgeführt werden). Die *LabVIEW-Hilfe* enthält weitere Informationen zur Funktionsweise der Ausgangstunnel von Case-Strukturen in LabVIEW 2009.

Änderungen der Funktionsweise von Eigenschaften, Methoden und Ereignissen

Die Funktionsweise folgender Eigenschaften, Methoden und Ereignisse hat sich in LabVIEW-Versionen ab 8.5 geändert:

- Die Eigenschaft "Datenbindung:Pfad" der Klasse "Element" kann gelesen und geschrieben werden und lässt sich einstellen, während das VI ausgeführt wird. Zum Schreiben muss das Element an eine NI-Publish-Subscribe-Protocol-URL (NI-Protokoll zum Senden und Empfangen) gebunden werden.

- Die Eigenschaft "Ziel-CPU" der Klasse "Applikation" enthält den Wert `AMD/Intel x64`.
- Die Eigenschaft "Ziel:Betriebssystem" der Klasse "Applikation" enthält die Werte `Windows x64` und `Linux x64`.
- Die Methode "Punkt nach Zeile/Spalte" der Klasse "Baumstruktur" gibt das Tag `TREE_COLUMN_HEADERS` aus, wenn Sie einen Punkt der Spaltenkopfzeile der Baumstruktur verbinden.
- Die Methode "LabVIEW-Klasse:Erstellen" enthält einen Eingang für Namen. Wenn Sie den Eingang **Name** nicht verbinden, wird der Benutzer während der Laufzeit aufgefordert, den Namen der Klasse anzugeben.
- Die Methoden "Elementwert:Abfragen [Variant]", "Elementwert:Lesen [Serialisiert]", "Elementwert:Festlegen [Variant]" und "Elementwert:Setzen [Serialisiert]" löschen bei der Suche nach Elementen keine Leerzeichen mehr am Zeilenanfang und -ende.

Veraltete Eigenschaften, Methoden und Ereignisse

Die folgenden Eigenschaften, Methoden und Ereignisse werden von LabVIEW-Versionen ab 8.5 nicht mehr unterstützt:

- Die Eigenschaft "Standardinstanz" der Klasse "LV-Klassenbibliothek". Arbeiten Sie stattdessen mit dem VI "Standardwert der LV-Klasse lesen".
- Die Eigenschaft "Geometrie" der Klasse "Szenenobjekt". Verwenden Sie stattdessen die Eigenschaft "Darstellbar".
- Die Eigenschaft "Gitterfarben" der Klasse "GraphDiagramm". Verwenden Sie stattdessen die Eigenschaft "Gitterfarben" der Klasse "GraphSkala".
- Die Eigenschaft "Gitterfarben:X-Farbe" der Klasse "GraphDiagramm". Arbeiten Sie stattdessen mit den Eigenschaften "Gitterfarben:Hauptgitterfarbe" und "Gitterfarben:Feingitterfarbe".
- Die Eigenschaft "Gitterfarben:X-Farbe" der Klasse "GraphDiagramm". Arbeiten Sie stattdessen mit den Eigenschaften "Gitterfarben:Hauptgitterfarbe" und "Gitterfarben:Feingitterfarbe".
- Die Eigenschaft "Gitterfarben:Y-Farbe" der Klasse "GraphDiagramm". Arbeiten Sie stattdessen mit den Eigenschaften "Gitterfarben:Hauptgitterfarbe" und "Gitterfarben:Feingitterfarbe".
- Die Eigenschaft "Legende:Angezeigte Plots" der Klasse "Signalverlaufdiagramm". Verwenden Sie stattdessen die Eigenschaft "Legende:Zeilenanzahl".
- Die Eigenschaft "Legende:Angezeigte Plots" der Klasse "Signalverlaufsgraph". Verwenden Sie stattdessen die Eigenschaft "Legende:Zeilenanzahl".
- Die Eigenschaft "Pixelbreite" der Klasse "Listenfeld". Verwenden Sie stattdessen die Eigenschaft "Maße:Bereichsbreite".
- Die Eigenschaft "Bildlaufleiste sichtbar" der Klasse "Bild". Verwenden Sie stattdessen die Eigenschaften "Horizontale Bildlaufleiste sichtbar" und "Vertikale Bildlaufleiste sichtbar".
- Die Eigenschaft "Geometrische Form festlegen" der Klasse "Szenenobjekt". Verwenden Sie stattdessen die Methode "Darstellbares Objekt festlegen".
- Die Methode "Szene:Geometrie:Neues Netz" der Klasse "Applikation". Verwenden Sie stattdessen die Methode "Szene:Darstellbares Objekt:Geometrie:Neues Netz".
- Das Ereignis "Ziehen begonnen" der Klasse "Element". Verwenden Sie stattdessen das Ereignis "Ziehen begonnen" der entsprechenden Elementklasse.
- Das Ereignis "Ziehen begonnen?" der Klasse "Element". Verwenden Sie stattdessen das Ereignis "Ziehen begonnen?" der entsprechenden Elementklasse.

Umbenannte Eigenschaften, Methoden und Ereignisse

Folgende Eigenschaften, Methoden und Ereignisse wurden in LabVIEW-Versionen ab 8.5 umbenannt:

Klasse	Name in LabVIEW 8.2	Name in LabVIEW 8.5	Typ
Element für absolute Zeit, numerisches Element	Bei Bereichsüberschreitung	Werte außerhalb der Grenzen	Eigenschaft
Element für absolute Zeit, numerisches Element	Bei Bereichsüberschreitung:Inkrement	Werte außerhalb der Grenzen:Inkrement	Eigenschaft
Element für absolute Zeit, numerisches Element	Bei Bereichsüberschreitung:Maximum	Werte außerhalb der Grenzen:Maximum	Eigenschaft
Element für absolute Zeit, numerisches Element	Bei Bereichsüberschreitung:Minimum	Werte außerhalb der Grenzen:Minimum	Eigenschaft
Applikation	Bibliothek:Version der Projektbibliotheksdatei ermitteln	Bibliothek:LabVIEW-Datei-Version ermitteln	Methode
Applikation	Szene:Geometrie:Neue Box	Szene:Darstellbares Objekt:Geometrie:Neuer Würfel	Methode
Applikation	Szene:Geometrie:Neuer Kegel	Szene:Darstellbares Objekt:Geometrie:Neuer Kegel	Methode
Applikation	Szene:Geometrie:Neuer Zylinder	Szene:Darstellbares Objekt:Geometrie:Neuer Zylinder	Methode
Applikation	Szene:Geometrie:Neues Höhenfeld	Szene:Darstellbares Objekt:Geometrie:Neues Höhenfeld	Methode
Applikation	Szene:Geometrie:Neues Netz	Szene:Darstellbares Objekt:Geometrie:Neues Netz	Methode
Applikation	Szene:Geometrie:Neue Kugel	Szene:Darstellbares Objekt:Geometrie:Neue Kugel	Methode
Application (ActiveX)	LibraryGetProjectLibFileVersion	LibraryGetFileLVVersion	Methode
Digital, NumerischerText und Skalierung	Format und Genauigkeit	Anzeigeformat	Eigenschaft
Digital, NumerischerText und Skalierung	Format und Genauigkeit:Format	Anzeigeformat:Format	Eigenschaft
Digital, NumerischerText und Skalierung	Format und Genauigkeit:Genauigkeit	Anzeigeformat:Genauigkeit	Eigenschaft
Digitalwerttabelle	Spaltentitel sichtbar	Signalnummer sichtbar	Eigenschaft
Digitalwerttabelle	Zeilentitel sichtbar	Übergänge sichtbar	Eigenschaft
Szenengraph-Anzeige und Szenenfenster	Farbe löschen	Hintergrundfarbe	Eigenschaft
Szenenobjekt	Geometrische Form festlegen	Darstellbares Objekt festlegen	Methode

Klasse	Name in LabVIEW 8.2	Name in LabVIEW 8.5	Typ
VI	Anschlussfeld	Anschlussfeld:Festlegen	Eigenschaft

Änderungen an der Funktionsweise von LabVIEW-MathScript (Windows, Nicht im Basispaket enthalten)

In LabVIEW-Versionen ab 8.5 gibt es folgende Änderungen an LabVIEW MathScript:

- Änderungen an der Suchpfadliste oder dem Arbeitsverzeichnis, die mit Hilfe der folgenden MathScript-Funktionen vorgenommen wurden, gelten nur für die aktuelle Instanz des **LabVIEW-MathScript-Fensters** oder des MathScript-Knotens, von dem die Funktion aufgerufen wird:

- `addpath`
- `cd`
- `path`
- `rmpath`

Wenn Sie das **LabVIEW-MathScript-Fenster** schließen oder die Ausführung des VIs beenden, das den MathScript-Knoten enthält, werden die Suchpfadliste und das Arbeitsverzeichnis wieder auf die Voreinstellungen zurückgesetzt.

- Die Syntax für die `qz`-Funktion hat sich von `[q, z, alpha, beta, evec] = qz(a, b)` in `[S, T, Q, Z, R, L] = qz(A, B, type)` geändert.

Symbole für LabVIEW-Klassen

Wenn Sie in LabVIEW 8.2 ein Symbol für eine LabVIEW-Klasse erstellt haben und dieses beim Einfügen eines Klasselements in ein Blockdiagramm angezeigt werden soll, muss das Klassensymbol verkleinert werden, so dass die Klassenmaske das Symbol nicht verdeckt. Das Bild sollte maximal 32 Pixel breit und 19 Pixel hoch sein.

Öffnen von LLBs in LabVIEW

Die Option **Anzeige von LLBs im Windows Explorer möglich** auf der Seite **Umgebung** im Dialogfeld **Optionen** gibt es nicht mehr. LabVIEW öffnet LLBs im **LLB-Manager**. Für weitere Informationen zum Öffnen von LLBs besuchen Sie die Website von National Instruments ni.com/info und geben Sie den Infocode `exvfc5` ein.

Beschränkung der Prioritätslevel von zeitgesteuerten Schleifen

In LabVIEW 8.2.x und Vorgängerversionen kann bis zu 2^{32} für das Prioritätslevel einer zeitgesteuerten Schleife ausgewählt werden. LabVIEW-Versionen ab 8.5 unterstützen nur Prioritätslevel kleiner als 65.535.

Datentyp "Signalverlauf"

Wenn bei einem Array aus Signalverläufen ein größerer Index angegeben wird als Elemente im Array enthalten sind, wird der ausgegebene Signalverlauf dennoch normal mit `dt = 1` dargestellt (bisher wurde ein Signalverlauf mit `dt = 0` ausgegeben). Das Gleiche gilt bei einer For-Schleife mit einem Ausgangstunnel für Skalarwerte, die null Mal ausgeführt wird.

Anpassen von Enums

In LabVIEW-Versionen ab 8.5 werden Enums, die außerhalb des zulässigen Bereichs liegen, auf den Höchst- oder Mindestwert gesetzt, der in dem Bereich zulässig ist. In älteren LabVIEW-Versionen werden solche Enums auf 0 gesetzt.

Upgrade von LabVIEW 8.0

Beim Upgrade von LabVIEW 8.0 auf LabVIEW 2009 können die nachfolgend genannten Kompatibilitätsprobleme auftreten. Weitere Probleme, die beim Upgrade von LabVIEW auftreten können, sind in den Abschnitten [Upgrade von LabVIEW 8.2](#), [Upgrade von LabVIEW 8.5](#) und [Upgrade von LabVIEW 8.6](#) beschrieben.



Hinweis Zusätzliche Informationen zu weiteren Problemen, die beim Upgrade von LabVIEW 8.0 auftreten können, erhalten Sie auf der Website von National Instruments ni.com/info nach Eingabe des Infocodes `rdg81v`.

Unterstützte Plattformen

Im Hinblick auf die Plattformkompatibilität gibt es bei LabVIEW-Versionen ab 8.2 folgende Änderungen:

- LabVIEW-Versionen ab 8.2 arbeiten weder mit Mac OS X 10.3.8 noch mit Vorgängerversionen davon.
- LabVIEW 8.2 arbeitet mit Macintosh-Rechnern mit Intel-Prozessor. Weitere Informationen zur Unterstützung von Macintosh-Rechnern finden Sie auf der Website von National Instruments ni.com/info nach Eingabe des Infocodes `macintel`. Die LabVIEW-Versionen ab einschließlich 8.6 unterstützen keine Macintosh-Computer mit PowerPC-Prozessoren.

Systemvoraussetzungen

(Windows) Für LabVIEW 8.2 ist ein Festplattenspeicher von mindestens 1,2 GB erforderlich. Für LabVIEW 2009 ist ein Festplattenspeicher von mindestens 1,6 GB erforderlich.

(Mac OS) Für die Minimalversion von LabVIEW 8.2 sind mindestens 500 MB Festplattenspeicher und für die vollständige Installation 700 MB erforderlich. Für die Minimalversion von LabVIEW 2009 sind mindestens 563 MB Festplattenspeicher erforderlich. Für die vollständige Installation benötigt die Version 1,2 GB.

(Linux) Für die Minimalversion von LabVIEW 8.2 sind mindestens 430 MB Festplattenspeicher und für die vollständige Installation 620 MB erforderlich. Für die Minimalversion von LabVIEW 2009 sind mindestens 630 MB Festplattenspeicher erforderlich und für die vollständige Installation 835 MB.

Änderungen an der Funktionsweise von VIs und Funktionen

Die Funktionsweise der nachfolgenden VIs und Funktionen hat sich in LabVIEW-Versionen ab 8.2 geändert.

Kommunikation zwischen Applikationsinstanzen

In LabVIEW-Versionen ab 8.2 können die Funktionen "Queue anfordern", "Melder anfordern", "Benutzerereignis erstellen", "Semaphor erstellen" und "Rendezvous erstellen" nicht mehr zwischen verschiedenen LabVIEW-Applikationsinstanzen kommunizieren. Wenn Sie eine Queue-, Melder-, Benutzerereignis-, Semaphor- oder Rendezvous-Referenz in einer Applikationsinstanz anfordern oder erstellen, kann diese nicht in einer anderen Applikationsinstanz verwendet werden.

VI "Rücktransformation der Eigenvektoren"

An den Eingängen **Unterer Index**, **Oberer Index** und **Skalierung** des VIs "Rücktransformation der Eigenvektoren" müssen Angaben gemacht werden, damit das VI funktioniert.

Funktion "DataSocket: Schreiben"

In LabVIEW 8.0.1 hat sich das Standardverhalten der Funktion "DataSocket: Schreiben" in "asynchron" geändert. Wenn sowohl LabVIEW 8.0 als auch LabVIEW 8.2 oder eine neuere Version installiert sind, gibt das Beispiel-VI "DataSocket API Client" im Verzeichnis `labview\examples\Shared Variable`

bei Anhalten des VIs einen Fehler aus. Sie müssen daher LabVIEW 8.0 auf 8.0.1 aktualisieren, um das Beispiel in einer LabVIEW-Version ab 8.2 ausführen zu können.

Datei-I/O-VIs

Die VIs "In Tabellenkalkulationsdatei schreiben" und "Aus Tabellenkalkulationsdatei lesen" sind polymorph. Das VI "In Tabellenkalkulationsdatei schreiben" passt sich an den mit dem Eingang **Format** verbundenen Wert an. Das VI "Aus Tabellenkalkulationsdatei lesen" enthält die Instanzen "DBL", "I64" und "String".

Funktion "GPIB: Status"

In LabVIEW 8.0 wurde die Funktion "GPIB: Status" nicht ausgeführt, wenn **Fehler (Eingang)** einen Fehler empfangen hat. In LabVIEW ab Version 8.2 arbeitet die Funktion "GPIB: Status" unabhängig von Eingangsfehlern.

Histogramm-VI

Der Standardwert für den Eingang **Intervalle** des VIs "Histogramm" wurde in 10 geändert.

Funktion "VI-Referenz öffnen"

Die Voreinstellung für den Eingang **Optionen** der Funktion "VI-Referenz öffnen" ist die Anzeige eines Dialogfelds, wenn nach fehlenden SubVIs des referenzierten VIs gesucht wird. Der Benutzer wird nicht aufgefordert, VIs zu suchen, die LabVIEW nicht automatisch findet. Es gibt jedoch eine neue Einstellung, 0x20, bei der das Dialogfeld zum Suchen fehlender SubVIs nicht angezeigt wird. Zum Auffordern des Benutzers, nach den fehlenden SubVIs zu suchen, muss die Einstellung 0x10 verwendet werden.

VI "Nullstellen eines Polynoms"

Wenn **P(x)** ein konstanter Wert ungleich 0 ist, gibt das VI "Nullstellen eines Polynoms" keinen Fehler aus. Wenn **P(x)** aber gleich 0 ist, gibt das VI "Nullstellen eines Polynoms" den Fehler -20111 aus. Es können nicht alle Polynomkoeffizienten am Eingang dieses VIs Null sein.

VI "Rampenmuster"

Wenn der Eingang **Samples** im VI "Rampenmuster" auf 1 eingestellt ist und **Ende ausschließen?** TRUE ist, wird ein Array mit einem **Start**-Element und kein Fehler ausgegeben. In LabVIEW 8.0 hat das VI unter diesen Bedingungen einen Fehler ausgegeben.

VI "Registry: Wert lesen (einfach)"

In LabVIEW 8.0 traten bei der Formatierung des Strings `REG_MULTI_SZ`, den das VI für ein serialisiertes Array aus Strings benötigt, Fehler auf. Daher musste ein Parser erstellt werden, der diesen Datentyp für das VI "Registry: Wert lesen (einfach)" verarbeiten konnte. In LabVIEW-Versionen ab 8.2 gibt das VI "Registry: Wert lesen (einfach)" diesen Datentyp im gleichen Format aus wie das VI "Registry: Wert schreiben (einfach)". Sie müssen keinen eigenen Parser mehr hinzufügen. Wenn Sie Ihren eigenen Parser mit diesen VIs in LabVIEW-Versionen ab 8.2 verwenden, gibt das VI "Registry: Wert lesen (einfach)" ungültige Daten aus.

VI "Signalverläufe erneut abtasten (einmalig)"

Der Standardwert des Eingangs **Offenes Intervall?** des VIs "Signalverläufe erneut abtasten (einmalig)" wurde von TRUE auf FALSE geändert, wodurch ein geschlossenes Intervall ausgewählt wird. Wenn Sie bestehenden Code nicht entsprechend anpassen, gibt das VI ggf. nicht das erwartete Ergebnis aus.

Audio-VIs

In den VIs "Audioaufnahme lesen" und "Audiodatei lesen (einfach)" gibt die Komponente "t0" des Ausgangs "Daten" den t0-Wert des ersten Samples aus. Der t0-Wert, bei dem das erste Sample gelesen wird, ist ein geschätzter Wert.

Das VI "Audioausgabe beenden" muss nicht mehr aufgerufen werden, um die Audioausgabe in einem fortlaufenden Audiotask zu beenden.

Das VI "Audioausgabe abwarten" arbeitet nun in den Modi **Kontinuierlich** und **Endliche Anzahl**.

Signalverlaufs-VIs

In LabVIEW ab Version 8.2 wurden die folgenden Signalverlaufs-VIs geändert:

- **"Einfache Trigger-Erkennung"**—In beiden Instanzen dieses VIs wurde der Eingang **Steigung** in **Trigger-Polarität** geändert.
- **"Signalverlaufsabschnitt lesen"**—Enthält folgende Instanzen: "WDT: Signalverlaufsabschnitt lesen (DBL)", "WDT: Signalverlaufsabschnitt lesen (CDB)", "WDT: Signalverlaufsabschnitt lesen (EXT)", "WDT: Signalverlaufsabschnitt lesen (I16)", "WDT: Signalverlaufsabschnitt lesen (I32)", "WDT: Signalverlaufsabschnitt lesen (I8)" und "WDT: Signalverlaufsabschnitt lesen (SGL)". Die Option **Absolute Zeit** für den Eingang **Format von Start/Dauer** ist nicht mehr vorhanden. Der Eingang **Start** wurde in **Start-Samples/Zeit** und der Ausgang **Tatsächlicher Start** in **Tatsächlich(e) Start-Samples/Zeit** geändert.
- **"Zeit-Array für Signalverlauf lesen"**—Der Datentyp des Ausgangs **X-Array** wurde von "numerischer Fließkommawert doppelter Genauigkeit" in "Zeitstempel" geändert.
- **"XY-Wert abfragen"**—Dieses polymorphe VI wurde zusammen mit seinen Instanzen in "XY-Wert abfragen" umbenannt. Das VI "XY-Wert abfragen" enthält jetzt den Ausgang **X-Wert**. Der Ausgang **Datenwert** wurde in **Y-Wert** geändert.
- **"Anzahl der Signalverlaufswerte"**—Dieses polymorphe VI enthält folgende Instanzen: "WDT: Anzahl der Signalverlaufswerte (DBL)", "Anzahl der Signalverlaufswerte (CDB)", "WDT: Anzahl der Signalverlaufswerte (EXT)", "WDT: Anzahl der Signalverlaufswerte (I16)", "WDT: Anzahl der Signalverlaufswerte (I32)", "WDT: Anzahl der Signalverlaufswerte (I8)" und "WDT: Anzahl der Signalverlaufswerte (SGL)".
- **"Signalverlauf aus Datei lesen"**—Gibt bei einem EOF-Fehler im Ausgang **Fehler (Ausgang)** den Fehlerstatus TRUE aus.
- **"Abschnitt ersetzen"**—Der Eingang **Start** wurde in **Start-Samples/Zeit** und der Ausgang **Tatsächlicher Startwert** in **Tatsächlich(e) Start-Samples/Zeit** geändert.
- **"Nach Digitalmuster suchen"**—Der Eingang **Start** wurde in **Startindex/Zeit** geändert.
- **"Signalverlauf suchen"**—Der Datentyp der Ausgänge **Zeitpunkt der besten Anpassung** und **Zeitpunkt der Anpassung** wurde von numerischem Fließkommawert doppelter Genauigkeit in Zeitstempel geändert.
- **"Signalverlauf - Minimum und Maximum"**—Der Datentyp der Ausgänge **Min. Zeit** und **Max. Zeit** wurde von numerischem Fließkommawert doppelter Genauigkeit in Zeitstempel geändert.
- **"Signalverlauf in XY-Paare"**—Der Datentyp des x-Elements des Ausgangs **XY-Paare** wurde von numerischem Fließkommawert doppelter Genauigkeit in Zeitstempel geändert.

Änderungen der Funktionsweise von Eigenschaften, Methoden und Ereignissen

Die Funktionsweise folgender Eigenschaften, Methoden und Ereignisse hat sich in LabVIEW-Versionen ab 8.2 geändert:

- Per Voreinstellung wird der Benutzer durch den Eingang **Optionen** der ActiveX-Methode "GetVIReference" aufgefordert, fehlende SubVIs bereitzustellen. Es gibt jedoch eine neue Einstellung, 0×20, bei der das Dialogfeld **Suchen** nicht angezeigt wird.

- Die Methode "Element hinzufügen" der Klasse "ProjektElement" gibt einen Fehler aus, wenn Sie versuchen, eine Umgebungsvariable zu einer Bibliothek hinzuzufügen, die nicht in einem Projekt geöffnet ist.
- Wenn der Eingang **Ref automatisch schließen** der Methode "VI ausführen" TRUE ist und die Methode einen Fehler ausgibt, wird die Referenz nicht geschlossen.
- Die LabVIEW-Umgebung kann auf vereinfachtes Chinesisch eingestellt werden. Setzen Sie dazu die Eigenschaft "Applikation:Sprache" auf zh-cn.
- In LabVIEW 8.0 leiten .NET-Methoden, die Array-Datentypen anhand von Referenzen verarbeiten, alle Daten als Referenzdatentyp weiter. .NET-Methoden in LabVIEW-Versionen ab 8.2, die Array-Datentypen mittels Referenzen verarbeiten, leiten alle Daten mit ihrem individuellen Datentyp weiter.
- Die Eigenschaft "Position bearbeiten" der Klassen "Digitalwerttabelle", "mehrspaltigesListenfeld", "Tabelle" und "Baumstruktur" gibt die Werte (-2, -2) aus. Das bedeutet, dass der Benutzer keine Änderungen am Text des Elements vornimmt. Die Eigenschaft "Zeile bearbeiten" der Klasse "Listenfeld" gibt den Wert -2 aus. Das bedeutet, dass der Benutzer keine Änderungen am Text des Elements vornehmen kann.
- In LabVIEW 8.0 gilt die Eigenschaft "Panel-Aktualisierungen verzögern" nicht für Frontpanels in einem Unterpanel. In LabVIEW-Versionen ab 8.2 gilt die Eigenschaft "Panel-Aktualisierungen verzögern" auch für Unterpanel.
- Die Ereignisse "Applikationsinstanz beenden" und "Applikationsinstanz beenden?" wurden durch "Applikationsinstanz schließen" und "Applikationsinstanz schließen?" ersetzt. Wenn Sie das Ereignis "Applikationsinstanz schließen" in einem VI verwenden, das außerhalb eines **LabVIEW-Projekts** ausgeführt wird, erzeugt LabVIEW ein Ereignis, wenn Sie das Programm über die Benutzeroberfläche oder programmatisch beenden. LabVIEW erzeugt das Ereignis "Applikationsinstanz schließen?", wenn Sie das Programm über die Benutzeroberfläche schließen. Wenn Sie die Ereignisse "Applikationsinstanz schließen" und "Applikationsinstanz schließen?" für ein VI in einem LabVIEW-Projekt registrieren, erzeugt LabVIEW die Ereignisse, wenn die Applikationsinstanz geschlossen oder LabVIEW beendet wird.

Veraltete Eigenschaften, Methoden und Ereignisse

Die folgenden Eigenschaften, Methoden und Ereignisse werden von LabVIEW-Versionen ab 8.2 nicht mehr unterstützt:

- LabVIEW-Versionen ab 8.2. unterstützen die Anschlussfeld-Eigenschaft nicht mehr.
- LabVIEW 8.x unterstützt die Datentyp-Eigenschaft in der Klasse "Variable" nicht mehr. Verwenden Sie stattdessen die Eigenschaft "Datentyp (Variant)" in der Klasse "Variable".

Umbenannte Eigenschaften, Methoden und Ereignisse

Folgende Eigenschaften, Methoden und Ereignisse wurden in LabVIEW-Versionen ab 8.2 umbenannt:

Klasse	Name in LabVIEW 8.0	Name in LabVIEW-Versionen ab 8.2	Typ
Applikation	Verbindung mit Slave beenden	LVRT:Verbindung mit Slave beenden	Methode
Applikation	Applikation beenden	Applikationsinstanz schließen	Ereignis
Applikation	Applikation beenden?	Applikationsinstanz schließen?	Ereignis
Intensitätsgraph, Mischsignalgraph und Signalverlaufsgraph	Cursor-Palette sichtbar	Cursor-Legende sichtbar	Eigenschaft
Bibliothek	Bibliothek-Tag löschen	Bibliothek-Tag:Löschen	Methode

Klasse	Name in LabVIEW 8.0	Name in LabVIEW-Versionen ab 8.2	Typ
Bibliothek	Symbol ermitteln	Symbol:Ermitteln	Methode
Bibliothek	Bibliotheks-Tag ermitteln	Bibliotheks-Tag:Ermitteln	Methode
Bibliothek	Bibliotheks-Tag - Namen ermitteln	Bibliotheks-Tag:Namen ermitteln	Methode
Bibliothek	Sperrzustand ermitteln	Sperrzustand:Ermitteln	Methode
Bibliothek	Quellbereich ermitteln	Quellbereich:Ermitteln	Methode
Bibliothek	Speichern	Speichern:Bibliothek	Methode
Bibliothek	Kopie speichern	Speichern:Kopie	Methode
Bibliothek	Symbol festlegen	Symbol:Festlegen	Methode
Bibliothek	Bibliotheks-Tag festlegen	Bibliotheks-Tag:Festlegen	Methode
Bibliothek	Sperrzustand festlegen	Sperrzustand:Festlegen	Methode
Bibliothek	Quellbereich festlegen	Quellbereich:Festlegen	Methode
Listenfeld, mehrspaltiges Listenfeld und Baumstruktur	Ziehen/Ablegen:Verschieben von Objekten möglich	Ziehen/Ablegen:Ziehen möglich	Eigenschaft
Projektoobjekte	Tag löschen	Tag:Löschen	Eigenschaft
Projektoobjekte	Tag lesen	Tag:Tag ermitteln	Eigenschaft
Projektoobjekte	Tag-Namen ermitteln	Tag:Namen ermitteln	Eigenschaft
Projektoobjekte	XML-Tag lesen	Tag:XML-Tag ermitteln	Eigenschaft
Projektoobjekte	Tag festlegen	Tag:Tag festlegen	Eigenschaft
Projektoobjekte	XML-Tag festlegen	Tag:XML-Tag festlegen	Eigenschaft
Projektoobjekte	Bibliotheksobjektyp-String	Bibliotheksobjektyp:String	Eigenschaft
Projektoobjekte	Bibliotheksobjektyp	Bibliotheksobjekt:Typ	Eigenschaft

Änderungen am Application Builder

In LabVIEW ab Version 8.2 kann der Inhalt einer EXE- oder DLL-Datei nicht durch Umbenennen der Erweiterung `.lib` angezeigt werden. Ebenso wenig können Sie auf ein VI in einer EXE- oder DLL-Datei zugreifen, indem Sie den Pfad zum VI außerhalb der Datei festlegen. Weitere Informationen zur Anzeige von und zum Zugriff auf Applikationen und DLLs finden Sie auf der Website von National Instruments ni.com/info nach Eingabe des Infocodes `exjk3b`.

Upgrade von LabVIEW 7.x

Bei einem Upgrade von LabVIEW 7.x auf LabVIEW 2009 können die nachfolgend aufgeführten Kompatibilitätsprobleme auftreten. Weitere Probleme, die beim Upgrade von LabVIEW auftreten können, sind in den Abschnitten [Upgrade von LabVIEW 8.0](#), [Upgrade von LabVIEW 8.2](#), [Upgrade von LabVIEW 8.5](#) und [Upgrade von LabVIEW 8.6](#) beschrieben.



Hinweis Zusätzliche Informationen zu weiteren Problemen, die beim Upgrade von LabVIEW 8.0 auftreten können, erhalten Sie auf der Website von National Instruments ni.com/info nach Eingabe des Infocodes `rd70un`. Welche Probleme beim Upgrade von LabVIEW 7.1 auftreten können, erfahren Sie nach Eingabe des Infocodes `exucmeDE`.

Unterstützte Plattformen

Im Hinblick auf die Plattformkompatibilität gibt es bei LabVIEW 8.x folgende Änderungen:

- LabVIEW-Versionen ab 7.1 laufen nicht unter Windows Me/98/95. LabVIEW 8.x läuft nicht unter Windows NT.
- LabVIEW 8.x läuft weder unter Mac OS X 10.2 noch unter Vorgängerversionen davon.
- LabVIEW 8.x läuft nicht unter Sun Solaris.

Systemvoraussetzungen

Für LabVIEW 7.x ist eine Bildschirmauflösung von 800×600 Pixeln erforderlich. Empfohlen werden jedoch 1.024×768 Pixel. Für LabVIEW 2009 ist eine Bildschirmauflösung von 1.024×768 Pixeln erforderlich.

(Windows) Für LabVIEW 7.x benötigen Sie mindestens einen Pentium III, einen Celeron-Prozessor mit 600 MHz oder einen gleichwertigen Prozessor. Es wird jedoch ein Pentium 4 oder ein gleichwertiger Prozessor empfohlen. Für LabVIEW 2009 wird mindestens ein Pentium III, ein Celeron mit 866 MHz oder ein vergleichbarer Prozessor benötigt (empfohlen wird ein Pentium 4/M oder ein vergleichbarer Prozessor).

LabVIEW 7.x benötigt für die Minimalversion einen Festplattenspeicher von mindestens 130 MB und für die vollständige Installation 550 MB. Für die vollständige Installation von LabVIEW 2009 ist ein Festplattenspeicher von mindestens 1,6 GB erforderlich.

(Mac OS) Für die Minimalversion von LabVIEW 7.x sind mindestens 280 MB Festplattenspeicher und für die vollständige Installation 350 MB erforderlich. Für die Minimalversion von LabVIEW 2009 sind mindestens 563 MB Festplattenspeicher erforderlich. Für die vollständige Installation benötigt die Version 1,2 GB.

(Linux) Für LabVIEW 7.x benötigen Sie mindestens einen Pentium III, einen Celeron-Prozessor mit 600 MHz oder einen gleichwertigen Prozessor. Es wird jedoch ein Pentium 4 oder ein gleichwertiger Prozessor empfohlen. Für LabVIEW 2009 wird mindestens ein Pentium III, ein Celeron mit 866 MHz oder ein vergleichbarer Prozessor benötigt (empfohlen wird ein Pentium 4/M oder ein vergleichbarer Prozessor).

LabVIEW 7.x benötigt für die Minimalversion einen Festplattenspeicher von mindestens 200 MB und für die vollständige Installation 300 MB. Für die Minimalversion von LabVIEW 2009 sind mindestens 630 MB Festplattenspeicher erforderlich und für die vollständige Installation 835 MB.

Für LabVIEW 7.x wird die GNU C Library (`glibc`) ab Version 2.1.3 benötigt. Empfohlen wird eine Version ab 2.2.4. Für LabVIEW 2009 ist eine GNU C Library ab 2.2.4 erforderlich.

LabVIEW 7.x läuft nur unter Red Hat Linux ab 7.0, Mandrake Linux ab 8.0, SuSE Linux ab 7.1 oder Debian Linux ab 3.0. LabVIEW 2009 kann auf Red Hat Enterprise Linux WS 4 und SuSE Linux 10.3 (oder jeweils einer neueren Version) ausgeführt werden.

Benutzerdefinierte Palettenansichten

In LabVIEW 8.x gibt es keine benutzerdefinierten Palettenansichten mehr. Paletten können jetzt ohne Anlegen einer benutzerdefinierten Palettenansicht bearbeitet werden. Weitere Informationen zu Änderungen der Paletten in LabVIEW 8.0 finden Sie auf der Website von National Instruments ni.com/info nach Eingabe des Infocodes `lv8palette`.

Änderungen an der Funktionsweise von VIs und Funktionen

Die Funktionsweise der nachfolgenden VIs und Funktionen hat sich in LabVIEW 7.1 oder 8.0 geändert.

.NET-VIs und -Applikationen

Für die .NET-Funktionen und -Applikationen in LabVIEW 8.x wird mindestens das .NET Framework 1.1 Service Pack 1 benötigt. Vor der Installation des .NET Framework 1.1 Service Pack 1 muss Microsoft .NET Framework 1.1 Hotfix KB886904 entfernt werden.

Wenn Sie ein in LabVIEW 7.x gespeichertes .NET-VI laden, werden Sie möglicherweise in LabVIEW 8.x zum Suchen der Assemblys aufgefordert. Das gilt auch dann, wenn sich die Assembly-Dateien im selben Verzeichnis wie das VI befinden oder im Dialogfeld

Werkzeuge»Fortgeschritten»NET-Assembly-Referenzen in LabVIEW 7.x registriert wurden.

Algorithmen der Analyse-VIs

In LabVIEW-Versionen ab 7.1 arbeiten Analyse-VIs mit dem BLAS/LAPACK-Algorithmus. Dadurch liefern diese VIs nun genauere Ergebnisse. In LabVIEW 8.x befinden sich diese VIs auf den Paletten

Mathematik und **Signalverarbeitung**.

Express-VI "Signale anfügen"

Wenn in LabVIEW 7.x beim Express-VI "Signale anfügen" der Eingang **Eingangssignal A** ein leeres Array ist oder offen gelassen wird und Sie den Eingang **Eingangssignal B** mit einem einzelnen oder kombinierten Signal verbinden, ist auch der Ausgang **Angefügte Signale** leer. Wenn es in LabVIEW 8.x kein **Eingangssignal A** gibt oder dieser Eingang offen gelassen wird und Sie an **Eingangssignal B** ein einzelnes Signal anlegen, gibt das Express-VI **Eingangssignal B** aus. Wenn **Eingangssignal B** ein kombiniertes Signal enthält, werden alle Signale, aus denen es sich zusammensetzt, aneinander angehängt.

Vergleichsfunktionen

In LabVIEW 7.x und Vorgängerversionen werden Variantdaten von den Vergleichsfunktionen zuerst in Bezug auf die Länge und anschließend bitweise verglichen. In LabVIEW 8.x beginnt der Vergleich der Variantdaten mit den Typencodes, in denen der tatsächliche Typ der einzelnen Daten verschlüsselt ist, und wird dann bei den anderen typspezifischen Attributen fortgeführt.

VI "Skalarprodukt"

In LabVIEW 7.0 wird das Skalarprodukt der Eingangsvektoren X und Y beim VI "Skalarprodukt" mit Hilfe folgender Gleichung berechnet:

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

In LabVIEW-Versionen ab 7.1 wird das Skalarprodukt komplexer Zahlen dagegen mit folgender Gleichung berechnet:

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

wobei y_i^* die komplex Konjugierte von y_i ist.

VI "Einfaches Textprotokoll" (Mac OS und Linux)

Das Anschlussfeld des VIs "Einfaches Textprotokoll" hat sich geändert. Wenn Sie in LabVIEW 8.x ein VI öffnen, in dem das VI "Einfaches Textprotokoll" verwendet wird und das zuvor in LabVIEW 7.x oder einer älteren Version gespeichert wurde, müssen Sie das SubVI mit der rechten Maustaste anklicken und aus dem Kontextmenü die Option **SubVI neu zuweisen** auswählen.

Funktion "In String formatieren"

Wenn Sie in LabVIEW 7.x die %o-, %b- und %x-Syntaxelemente für Formatbezeichner mit der Funktion "In String formatieren" verwenden, werden die Werte des Fließkommazahl-Eingangs auf 32-Bit-Integer gerundet, bevor die Eingänge in einen String konvertiert werden.

In LabVIEW 8.x rundet diese Funktion aufgrund der Syntaxelemente für Formatbezeichner Fließkomma-Eingänge auf 64-Bit-Integer, bevor die Eingangswerte in Strings konvertiert werden.

Funktion "Zahlen verbinden"

In LabVIEW 7.x und Vorgängerversionen konvertiert die Funktion "Zahlen verbinden" 32-Bit-Integer-Eingänge in 16-Bit-Integer, um einen 32-Bit-Integer zu erstellen. In LabVIEW 8.x wird mit der Funktion "Zahlen verbinden" aus 32-Bit-Integer-Eingängen ein 64-Bit-Integer erzeugt.



Hinweis Wenn Sie ein in LabVIEW 7.x erstelltes VI in LabVIEW 8.x öffnen, wandelt LabVIEW 32-Bit-Integer-Eingänge in 16-Bit-Integer um.

Mathematik-VIs und Matrizen

In LabVIEW 8.x arbeiten die **Mathematik**-VIs auch mit dem Datentyp "Matrix". Wenn Sie ein VI aus LabVIEW 7.x in LabVIEW 8.x laden und das VI ein Mathematik-VI enthält, das mit einer Funktion verbunden ist, die mit dem Datentyp "Matrix" arbeiten kann und stattdessen ein 2D-Array verwendet, wird "7.x" in roter Schrift an der Funktion angezeigt. Mit dem roten Symbol wird angezeigt, dass LabVIEW das 2D-Array durch den Datentyp "Matrix" ersetzt hat.

Funktionen zur Umwandlung von Zahlen in Strings

In LabVIEW 7.x werden Fließkommazahlen bei den Funktionen "Zahl nach Hexadezimal-String", "Zahl nach Oktal-String" und "Zahl in Dezimal-String" vor der Umwandlung in einen String auf einen 32-Bit-Integer gerundet.

In LabVIEW 8.x runden diese Funktionen Fließkommazahlen auf 64-Bit-Integer, bevor die Eingangswerte in Strings konvertiert werden. Wenn Sie jedoch ein in LabVIEW 7.x erstelltes VI in LabVIEW 8.x öffnen, werden die Fließkommazahlen aus Gründen der Kompatibilität und Funktion auf 32-Bit-Integer gerundet.

Funktion "VI-Referenz öffnen"

Wenn in LabVIEW 7.x der Eingang "VI-Pfad" der Funktion "VI-Referenz öffnen" ein Pfad ist und sich im Speicher ein VI mit dem gleichen Namen befindet, gibt LabVIEW eine Referenz auf das VI im Speicher aus, selbst wenn der Pfad zu dem VI im Speicher nicht dem angegebenen Pfad entspricht.

Ist der Eingang "VI-Pfad" von "VI-Referenz öffnen" in LabVIEW 8.x ein String, wird ein VI nur geöffnet, wenn "VI-Pfad" mit dem vollständigen Namen des VIs im Speicher des Zielsystems übereinstimmt. Wenn "VI-Pfad" ein Pfad ist, durchsucht LabVIEW den Speicher nach einem VI mit dem angegebenen Pfad. Wenn kein VI mit passendem Pfad gefunden wird, versucht LabVIEW, das VI vom angegebenen Pfad zu laden. In LabVIEW-Versionen ab 8.5 tritt ein Fehler auf, wenn LabVIEW die Datei nicht finden kann oder wenn der VI-Name der Datei mit dem qualifizierten Namen eines anderen VIs übereinstimmt, das sich im Arbeitsspeicher des betreffenden Zielsystems befindet.

VI "Schnellskalierung"

Wenn in LabVIEW 7.1 und Vorgängerversionen der Eingang **X** des VIs "Schnellskalierung 1D" oder "Schnellskalierung 2D" ein Array aus Nullen ist, gibt das VI unter **max|X|** den Wert 0 und unter **Y[i]=X[i]/Max|X|** oder **Yij=Xij/Max|X|** ein Array aus NaN aus. Wenn in LabVIEW 8.x der Eingang **X** des VIs "Schnellskalierung" ein Array aus Nullen ist, gibt das VI unter **max|X|** den Wert 0 und unter **Y[i]=X[i]/Max|X|** oder **Yij=Xij/Max|X|** ein Array aus Nullen aus.

VI "Schlüssel lesen"

In LabVIEW 7.x und Vorgängerversionen können mit Hilfe der String-Instanz des VIs "Schlüssel lesen" japanische Multibyte-Zeichen gelesen werden, die in Shift-JIS kodiert sind. Dazu muss der Eingang **Multibyte-Kodierung** mit 1 oder <Shift-JIS> verbunden werden. In LabVIEW 8.x liest die String-Instanz des VIs "Schlüssel lesen" kodierte Multibyte-Zeichen automatisch. Dazu muss nur die Ländereinstellung des Betriebssystems auf die entsprechende Kodierung gesetzt werden.

VI "Skalieren"

Wenn in LabVIEW 7.1 und Vorgängerversionen der Eingang **X** des VIs "1D skalieren" oder "2D skalieren" ein Array aus Nullen ist, gibt das VI unter **Skalierung** den Wert 0, unter **Offset** den Wert 0 und unter **Y=(X-Offset)/Skalierung** ein Array aus NaN aus. Wenn in LabVIEW 8.x und Vorgängerversionen der Eingang **X** des VIs "Skalieren" ein Array aus Nullen ist, gibt das VI unter **Skalierung** den Wert 1, unter **Offset** den Wert 0 und unter **Y=(X-Offset)/Skalierung** ein Array aus Nullen aus.

Semaphor-VIs

In LabVIEW 7.x wird nicht versucht, die VIs "Semaphor freigeben" und "Semaphor belegen" auszuführen, wenn der Eingang "Fehler (Eingang)" einen Fehler empfängt. In LabVIEW 8.x wird auch bei einem Fehler an "Fehler (Eingang)" versucht, die VIs auszuführen. Wenn Sie allerdings ein in LabVIEW 7.x erstelltes VI in LabVIEW 8.x öffnen, wird das VI an die Funktionsweise in LabVIEW 7.x angepasst.

SMTP-E-Mail-VIs

In LabVIEW 7.x und Vorgängerversionen können Sie einen Zeichensatz festlegen, indem Sie einen Wert mit dem Eingang "Zeichensatz" eines SMTP-E-Mail-VIs verbinden. In LabVIEW 8.x wird bei den SMTP-E-Mail-VIs davon ausgegangen, dass die Nachricht im Zeichensatz des Systems vorliegt. Mit diesen VIs wird die Nachricht vor dem Versenden im UTF-8-Format kodiert. Die Eingänge "Zeichensatz" und "Zeichenkonvertierung" gibt es bei den neuen SMTP-E-Mail-VIs nicht mehr.

VI "Komplexe Zahlen sortieren"

Wenn Sie in LabVIEW 7.x und Vorgängerversionen den Eingang "Methode" des VIs "Komplexe Zahlen sortieren" auf **Betrag** stellen, wird die Reihenfolge der Elemente mit gleichem Betrag nicht geändert. Wenn Sie "Methode" in LabVIEW 8.x auf **Betrag** setzen, werden die Elemente mit gleichem Betrag zuerst nach ihrem Real- und dann nach ihrem Imaginärteil sortiert.



Hinweis In LabVIEW 8.6 und neueren Versionen wurde **Betrag** in **Betrag, Reell, Imaginär** umbenannt.

VI "Einheitsvektor"

In LabVIEW 7.x und Vorgängerversionen wird die Norm des Eingangsvektors beim VI "Einheitsvektor" anhand der folgenden Gleichung berechnet:

$$||\vec{x}|| = \sqrt{x_0^2 + x_1^2 + \dots + x_{n-1}^2}$$

In LabVIEW 8.x wird die Norm des Eingangsvektors beim VI "Einheitsvektor" anhand der folgenden Gleichung berechnet:

$$\|X\| = \left(|x_0|^p + |x_1|^p + \dots + |x_{n-1}|^p \right)^{\frac{1}{p}}$$

wobei X der Vektor (Eingang), $\|X\|$ die Norm und p der Normtyp ist.

Benutzer-VIs

VIs, die Sie in den Verzeichnissen `labview\help`, `labview\project` und `labview\wizard` ablegen, werden jeweils in den Menüs **Hilfe**, **Werkzeuge** und **Datei** angezeigt. VIs, die Sie in LabVIEW 7.x und Vorgängerversionen in diesen Verzeichnissen ablegen, funktionieren möglicherweise in LabVIEW 8.x nicht wie erwartet, da diese VIs in LabVIEW ab Version 8.0 in einer privaten Kopie der Applikation geöffnet werden.

Eine Liste aller Benutzer-VIs im Arbeitsspeicher aller Applikationsinstanzen erhalten Sie mit Hilfe des VIs "VIMemory Get VIs in Memory" in der Bibliothek

`labview\vi.lib\Utility\allVIsInMemory.llb`. Eine Referenz auf die aktuelle Applikationsinstanz wird mit dem VI "Get User Application Reference" in der Bibliothek

`labview\vi.lib\Utility\allVIsInMemory.llb` erzeugt. Weitere Informationen zu Applikationsinstanzen finden Sie in der *LabVIEW-Hilfe*.

Veraltete VIs und Funktionen

Folgende VIs und Funktionen werden von LabVIEW 8.x nicht unterstützt:

- Die LabVIEW-Versionen ab 7.1 enthalten das VI "Polynomial Real Zero Counter" (deutscher Titel: "Zähler für Nullstellen eines reellen Polynoms") nicht mehr. Es wurde umbenannt in "Polynomial Real Zeros Counter" (der deutsche Titel wurde beibehalten).
- (**Mac OS**) Die PPC-VIs werden bei LabVIEW-Versionen ab 7.1 nicht mehr installiert. Stattdessen gibt es jetzt die TCP-VIs.
- Das VI "QR-Faktorisierung" wird von LabVIEW 8.x nicht mehr unterstützt. Verwenden Sie stattdessen das VI "QR-Zerlegung".
- Die VIs "Levenberg Marquardt" und "Nichtlineare Levenberg-Marquardt-Anpassung" wurden in LabVIEW 8.x durch das VI "Nichtlineare Kurvenanpassung" ersetzt.
- In LabVIEW 8.x ist die Funktion "VISA: Statusinformation" nicht mehr auf der Palette **Funktionen** zu finden. Verwenden Sie anstelle dieser Funktion das VI "Einfacher Fehlerbehandler" oder "Allgemeiner Fehlerbehandler".
- Die VIs "Chi-Quadrat-Verteilung", "F-Verteilung", "Normalverteilung" und "T-Verteilung" wurden in LabVIEW 8.x durch die Instanzen "Chi-Quadrat", "F", "Normal" und "Student-t" des VIs "Stetige Verteilung" ersetzt.
- Die VIs "Inverse Chi-Quadrat-Verteilung", "Inverse F-Verteilung", "Inverse Normalverteilung" und "Inverse T-Verteilung" wurden in LabVIEW 8.x durch die Instanzen "Chi-Quadrat", "F", "Normal" und "Student-t" des VIs "Inverse stetige Verteilung" ersetzt.
- Die VIs "1D-Linearentwicklung" und "2D-Linearentwicklung" sind in LabVIEW 8.x nicht mehr auf der Palette **Funktionen** enthalten. Verwenden Sie stattdessen das VI "Linearentwicklung".

- Die VIs "1D-Polynomentwicklung" und "2D-Polynomentwicklung" sind in LabVIEW 8.x nicht mehr auf der Palette **Funktionen** enthalten. Verwenden Sie stattdessen das VI "Polynomentwicklung".
- Die VIs "1D: Kartesisch nach polar" und "1D: Polar nach kartesisch" befinden sich in LabVIEW 8.x nicht mehr auf der Palette **Funktionen**. Verwenden Sie stattdessen die Funktionen "Re/Im nach polar" oder "Polar nach Re/Im".
- In LabVIEW 8.x befindet sich das VI "Frequenzanalyse" nicht mehr auf der Palette **Funktionen**. Verwenden Sie stattdessen zur Messung von **Klirrfaktor** und **Oberwellenamplituden** das VI "Klirrfaktoranalyse" oder zum Messen der Ausgänge **SINAD** oder **Klirrfaktor plus Rauschen** das VI "SINAD-Analyse".
- In LabVIEW 8.x befindet sich das VI "Netzwerkfunktionen (avg)" nicht mehr auf der Palette **Funktionen**. Verwenden Sie stattdessen die VIs "Übertragungsfunktion (Betrag-Phase)", "Übertragungsfunktion (Real-Im)", "Kreuzspektrum (Betrag-Phase)" oder "Kreuzspektrum (Real-Im)".
- In LabVIEW 8.x befindet sich das VI "Impulsparameter" nicht mehr auf der Palette **Funktionen**. Verwenden Sie stattdessen das VI "Übergangsmessungen" zum Messen der Ausgänge **Flankensteilheit**, **Dauer**, **Überschwingen** (das Äquivalent vom VI "Übergangsmessungen" dazu ist der Ausgang **Nach Übergang**) und **Vorschwingen** (das Äquivalent vom VI "Übergangsmessungen" dazu ist der Ausgang **Vor Übergang**), das VI "Impulsmessungen" zum Messen der Ausgänge **Periode**, **Impulsdauer** und **Tastverhältnis** und das VI "Amplitude und Pegel" zum Messen der Ausgänge **Amplitude**, **High-Pegel** und **Low-Pegel**.
- In LabVIEW 8.x ist das VI "Übertragungsfunktion" nicht mehr auf der Palette **Funktionen** zu finden. Verwenden Sie stattdessen die VIs "Übertragungsfunktion (Betrag-Phase)" oder "Übertragungsfunktion (Real-Im)".
- In LabVIEW 8.x befindet sich das Express-VI "DIAdem-Protokollassistent" nicht mehr auf der Palette **Funktionen**. Verwenden Sie stattdessen das Express-VI "DIAdem-Protokoll".
- In LabVIEW 8.x befinden sich die Konstanten "VISA-Ressourcenname" und "Logischer IVI-Name" nicht mehr auf der Palette **Funktionen**. Der VISA-Ressourcenname wird nun mit Hilfe des Eingangs "VISA-Ressourcenname" der VISA-VIs und der logische IVI-Name mit dem entsprechenden Eingang des Treiber-VIs festgelegt, mit dem das Gerät initialisiert wird.
- In LabVIEW 8.x ist die Fehler-Ringkonstante nicht mehr auf der Palette **Funktionen** zu finden. Verwenden Sie stattdessen zur Eingabe des gewünschten Fehlercodes eine Konstante für einen vorzeichenlosen 32-Bit-Integer.
- **(Windows und Linux)** In LabVIEW 8.x befinden sich die Audio-VIs, die es in LabVIEW 7.x auf der Palette **Audio** gab, nicht mehr auf der Palette **Funktionen**. Verwenden Sie stattdessen die Audio-VIs von LabVIEW 8.x. Die mit LabVIEW 7.x gelieferten Beispiele gibt es in LabVIEW 8.x nicht mehr.

Datei-I/O-VIs und -Funktionen

In LabVIEW 8.x befindet sich das VI "Zeichen aus Datei lesen" nicht mehr auf der Palette **Funktionen**. Verwenden Sie stattdessen die Funktion "Aus Textdatei lesen".

In LabVIEW 8.x befindet sich das VI "Öffnen/Erstellen/Ersetzen einer Datei" nicht mehr auf der Palette **Funktionen**. Verwenden Sie stattdessen die Funktion "Öffnen/Erstellen/Ersetzen einer Datei". Die folgenden Funktionen enthalten einen Teil des Funktionsumfangs, den das VI "Öffnen/Erstellen/Ersetzen einer Datei" in LabVIEW 7.x und Vorgängerversionen hatte:

- Die Größe einer Datei lässt sich mit der Funktion "Dateigröße ermitteln" feststellen.
- Der Startpfad, das Dateimuster und der Standardname einer Datei oder eines Verzeichnisses für ein Dateidialogfeld werden im Express-VI "Dateidialog" festgelegt.
- Eine Referenz lässt sich mit Hilfe der Funktion "Referenz nach Pfad" in einen Pfad umwandeln.

- Mit der Funktion "In Binärdatei schreiben" können Sie plattformunabhängige Textdateien oder andere Arten von Binärdateien erstellen und mit der Funktion "Aus Binärdatei lesen" können Sie die resultierende Binärdatei auslesen.

In LabVIEW 8.x befinden sich die Funktionen "Datei lesen" und "Datei schreiben" nicht mehr auf der Palette **Funktionen**. Verwenden Sie stattdessen die Funktionen "Aus Binärdatei lesen" und "In Binärdatei schreiben".

In LabVIEW 8.x befindet sich das VI "Zeichen in Datei schreiben" nicht mehr auf der Palette **Funktionen**. Verwenden Sie stattdessen die Funktion "In Textdatei schreiben".

In LabVIEW 8.x ist die Funktion "Zugriffsrechte" nicht mehr auf der Palette **Funktionen** zu finden. Verwenden Sie stattdessen die Funktionen "Berechtigungen ermitteln" und "Berechtigungen festlegen".

In LabVIEW 8.x befindet sich die Funktion "EOF" nicht mehr auf der Palette **Funktionen**. Verwenden Sie stattdessen die Funktionen "Dateigröße ermitteln" und "Dateigröße festlegen".

In LabVIEW 8.x ist die Funktion "Verzeichnis auflisten" nicht mehr auf der Palette **Funktionen** zu finden. Verwenden Sie stattdessen die Funktion "Ordner anzeigen".

In LabVIEW 8.x ist die Funktion "Bereich sperren" nicht mehr auf der Palette **Funktionen** zu finden. Verwenden Sie stattdessen die Funktion "Zugriff verweigern".

Wenn Sie ein in LabVIEW 7.x erstelltes VI öffnen und auf dem Blockdiagramm die Funktion "Neues Verzeichnis" enthalten ist, wird diese in LabVIEW 8.x durch die Funktion "Ordner erstellen" ersetzt. Wenn der von Ihnen im Eingang "Pfad" festgelegte Ordner nicht existiert, erstellt die Funktion "Ordner erstellen" das Verzeichnis. Die Funktion gibt keine Fehlermeldung aus, wie es bei der Funktion "Neues Verzeichnis" der Fall war.

In LabVIEW 8.x befindet sich die Funktion "Suchen" nicht mehr auf der Palette **Funktionen**. Verwenden Sie stattdessen die Funktionen "Dateiposition ermitteln" und "Dateiposition festlegen".

In LabVIEW 8.x ist die Funktion "Datei- und Programmtyp" nicht mehr auf der Palette **Funktionen** zu finden. Verwenden Sie stattdessen die Funktionen "Datei- und Programmtyp ermitteln" oder "Datei- und Programmtyp festlegen".

In LabVIEW 8.x ist die Funktion "Datenträger-Info" nicht mehr auf der Palette **Funktionen** zu finden. Verwenden Sie stattdessen die Funktion "Datenträger-Info lesen".

In LabVIEW 8.x befinden sich die Funktionen "Datei öffnen" und "Neue Datei" nicht mehr auf der Palette **Funktionen**. In LabVIEW 8.0 befindet sich das VI "Zeichen aus Datei lesen" ebenfalls nicht mehr auf der Palette **Funktionen**, wird jedoch aus Gründen der Kompatibilität mitgeliefert.

In LabVIEW 8.x befinden sich die VIs "Aus I16-Datei lesen", "Aus Sgl-Datei lesen", "In I16-Datei schreiben" und "In Sgl-Datei schreiben" nicht mehr auf der Palette **Funktionen**. Verwenden Sie stattdessen die VIs "Aus Binärdatei lesen" und "In Binärdatei schreiben".

Änderungen der Funktionsweise von Eigenschaften, Methoden und Ereignissen

Die Funktionsweise folgender Eigenschaften, Methoden und Ereignisse hat sich in LabVIEW 7.1 oder 8.0 geändert:

Eigenschaften und Methoden von Applikationen

In LabVIEW 8.x hängt die Funktionsweise einiger Applikationseigenschaften und -methoden von der Kopie der Anwendung (Applikationsinstanz) ab, zu der sie gehören. So richtet sich zum Beispiel die Funktionsweise der Eigenschaft "Applikation: Alle VIs im Speicher" nach der Applikationsinstanz, in der sich diese befindet. Diese Eigenschaft gibt eine Liste mit allen VIs aus, die sich im Speicher derselben Applikationsinstanz befinden wie die Eigenschaft. Dagegen ist die Funktionsweise der Eigenschaft

"Applikation:Verzeichnispfad" von der Applikationsinstanz unabhängig. Diese Eigenschaft gibt einen absoluten Pfad zu dem Verzeichnis aus, in dem sich die Applikation befindet. Diese Angabe ist für alle Applikationsinstanzen gleich.

Weitere Informationen zu Applikationsinstanzen finden Sie in der *LabVIEW-Hilfe*.

Methode "Frontpanel:Öffnen"

Die Methode "FP öffnen" aus LabVIEW 7.0 wurde in LabVIEW 7.1 in "FP öffnen - Alt" umbenannt. LabVIEW 7.1 enthält eine andere Methode "FP öffnen", die keine Fehlermeldung mehr ausgibt, wenn das Frontpanel bereits geöffnet ist. Die Methode "FP öffnen" aus LabVIEW 7.1 wurde in LabVIEW 8.x in "Frontpanel:Öffnen" umbenannt. Ersetzen Sie bei VIs, in denen die Methode "FP öffnen - Alt" enthalten ist, diese Methode durch "Frontpanel:Öffnen".

Methode "VI ausführen"

Wenn Sie in LabVIEW 7.1 den Eingang **Ref automatisch schließen** der Methode "VI ausführen" auf TRUE setzen, wird die Referenz nach Beendigung der VI-Ausführung automatisch geschlossen. Wenn die Methode "VI ausführen" einen Fehler ausgibt, wird die Referenz nicht automatisch geschlossen. In LabVIEW 8.0 wird die Referenz automatisch geschlossen, wenn die Methode einen Fehler ausgibt. Dadurch kann in einem VI, das auf die Referenz angewiesen ist, während der Ausführung ein Fehler auftreten. In LabVIEW-Versionen ab 8.2 wurde die alte Funktionsweise wiederhergestellt.

Ereignisse "Taste gedrückt" und "Tastenwiederholung"

Das Datenfeld "VTaste" der Ereignisse "Taste gedrückt", "Taste gedrückt?", "Tastenwiederholung" und "Tastenwiederholung?" für VIs und Elemente gibt nun zwei verschiedene Werte aus, je nachdem, ob die Taste <Return> im alphanumerischen Block oder die Taste <Enter> im Ziffernblock gedrückt wurde. Wenn in LabVIEW 7.x oder Vorgängerversionen durch Betätigen der <Enter>- oder <Return>-Taste eines dieser Ereignisse ausgelöst wird, gibt LabVIEW im Datenfeld "VTaste" immer <Enter> aus. Wenn in LabVIEW 8.x durch Betätigen der <Enter>- oder <Return>-Taste eines dieser Ereignisse ausgelöst wird, gibt LabVIEW unter "VTaste" entweder <Enter> oder <Return> aus.

(Mac OS) In LabVIEW 8.x können Kontextmenüs nur noch durch einen Klick bei gedrückter <Control>-Taste und nicht mehr bei gedrückter <Command>-Taste geöffnet werden. Wenn Sie dieses Verhalten mit einer Ereignisstruktur nachahmen, müssen Sie Ihre VIs der neuen Funktionsweise entsprechend anpassen.

Eigenschaften für Listenfeld

Wenn Sie in LabVIEW 7.x oder Vorgängerversionen die Eigenschaft "Erste Zeile" in einem Listenfeld auf eine Zeile einstellen, die sich unter dem untersten Element im Listenfeld befindet, so wird die Zeile am letzten sichtbaren Element fixiert. In LabVIEW 8.x hat die Anzahl sichtbarer Einträge keinen Einfluss auf die Anzahl der Zeilen, die mit dieser Eigenschaft verbunden werden können.

LabVIEW 8.x unterstützt die Eigenschaft "Doppelklick" für einspaltige Listenfelder nicht. Verwenden Sie stattdessen die Methode "Doppelt angeklickte Zeile ermitteln".

Eigenschaft "Übergeordnetes VI"

In LabVIEW 7.x und Vorgängerversionen gibt die Eigenschaft "Übergeordnetes VI" eine Referenz auf das VI aus, zu dem das Objekt gehört. Diese Referenz behält das VI im Speicher. In LabVIEW 8.x behält die von der Eigenschaft "Übergeordnetes VI" ausgegebene Referenz das VI nicht im Speicher. Beim Löschen des übergeordneten VIs aus dem Arbeitsspeicher verliert die Referenz ihre Gültigkeit. Mit der Funktion "VI-Referenz öffnen" kann eine Referenz erzeugt werden, mit der ein VI bis zum manuellen Schließen der Referenz im Arbeitsspeicher verbleibt.

Texteigenschaft

In LabVIEW 7.x und Vorgängerversionen gibt die Eigenschaft "Text" einen String in normaler Darstellung aus. In LabVIEW 8.x dagegen gibt die Eigenschaft einen String in derselben Darstellungsart aus wie das Frontpanel-Objekt. Wenn Sie das Frontpanel-Objekt beispielsweise auf Passwortanzeige gestellt haben, gibt die Eigenschaft den Text auch in der Passwortanzeige aus.

Eigenschaft "Modifikationen:Bitset VI-Modifikationen"

In LabVIEW-Versionen bis einschließlich 7.x ändert sich der Wert der Eigenschaft "Modifikationen:Bitset VI-Modifikationen" auf einen anderen Wert als Null, wenn Sie ein VI neu verknüpfen müssen. In LabVIEW-Versionen ab 8.x bleibt der Wert Null, wenn keine Änderungen vorgenommen werden, selbst wenn Sie ein VI neu verknüpfen müssen.

Eigenschaften für Baumstruktur

In LabVIEW 7.x und Vorgängerversionen geben "Eigenschaften aktiver Zellen:Zellengröße:Höhe" und "Eigenschaften aktiver Zellen:Zellengröße:Breite" 1 Pixel für jede Zeile im Baumstruktur-Element weniger aus als in LabVIEW 8.x. Wenn Sie z. B. ein in LabVIEW 7.x erstelltes VI in LabVIEW 8.x laden, das einen Eigenschaftsknoten enthält, der die Höhe eines Baumstruktur-Elements mit 70 Pixeln und die Breite mit 16 Pixeln ausgibt, geben alle neuen eingefügten Eigenschaftsknoten zum Bestimmen der Höhe und Breite 69 Pixel und 15 Pixel aus.

VI-String-Methoden

Strings, die Sie in älteren LabVIEW-Versionen mit Hilfe der Methode "VI-Strings exportieren" gespeichert haben, werden mit der Methode "VI-Strings:Importieren" möglicherweise nicht ordnungsgemäß in LabVIEW 8.x importiert.

Veraltete Eigenschaften, Methoden und Ereignisse

Die nachfolgenden Eigenschaften, Methoden und Ereignisse werden von LabVIEW 8.x nicht mehr unterstützt.

Eigenschaften für Cursor

Die Eigenschaft "Cursor-Fixiermodus" wird in LabVIEW 8.x nicht mehr unterstützt. Verwenden Sie stattdessen die Eigenschaft "Cursor-Modus".

Eigenschaften und Ereignisse für ein- und mehrspaltige Listenfelder, Tabellen, Binärtabellen, Baumstrukturen

In LabVIEW 8.x wird die Eigenschaft "Zellenvordergrundfarbe" für mehrspaltige Listenfelder nicht mehr unterstützt. Verwenden Sie stattdessen die Eigenschaft "Aktive Zelle:Schriftart der Zelle:Farbe".

In LabVIEW 8.x wird die Eigenschaft "Zellenvordergrundfarbe" für Text- und Binärtabellen nicht mehr unterstützt. Verwenden Sie stattdessen die Eigenschaft "Aktive Zelle:Schriftart für Zelle:Farbe" für Tabellen und Digitaltabellen.

In LabVIEW 8.x wird die Eigenschaft "Eigenschaften aktiver Zellen:Vordergrundfarbe" für Baumstrukturelemente nicht mehr unterstützt. Verwenden Sie stattdessen die Eigenschaft "Aktive Zelle:Schriftart der Zelle:Farbe".

In LabVIEW 8.x werden die Ereignisse "Ziehen", "Ziehen?", "Ablegen" und "Ablegen?" in der Klasse "Baumstruktur" nicht mehr unterstützt. Verwenden Sie stattdessen die Ereignisse "Ziehen beendet", "Ziehen - Eintritt", "Ziehen - Austritt", "Ziehen - Über", "Ziehen - Quelle aktualisiert", "Ziehen begonnen", "Ziehen begonnen?" und "Ablegen" in der Klasse "Element".

Eigenschaften für NumerischMitText

In LabVIEW 8.x werden die Eigenschaften "Farben angegebenes NumElement", "Farben angegebenes NumElement:Hintergrundfarbe" und "Farbe angegebenes numerisches Element:Textfarbe" für numerische Elemente nicht mehr unterstützt. Verwenden Sie stattdessen die Eigenschaften "Textfarben", "Textfarben:Hintergrundfarbe" und "Textfarben:Textfarbe".

Eigenschaften für Panel

In LabVIEW 8.x wird die Eigenschaft "Farbe" in der Klasse "Frontpanel" nicht mehr unterstützt. In LabVIEW 8.x gilt die Eigenschaft nur für den Fensterbereich links oben. Verwenden Sie stattdessen die Eigenschaft "Farbe des Fensterbereichs" in der Klasse "Fensterbereich".

Eigenschaften für Unterpanel

Verwenden Sie in LabVIEW 8.x das Feld eines SubVIs in einem Unterpanel, um die Anzeige der Bildlaufleisten für Unterpanel-Elemente zu konfigurieren und das Frontpanel in den Unterpanel-Elementen zu skalieren.

In LabVIEW 8.x wird die Unterpanel-Eigenschaft "X-Bildlaufleiste sichtbar" nicht mehr unterstützt. Verwenden Sie stattdessen die Eigenschaft "Sichtbarkeit der horizontalen Bildlaufleiste" für den Fensterbereich.

In LabVIEW 8.x wird die Unterpanel-Eigenschaft "Y-Bildlaufleiste sichtbar" nicht mehr unterstützt. Verwenden Sie stattdessen die Eigenschaft "Sichtbarkeit der vertikalen Bildlaufleiste" für den Fensterbereich.

In LabVIEW 8.x wird die Unterpanel-Eigenschaft "Frontpanel skalieren" nicht mehr unterstützt. Verwenden Sie stattdessen die Methode "Skalierungsmodus festlegen" für den Fensterbereich.

Eigenschaften, Methoden und Ereignisse für VIs

In LabVIEW 8.x wird die Eigenschaft "Frontpanel-Fenster:Autom. zentrieren" nicht mehr unterstützt. Verwenden Sie stattdessen die Methode "Frontpanel:Zentriert".

In LabVIEW 8.x wird die Eigenschaft "Frontpanel-Fenster:Bildschirmfüllend anzeigen" nicht mehr unterstützt. Verwenden Sie stattdessen die Eigenschaft "Frontpanel-Fenster:Status".

In LabVIEW 8.x wird die Eigenschaft "Frontpanel-Fenster:Ursprung" in der VI-Klasse nicht mehr unterstützt. In LabVIEW 8.x gilt die Eigenschaft nur für den Fensterbereich links oben. Verwenden Sie stattdessen die Eigenschaft "Ursprung" in der Klasse "Fensterbereich".

In LabVIEW 8.x wird die Eigenschaft "Frontpanel-Fenster:Bildlaufleisten anzeigen" in der VI-Klasse nicht unterstützt. In LabVIEW 8.x gilt die Eigenschaft nur für den Fensterbereich links oben. Verwenden Sie stattdessen die Eigenschaften "Sichtbarkeit der horizontalen Bildlaufleiste" und "Sichtbarkeit der vertikalen Bildlaufleiste" in der Klasse "Fensterbereich".

In LabVIEW 8.x werden die Methoden "Frontpanel-Skalierungsmodus ermitteln" und "Frontpanel-Skalierungsmodus einstellen" in der VI-Klasse nicht mehr unterstützt. In LabVIEW 8.x gelten diese Methoden nur für den Fensterbereich links oben. Verwenden Sie stattdessen die Methoden "Skalierungsmodus ermitteln" und "Skalierungsmodus festlegen" in der Klasse "Fensterbereich".

In LabVIEW 8.x können die Ereignisse "Maustaste gedrückt", "Maustaste gedrückt?", "Mausbewegung" und "Maustaste losgelassen" der VI-Klasse nicht im Dialogfeld **Ereignisse bearbeiten** ausgewählt werden. Verwenden Sie stattdessen jeweils die Ereignisse "Maustaste gedrückt", "Maustaste gedrückt?", "Mausbewegung" oder "Maustaste losgelassen" in der Klasse "Fensterbereich".

Neuerungen am Application Builder

Der Application Builder ist im **Projekt-Explorer** integriert. Wenn Sie mit dem LabVIEW Base Package oder Full Development System arbeiten, müssen Sie den Application Builder separat erwerben. Besuchen

Sie dazu die Website von National Instruments unter ni.com/info und geben Sie den Infocode `rd1v21` ein.

Mit den **Build-Spezifikationen** im **Projekt-Explorer** können Sie Build-Spezifikationen für Applikationen (EXE-Dateien), DLLs und Zip-Dateien anlegen und entsprechende Builds erstellen. **(Windows)** Mit den **Build-Spezifikationen** können Sie auch Installer-Spezifikationen anlegen und daraus Installer erzeugen. Build-Spezifikationen entsprechen den `*.bld`-Dateien in älteren Versionen des Application Builders. Sie gehören nun jedoch zu einem LabVIEW-Projekt und sind keine eigenständigen Dateien mehr.



Hinweis Die Application-Builder-Tools müssen in einem Projekt verwendet werden.

Die Umwandlung einer `*.bld`-Datei in eine Build-Spezifikation ist in einem neuen Projekt möglich. Wählen Sie **Werkzeuge»Build-Skript konvertieren**, um zu der `*.bld`-Datei zu gelangen, die konvertiert werden soll. Die Datei wird zum Erstellen eines Projekts verwendet, die die Quelldateien und die Build-Spezifikationen enthält.

Tags der Applikationsmenüpunkte

Nachfolgend werden alle Tags für Standardmenüpunkte aufgeführt, die nicht mehr in LabVIEW enthalten sind – entweder weil der entsprechende Menüpunkt abgeschafft wurde oder mit einer anderen Funktion verschmolzen wurde:

- `APP_SAVE_WITH_OPTIONS`
- `APP_UPDATE_VXI`
- `APP_DSC_TOOLBAR`
- `APP_DSC_TAGEDITOR`
- `APP_DSC_TAGMONITOR`
- `APP_DSC_HTV`
- `APP_DSC_ENGINE`
- `APP_DSC_SECURITY`
- `APP_DSC_LOGOUT`
- `APP_DSC_CPWD`
- `APP_DSC_USERINFO`
- `APP_DSC_USEREDITOR`
- `APP_DSC_ADVANCED`
- `APP_DSC_STARTUP`
- `APP_DSC_SRVBRW`
- `APP_DSC_IST`
- `APP_DSC_IMAGENAV`
- `APP_DSC_OPTIONS`
- `APP_SRC_CODE_CTRL`
- `APP_BUILD_STANDALONE_APP`
- `APP_EDIT_VI_LIBRARY`
- `APP_DN_ASSEMBLY_REFS`
- `APP_SHOW_CLIPBOARD`
- `APP_VIEW_PRINTED_MANUALS`
- `APP_RT_ENGINE_INFO`
- `APP_SWITCH_EXEC_TARGET`

- APP_REALTIME

Bei Laufzeitmenü-Dateien (*.rtm-Dateien), die in einer Vorgängerversion von LabVIEW gespeichert wurden, wird das Tag in LabVIEW 8.x automatisch aus der *.rtm-Datei entfernt, wenn die Datei im **Menü-Editor** gespeichert wird. Die gelöschten Tags für Applikationsmenüpunkte sind für LabVIEW reserviert und können nicht als Benutzer-Tags verwendet werden.

Unterstützung von HiQ

HiQ wird in LabVIEW 8.x nicht mehr unterstützt. Wenn in einer Applikation HiQ-VIs verwendet werden, sollten diese eventuell durch Signalverarbeitungs- oder Mathematik-VIs ersetzt werden.

Fehlerliste

In LabVIEW 7.x und Vorgängerversionen werden im Abschnitt **VI-Liste** des Fensters **Fehlerliste** alle Fehler in den VIs im Speicher angezeigt. In LabVIEW 8.x werden in der **Fehlerliste** unter **Fehlerhafte Elemente** die Fehler an allen Objekten im Speicher, wie VIs oder Bibliotheken, angezeigt. Wenn mehrere Elemente den gleichen Namen haben, wird hier die Applikationsinstanz für jedes Element angezeigt. Weitere Informationen zu Applikationsinstanzen finden Sie in der *LabVIEW-Hilfe*.

Syntax für VI-String-Dateien

Wenn mit der Option **Werkzeuge»Fortgeschritten»Strings importieren** oder der Methode "VI-Strings:Importieren" VI-String-Dateien importiert werden, sucht LabVIEW 8.x nach den Tags `<GROUPER></GROUPER>`. Diese Tags stehen für Frontpanel-Objekte, die zu einer Gruppe zusammengefasst sind. Daher können in LabVIEW 8.x keine VI-String-Dateien importiert werden, die in älteren Versionen von LabVIEW erstellt wurden.

In LabVIEW 7.1 und Vorgängerversionen werden Listenfeld-Strings im Abschnitt `<ITEMS>` der Privatdaten aufgelistet. In LabVIEW 8.x werden Strings in einem Listenfeld im Abschnitt `<STRINGS>` der Privatdaten aufgelistet. In LabVIEW 7.1 und Vorgängerversionen kann für ein Listenfeld nur eine Schriftart verwendet werden. Diese wird im Abschnitt `<LBLABEL>` der Privatdaten aufgeführt. In LabVIEW 8.x dagegen können in einem Listenfeld mehrere Schriftarten vorkommen. Diese werden im Abschnitt `<CELL_FONTS>` der Privatdaten aufgelistet.

In LabVIEW 7.1 und Vorgängerversionen werden Strings in einem mehrspaltigen Listenfeld in den Standarddaten aufgelistet. Der Standardwert für ein mehrspaltiges Listenfeld ist jedoch ein Integer oder ein Array aus Integeren. In LabVIEW 8.x werden Strings in einem mehrspaltigen Listenfeld in den Privatdaten aufgelistet.

In LabVIEW 7.1 und Vorgängerversionen werden weder Text noch Schriftarten aus Baumstrukturen exportiert. In LabVIEW 8.x können sowohl Baumstruktur-Strings als auch -Schriftarten exportiert werden. Der Export erfolgt im gleichen Format wie bei ein- oder mehrspaltigen Listenfeldern.

In LabVIEW 8.x enthält jede Zeile einer Exportdatei nicht mehr als zwei Tags für Privat- oder Standarddaten. Außerdem wird jede Unterebene in LabVIEW 8.x um eine Stelle eingerückt.

Führen Sie folgende Schritte aus, um VI-String-Dateien in das Format von LabVIEW 8.x zu konvertieren:

1. Importieren Sie die VI-String-Datei in die vorherige LabVIEW-Version.
2. Speichern Sie das VI.
3. Laden Sie das VI in LabVIEW 8.x.
4. Wählen Sie **Werkzeuge»Fortgeschritten»Strings exportieren**, um die VI-String-Datei im Format von LabVIEW 8.x zu speichern.

Konvertieren von Typdeskriptordaten in das oder aus dem Format von LabVIEW 7.x

Das Speicherformat von Typdeskriptoren hat sich in LabVIEW 8.x geändert. In LabVIEW 7.x werden Typdeskriptoren in Form von 16-Bit-Strings gespeichert. In LabVIEW 8.x werden Typdeskriptoren in

Form von 32-Bit-Strings gespeichert. Dadurch wird die Beschränkung der Größe auf 64 KB für Typdeskriptoren aufgehoben.

In LabVIEW 8.x ist es allerdings möglich, die in LabVIEW 7.x gespeicherten Typdeskriptoren zu lesen und so zu speichern, dass sie mit LabVIEW 7.x kompatibel sind. Die Funktion "Daten serialisieren" hat dazu nun den Kontextmenüpunkt **7.x-Daten konvertieren**. Wenn Sie die Funktion mit der rechten Maustaste anklicken und diesen Menüpunkt auswählen, wird mit den Eingangsdaten so verfahren, als ob in LabVIEW 7.x gearbeitet würde. Wenn Sie die Option **7.x-Daten konvertieren** aus dem Kontextmenü auswählen und am Ausgang "Daten-String" ein String anliegt, wird die Funktion in LabVIEW 8.x mit einem roten 7.x gekennzeichnet, um anzuzeigen, dass die Daten nicht im üblichen LabVIEW-8.x-Format vorliegen. Sollen die Daten nicht (mehr) umgewandelt werden, wird die Menüoption **7.x-Daten konvertieren** beim erneuten Anklicken deaktiviert.

Wenn in LabVIEW 8.x ein VI geladen wird, das zuletzt in LabVIEW 7.x oder einer älteren Version gespeichert wurde, wird in LabVIEW 8.x bei der Funktion "Daten serialisieren" die Einstellung **7.x-Daten konvertieren** automatisch aktiviert. Die Funktion arbeitet weiterhin wie in LabVIEW 7.x und Vorgängerversionen. Wenn für ein VI das Typdeskriptorformat von LabVIEW 8.x verwendet werden soll, klicken Sie mit der rechten Maustaste auf die Funktion "Daten serialisieren" und deaktivieren Sie die Option **7.x-Daten konvertieren**. Wenn VIs weder mit Dateien arbeiten, die Daten im LabVIEW-7.x- oder einem älteren Format enthalten, noch Daten mit VIs austauschen, die in LabVIEW 7.x oder einer Vorgängerversion ausgeführt werden, empfiehlt sich in der Regel das LabVIEW-8.x-Typdeskriptorformat. Das alte Format wird unter Umständen in künftigen Versionen nicht mehr unterstützt.

Umstellung vom integrierten LabVIEW-Versionsverwaltungsprovider

Den Versionsverwaltungsprovider von LabVIEW 7.x und Vorgängerversionen gibt es in LabVIEW 8.x nicht mehr. Wenn Sie in LabVIEW die Versionsverwaltung nutzen möchten, müssen Sie einen Versionsverwaltungsprovider eines Drittanbieters auswählen. Wenn Sie bisher den LabVIEW-Provider verwendet haben, müssen Sie die Dateien auf einen anderen Versionsverwaltungsprovider umstellen. Die aktuelle Liste der Versionsverwaltungsprovider von Drittanbietern, mit denen LabVIEW arbeitet, finden Sie auf der Website von National Instruments unter ni.com/info nach Eingabe des Infocodes `exgucn`.

Beim Umstellen von Dateien auf einen neuen Versionsverwaltungsprovider geht die Versionshistorie verloren, die im LabVIEW-Provider gespeichert war. Ältere Versionen der Dateien lassen sich nicht auf den neuen Provider übertragen.

Führen Sie folgende Schritte aus, um Dateien vom integrierten Versionsverwaltungsprovider auf einen Versionsverwaltungsprovider von einem Drittanbieter zu übertragen:

1. Öffnen Sie die Vorgängerversion von LabVIEW und vergewissern Sie sich, dass die Dateien im Versionsverwaltungsprovider von allen Nutzern eingchecked sind.
2. Rufen Sie auf dem Computer, auf dem Sie den Versionsverwaltungsprovider umstellen möchten, im LabVIEW-Provider die neueste Version aller Dateien ab.
3. Checken Sie im LabVIEW-Provider die betreffenden Dateien aus der Versionsverwaltung aus.
4. Nehmen Sie im Versionsverwaltungsprovider des Drittanbieters die im neuen Versionsverwaltungsprojekt gewünschten Einstellungen vor.
5. Konfigurieren Sie LabVIEW für die Zusammenarbeit mit dem neuen Versionsverwaltungsprovider.

Informationen zur Konfiguration von LabVIEW für die Verwendung eines Versionsverwaltungsproviders von Drittanbietern finden Sie im Buch **Grundlagen»Organisieren und Verwalten von Projekten»Anleitung»Verwendung der Versionsverwaltung in LabVIEW** im Register **Inhalt** der *LabVIEW-Hilfe*.

6. Erstellen Sie ein LabVIEW-Projekt. Fügen Sie dem Projekt die Dateien hinzu, die sich im LabVIEW-Versionsverwaltungsprovider befinden. Fügen Sie die Dateien der Versionsverwaltung hinzu, wenn Sie dazu aufgefordert werden. Sie können die Dateien auch direkt in den Provider des Drittanbieters einfügen.

Informationen zum Erstellen eines LabVIEW-Projekts finden Sie im Buch
Grundlagen»Organisieren und Verwalten von Projekten»Anleitung»Erstellen eines LabVIEW-Projekts im Register **Inhalt** der *LabVIEW-Hilfe*

Umwandeln von NaN-Strings in Integer (Windows)

Wenn NaN in LabVIEW 7.x in einen Integer umgewandelt wird (entweder programmatisch oder durch den Benutzer), stellt dieser Wert den kleinsten Wert für diesen Integer-Datentyp dar. So ergibt die Umwandlung von NaN in einen 16-Bit-Integer mit Vorzeichen beispielsweise -32.768. Das ist der kleinstmögliche Wert für einen 16-Bit-Integer mit Vorzeichen.

Wenn NaN in LabVIEW 8.x in einen Integer umgewandelt wird (entweder programmatisch oder durch den Benutzer), stellt dieser Wert den größten Wert für diesen Integer-Datentyp dar. So ergibt die Umwandlung von NaN in einen 16-Bit-Integer mit Vorzeichen beispielsweise 32.767. Das ist der größtmögliche Wert für einen 16-Bit-Integer mit Vorzeichen.

Mit Case-Strukturen verbundene Konstanten

In LabVIEW 7.x und älteren Versionen können SubVIs im Speicher gehalten werden, indem eine Konstante mit einer Case-Struktur verbunden wird und die SubVIs in einem niemals aufgerufenen Case untergebracht werden. Wenn Sie zum Beispiel eine TRUE-Konstante mit einer Case-Struktur verbinden und ein SubVI im FALSE-Case unterbringen, lädt LabVIEW das SubVI zusammen mit dem aufrufenden VI. In LabVIEW 8.x werden alle nicht benötigten Blockdiagrammabschnitte entfernt. Wenn Sie also ein in einer älteren Version gespeichertes VI mit einer Konstante, die mit einer Case-Struktur verbunden ist, in LabVIEW 8.x laden, wird die Konstante in ein verborgenes Element umgewandelt, um die Funktionsweise der älteren Version beizubehalten.

Verzögern von Meldungen des Betriebssystems

In LabVIEW 7.x werden Meldungen des Betriebssystems bearbeitet, während Callback-VIs zum Bearbeiten von .NET- und ActiveX-Ereignissen ausgeführt werden. In LabVIEW 8.x wird die Bearbeitung von Meldungen des Betriebssystems so lange verzögert, bis die Ausführung des Callback-VIs beendet ist oder bis Sie ein modales Dialogfeld geladen haben. Durch diese Verzögerung werden Callback-VIs ohne Unterbrechung ausgeführt. Es wird auch kein Ereignis in einem anderen Ereignis ausgelöst, was zu einem Sperrstatus führen kann.

Sie können von einem Callback-VI keine synchronen Aufrufe zu nicht modalen Dialogfenstern starten. Die Aufrufe eines nicht modalen Dialogfensters von einem Callback-VI müssen asynchron sein. Das wird erreicht, indem Sie die Methode "VI ausführen" im Dialogfenster aufrufen und eine boolesche Konstante mit dem Wert FALSE mit dem Eingang "Warten bis beendet" verbinden.

In LabVIEW 7.x werden Meldungen des Betriebssystems bearbeitet, während DLL- bzw. Shared-Library-Funktionen ausgeführt werden. In LabVIEW 8.x wird die Abarbeitung von Meldungen des Betriebssystems so lange aufgeschoben, bis die Aufrufe der DLL-Funktionen beendet sind oder ein modales Dialogfeld von der DLL aus geladen wurde. Durch diese Verzögerung werden DLL-Funktionen ohne Unterbrechung ausgeführt und LabVIEW kann während der Ausführung einer DLL-Funktion nicht auf die DLL zugreifen, mit der die Funktion arbeitet (was zu einer Blockade führen kann). Das Verzögern von Meldungen des Betriebssystems (z. B. Tastatureingaben) ist nützlich, um zu vermeiden, dass DLL-Funktionen in ihrer Ausführung gestört werden. Wenn eine DLL-Funktion also beispielsweise infolge eines Tastendrucks ausgeführt wird, sollte erst nach Ausführungsende der Funktion wieder auf ein Betätigen der Taste reagiert werden.

Bei dieser Standardeinstellung können nicht modale Dialogfelder nicht synchron aufgerufen werden, während eine DLL ausgeführt wird. Die Aufrufe eines nicht modalen Dialogfensters von einer DLL aus müssen asynchron sein. Das wird erreicht, indem Sie die Methode "VI ausführen" im Dialogfenster aufrufen und eine boolesche Konstante mit dem Wert FALSE mit dem Eingang "Warten bis beendet" verbinden.

In DLLs, die Sie selbst erstellen, können Sie bestimmen, ob Meldungen des Betriebssystems verzögert werden sollen oder nicht. Klicken Sie mit der rechten Maustaste auf die DLL im **Projekt-Explorer**, wählen Sie **Eigenschaften** aus dem Kontextmenü und dann **Fortgeschritten** aus der Liste **Kategorie**. Deaktivieren Sie die Option **Betriebssystemmeldungen in DLL verzögern**, um während der Ausführung von DLL-Funktionen Meldungen des Betriebssystems bearbeiten zu können.

Ressourcenmanager (Mac OS)

LabVIEW-Versionen bis einschließlich 7.x enthalten undokumentierte Eigenschaften zum Datenaustausch mit Macintosh-Ressourcendateien. Diese Methoden gibt es in LabVIEW 8.x nicht mehr. Hilfsprogramme, in denen diese undokumentierten Eigenschaften genutzt werden, funktionieren nicht mehr. Daher kann nicht mehr mit Macintosh-Ressourcendateien gearbeitet werden.

Dialogfelder mit einer oder zwei Schaltflächen

In LabVIEW 7.x und Vorgängerversionen können VIs zur Anzeige von Dialogfeldern mit einer oder zwei Schaltflächen nicht programmatisch abgebrochen werden. In LabVIEW 8.x dagegen können Sie ein solches VI programmatisch abbrechen, indem Sie die Methode "VI abbrechen" aufrufen.

Eigenschafts- und Methodenknoten

Wenn Sie in LabVIEW 7.x einen nicht sichtbar mit einem Objekt verbundenen Eigenschafts- oder Methodenknoten von einer Cursor-Legende aus erstellen, wird der Knoten gelöscht, wenn das VI in LabVIEW 8.x geöffnet wird.

Update von DLLs

Wenn Sie in LabVIEW 7.x oder einer Vorgängerversion eine DLL erstellen, die mit `labview.lib` verknüpft ist, verbinden Sie die DLL in LabVIEW 8.x stattdessen mit `labviewv.lib`. Weitere Informationen zum Verknüpfen von DLLs mit `labviewv.lib` finden Sie in der *LabVIEW-Hilfe*.

Seitenränder beim Drucken

In LabVIEW 7.x und Vorgängerversionen werden die Seitenränder in der Option **Seitenränder** der Seite **Drucken** im Dialogfeld **Optionen** in Zentimetern angegeben. In LabVIEW 8.x werden die **Seitenränder** in Millimetern angegeben.

Upgrade von LabVIEW 6.x

Bei einem Upgrade von LabVIEW 6.x auf LabVIEW 2009 können die nachfolgend aufgeführten Kompatibilitätsprobleme auftreten. Weitere Probleme, die beim Upgrade von LabVIEW auftreten können, sind in den Abschnitten [Upgrade von LabVIEW 7.x](#), [Upgrade von LabVIEW 8.0](#), [Upgrade von LabVIEW 8.2](#), [Upgrade von LabVIEW 8.5](#) und [Upgrade von LabVIEW 8.6](#) beschrieben.



Hinweis Alle Unterschiede zwischen den einzelnen Versionen ab 6.x sind in den *Hinweisen zum Upgrade von LabVIEW* jeder Version beschrieben, die Sie unter ni.com/manuals finden.

Änderungen am Datentyp "Signalverlauf"

Der Datentyp "Signalverlauf" verwendet in LabVIEW 7.0 anstelle einer Fließkommazahl mit doppelter Genauigkeit für die Komponente **t0** den Datentyp "Zeitstempel". Wenn Sie Daten in diesem Datenformat in LabVIEW 6.x speichern, ohne die Änderung in LabVIEW-Versionen ab 7.x zu berücksichtigen, tritt beim Abrufen der Daten unter Umständen ein Fehler auf.

In LabVIEW 7.x und neueren Versionen wandelt das VI "Signalverlauf aus Datei lesen" das alte Signalverlaufsdatentyp-Format in einer Datei in ein neues um. Ein Dialogfeld öffnet sich, in dem Sie aufgefordert werden, die Umwandlung zu akzeptieren. In der LabVIEW-Runtime-Engine kann das VI "Signalverlauf aus Datei lesen" diese Umwandlung jedoch nicht vornehmen. Daher wird eine Fehlermeldung ausgegeben. Wenn Sie auf der Website ni.com/info den Infocode `exd9zq` eingeben, erfahren Sie, was es im Hinblick auf Signalverläufe bei der Umstellung von LabVIEW 6.x auf LabVIEW 7.x oder neuere Versionen zu beachten gibt.

VIs für serielle Kompatibilität

In LabVIEW 7.x und neueren Versionen wird das VI "Serielle Kompatibilität" nicht mehr auf der Palette **Funktionen** angezeigt. VIs zur Kommunikation mit Geräten mit serieller Schnittstelle sollten mit VISA-VIs und -Funktionen erstellt werden.

In LabVIEW-Versionen ab 7.x wird zur Kommunikation mit dem seriellen Treiber des Betriebssystems nicht mehr mit dem `serpdrv`-Treiber gearbeitet. LabVIEW enthält kompatible VIs, die auf VISA basieren. Verwenden Sie in neuen Applikationen 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, funktionieren auch in LabVIEW 7.1 und neueren Versionen.

Wenn in der ursprünglichen LabVIEW-Anwendung Änderungen an der Zuweisung der Portnummern vorgenommen wurden, ist die Änderung anzugeben. Verwenden Sie dazu das VI "Set Serial Alias Ports" im Verzeichnis `labview\vi.lib\Instr\_sersup.llb`. Verbinden Sie ein String-Array mit dem Eingang **VISA Aliases** des VIs und geben Sie die Portbezeichnungen ein, die im Eingangs-Array 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 das VI "VISA: Seriellen Port konfigurieren" aufgerufen werden.

Anwendungsbeispiele für die VISA-VIs und -Funktionen zur Steuerung serieller Geräte finden Sie unter `labview\examples\instr\smplsrl.llb`.

Standardwerte in Schleifen

In LabVIEW 6.0 und Vorgängerversionen werden bei Nicht-Ausführung von For-Schleifen undefinierte Werte ausgegeben. In LabVIEW-Versionen ab 6.1 geben For-Schleifen voreingestellte Werte aus, wenn für die Anzahl der Durchläufe 0 angegeben wird oder ein Eingang mit aktivierter Auto-Indizierung mit einem leeren Array verbunden wird. In diesem Fall wird die Schleife nicht ausgeführt und für alle nicht automatisch indizierten Ausgabetunnel werden die Standardwerte für den Tunneldatentyp ausgegeben.

Lizenzen für Netzwerk-Frontpanel

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

Diese Lizenzen sind auf mehrere Benutzer erweiterbar.

Vergabe mehrerer Ausführungs-Threads

In LabVIEW-Versionen ab 7.1 werden zur Ausführung von VIs mehr Threads vergeben als in den Vorgängerversionen. Daher kann es bei der Verarbeitung mehrerer Threads zu Fehlern kommen, wenn ein Funktionsknoten zum Aufruf externer Bibliotheken fälschlicherweise als ablaufinvariant gekennzeichnet wurde, obwohl die DLL nicht ablaufinvariant ist. In der *LabVIEW-Hilfe* finden Sie weitere Informationen zum Knoten für den Aufruf externer Bibliotheken und zur Ablaufinvarianz.

Mit dem VI "Threadconfig" in der Bibliothek `labview\vi.lib\Utility\sysinfo.llb` können Sie die Zuweisung von Threads ändern. Sie können die Ablaufinvarianz von VIs deaktivieren, indem Sie

Datei»VI-Einstellungen wählen, dann im Pulldown-Menü **Kategorie** auf **Ausführung** klicken und die Option **Ablaufinvariante Ausführung** deaktivieren.

Weitere Informationen zum Zuweisen von Threads finden Sie in der *LabVIEW-Hilfe*.

Instrumententreiber

Bei LabVIEW-Versionen ab 7.x ist die LabVIEW-Instrumententreiber-CD nicht im Lieferumfang enthalten. Im Netzwerk für Gerätetreiber von National Instruments unter ni.com/idnet stehen die Treiber jedoch weiterhin zum Herunterladen zur Verfügung. Auf der Treiber-CD von National Instruments sind unter anderem NI-DAQ und NI-VISA enthalten.

Einheiten und Umrechnungsfaktoren

In LabVIEW-Versionen ab 7.x muss die zusätzliche Einheit nach Verwendung der Funktion "Mehrfacharithmetik" nicht mit der Funktion "Einheit umrechnen" entfernt werden.

Die Umrechnungsfaktoren für Einheiten wurden in LabVIEW 7.1 und neueren LabVIEW-Versionen 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 **Kalorie** in **Kalorie (thermisch)** und **Pferdestärke** in **Pferdestärke (elektrisch)** umbenannt. Die Abkürzungen für diese Einheiten haben sich nicht geändert. Nachfolgend finden Sie alle Änderungen an den Umrechnungsfaktoren für Einheiten zwischen LabVIEW 6.1 und 7.x:

Einheit	Definition (6.1)	Definition (ab 7. x)
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 verglichen mit Pferdestärke (elektrisch)	745,7 W	746 W. Die neue Umrechnung ist exakt.
Britische Gallone	4,54596 l	4,54609 l
Lichtjahr	9,4605 Pm	9,46073 Pm
Pound Force	4,448 N	4,448222 N
Rod	16,5 ft	5,029210 m
Slug	32,174 lb	14,59390 kg
Unitäre atomare Masseneinheit (u)	1,66057e–27 kg	1,66053873e–27 kg

Eigenschaft "Panel-Aktualisierungen verzögern"

Bei LabVIEW 6.1 und Vorgängerversionen werden ausstehende Wertänderungen an Frontpanel-Elementen erst angezeigt, wenn die Eigenschaft "Panel-Aktualisierungen verzögern" auf FALSE gesetzt wird. Wenn diese Eigenschaft in LabVIEW ab Version 7.0 den Wert TRUE hat, werden alle Frontpanel-Objekte mit ausstehenden Wertänderungen neu dargestellt und die nächste Aktualisierung der Objekte wird um eine bestimmte Zeit verschoben. In einigen Fällen nimmt die Neudarstellung der Elemente zusätzliche Zeit in Anspruch.

Wertebereiche numerischer Elemente

In LabVIEW 6.1 und Vorgängerversionen haben einige numerische Elemente per Voreinstellung den Mindestwert 0,00, den Höchstwert 0,00, den Inkrementwert 0,00 und die Einstellung **Ignorieren** bei

einer Bereichsüberschreitung. Für diese Elemente werden in LabVIEW ab Version 7.x die Werte des Standardwertebereichs für den jeweiligen Datentyp verwendet.

Typumwandlungspunkte und -definitionen

In LabVIEW-Versionen ab 6.1 sind in den Verbindungen Angaben über Typdefinitionen enthalten, so dass im Blockdiagramm mehr Typumwandlungspunkte vorliegen können. Wenn Sie ein Element mit Typdefinition mit einem VI oder Knoten ohne Typdefinition verbinden, wird ein Typumwandlungspunkt angezeigt. Typumwandlungspunkte werden auch angezeigt, wenn ein Ausgangsanschluss, der eine Typdefinition hat, mit einer Anzeige verbunden wird, die keine Typdefinition hat. Diese Typumwandlungspunkte zeigen, an welchen Stellen im VI Typdefinitionen nicht einheitlich verwendet werden. Die Ausführungsgeschwindigkeit des VIs wird davon nicht beeinträchtigt.

Um Typdefinitionen in Strings zu konvertieren, verwenden Sie die Funktion "Daten serialisieren", zu der Sie in der *LabVIEW-Hilfe* nähere Einzelheiten finden.

Schaltflächenbeschriftungen im Dialogfeld "Datei"

In LabVIEW 6.1 und Vorgängerversionen wird im Fenster der Dateidialog-Funktion die Schaltfläche **Speichern** angezeigt, wenn der Benutzer einen neuen Dateinamen eingeben kann. Andernfalls lautet die Beschriftung der Schaltfläche **Öffnen**. In LabVIEW 8.x lautet die Schaltflächenbeschriftung im Dateidialogfeld, das vom Express-VI "Dateidialog" angezeigt wird, in allen Fällen **OK**. Die Beschriftung lässt sich nachträglich mit Hilfe des Eingangs **Schaltflächenbeschriftung** des Express-VIs "Dateidialog" ändern. Wenn dieses Express-VI in einem anderen VI verwendet wird, überprüfen Sie das VI, um sicherzustellen, dass die neue Standardbeschriftung **OK** auch dem Zweck Ihres VIs gerecht wird.

Die Funktion "Online-Hilfe steuern"

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

Anzeigen des Frontpanels nach dem Laden

Wenn Sie in LabVIEW ab Version 7.x 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 "Frontpanel:Öffnen" programmatisch geöffnet werden.

Funktion "VI-Referenz öffnen"

Wird in LabVIEW 6.1 und Vorgängerversionen der Parameter "Optionen" der Funktion "VI-Referenz öffnen" nicht verbunden, so erstellt LabVIEW ein VI anhand einer Vorlage, sofern sich die Vorlage nicht bereits im Speicher befindet. Befindet sich die Vorlage im Speicher, erstellt LabVIEW für die Vorlage eine Referenz. Wenn Sie in LabVIEW 7.0 und 7.1 mit Hilfe der Funktion "VI-Referenz öffnen" eine Referenz für eine Vorlage erstellen, die sich bereits im Speicher befindet, gibt die Funktion eine Fehlermeldung aus, es sei denn, Sie legen `0x02` im Parameter "Optionen" fest. Ab LabVIEW 8.0 wird bei Erstellen einer Referenz auf eine Vorlage mithilfe der Funktion "VI-Referenz öffnen" ein VI von der Vorlage erstellt, auch wenn sich diese Vorlage bereits im Speicher befindet.

Exponentialschreibweise

In LabVIEW-Versionen vor 6.1 wird 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 zum Speichern von IVI-Konfigurationen wird nun zwischen Groß- und Kleinschreibung unterschieden. Wenn in Ihrer Applikation 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.

Upgrade von LabVIEW 5.x oder älteren Versionen

Weitere Informationen zum Konvertieren von VIs, die in vorigen Versionen von LabVIEW erstellt wurden, zur Kompatibilität mit neueren Versionen und umgekehrt finden Sie auf der Website von National Instruments unter ni.com/info nach Eingabe des Infocodes `ext8h9`.

Änderungen und Neuerungen in LabVIEW 2009

In der *LabVIEW-Hilfe* finden Sie Informationen zu Funktionen in LabVIEW 2009, Grundlagen der Programmierung mit LabVIEW, schrittweise Anleitungen und Referenzen. Zum Öffnen der *LabVIEW-Hilfe* klicken Sie auf **Hilfe»LabVIEW-Hilfe durchsuchen**.

Weitere Informationen zu bekannten Problemen, den in der aktuellen Version behobenen Fehlern, zusätzlichen Kompatibilitätsproblemen und Informationen zu zuletzt hinzugefügten Funktionen in LabVIEW 2009 finden Sie in der Datei `readme.html` im `labview`-Verzeichnis.

Installation von LabVIEW

(Windows) Mit den zu LabVIEW 2009 gelieferten LabVIEW-Plattform-DVDs können Sie sowohl LabVIEW als auch die meisten dafür erhältlichen Module und Toolkits installieren. Weitere Informationen finden Sie im Abschnitt *Upgrade von Modulen, Toolkits und Gerätetreibern* in diesem Dokument oder im Abschnitt *Installieren von LabVIEW 2009* in den *LabVIEW-Versionshinweisen*.

Neue Beispiel-VIs

Im Ordner **Neue Beispiele für LabVIEW 2009** auf der Registerkarte **Durchsuchen** der NI-Suchmaschine für Beispiele finden Sie neue Beispiel-VIs und Beschreibungen dazu. Die Beispiele können direkt gestartet werden.

Verbesserungen am Blockdiagramm

In LabVIEW 2009 gibt es folgende Verbesserungen am Blockdiagramm und dazugehörigen Funktionen:

Automatisches Aufräumen eines ausgewählten Blockdiagrammabschnitts

Objekte und Verbindungen im Blockdiagramm können automatisch neu geordnet werden. Markieren Sie dazu einen Bereich des Blockdiagramms und wählen Sie **Bearbeiten»Auswahl aufräumen**. Wenn Verbindungen und Objekte in einer Struktur nicht automatisch neu angeordnet werden sollen, klicken Sie mit der rechten Maustaste auf die Struktur und wählen Sie aus dem Kontextmenü die Option **Von Blockdiagrammsäuberung ausschließen** aus.

Aktivieren der parallelen Ausführung von For-Schleifen

Die Iterationen von For-Schleifen werden in LabVIEW nacheinander ausgeführt. Ist eine For-Schleife rechenintensiv, sollten Sie die Iterationen der Schleife zur Verbesserung der Leistung parallel ausführen. Zum Suchen nach parallel ausführbaren For-Schleifen wählen Sie **Werkzeuge»Profil»Parallel ausführbare Schleifen suchen**. Die ermittelten Schleifen werden im Fenster **Ergebnisse der Suche nach parallel ausführbaren Schleifen** angezeigt. Wenn Sie das Fenster vom **Projekt-Explorer** aus öffnen, analysiert LabVIEW alle VIs im Projekt.

Zur Anzeige des Dialogfelds **Parallele Ausführung von For-Schleifen-Iterationen** klicken Sie mit der rechten Maustaste auf den Rahmen der For-Schleife und wählen Sie **Parallele Ausführung von**

Iterationen konfigurieren aus dem Kontextmenü. In diesem Dialogfeld können Sie die parallele Ausführung von Iterationen einer For-Schleife aktivieren und konfigurieren.

Verbesserungen an Umgebungsvariablen

In LabVIEW 2009 wurden folgende Verbesserungen an Umgebungsvariablen vorgenommen:

Festkommaunterstützung

Umgebungsvariablen, einschließlich der I/O-Variablen, arbeiten mit dem Festkommatentyp.

Programmatisches Suchen von Variablen und programmatischer Datenaustausch mit Variablen

LabVIEW 2009 enthält neue Funktionen, Klassen, Eigenschaften und Methoden, mit denen Sie Netzwerk-Umgebungsvariablen und I/O-Variable programmatisch suchen und programmatisch Daten mit den Variablen austauschen können. Zum programmatischen Abfragen und Ändern von Umgebungsvariablen können Sie entweder die **Datasocket**-VIs und -Funktionen oder die neue flexiblere API für Umgebungsvariablen nutzen. Anders als die Datasocket-API unterstützt die neue API für Umgebungsvariablen auch I/O-Variablen und umfasst Klassen, die mit Variablen-Engines und -Containern arbeiten.

Scan-I/O-Unterstützung

In LabVIEW 2009 können Sie mit Hilfe der NI Scan Engine und der I/O-Variablen eine durch einen Scan-I/O-Vorgang durchführen. Bisher bestand diese Möglichkeit nur im LabVIEW Real-Time Module. Die NI Scan Engine ermöglicht eine effiziente Arbeit mit Einzelwerten auf mehreren Datenkanälen, wie z. B. I/O-Kanälen. Während eines Scan-Vorgangs werden Daten in einem globalen Speicherabbild gespeichert und alle Werte mit der gleichen Rate aktualisiert, die über die sogenannte Scan-Periode angegeben wird. Eine I/O-Variable ist eine bestimmte Art von Umgebungsvariable, bei der einzelne Daten über die NI Scan Engine ausgetauscht werden. Zum Veröffentlichungszeitpunkt von LabVIEW 2009 wird die NI Scan Engine von den Treibern für CompactRIO-, DeviceNet- und EtherCAT-Hardware sowie für Sensornetze (WSNs) unterstützt. Vor der nächsten LabVIEW-Version können jedoch noch weitere Treiber hinzukommen, die mit der NI Scan Engine arbeiten. Weitere Informationen zur Unterstützung der NI Scan Engine finden Sie in der Beschreibung Ihres Treibers.

Timing-Quellen für zeitgesteuerte Strukturen

LabVIEW 2009 enthält folgende Timing-Quellen für zeitgesteuerte Schleifen (mit einem oder mehreren Rahmen) und für zeitgesteuerte Sequenzstrukturen:

- **1 kHz <absolute Zeit>**
- **1 MHz <absolute Zeit>**

Mit den neuen Timing-Quellen **1 kHz <absolute Zeit>** und **1 MHz <absolute Zeit>** können Sie den Ausführungsbeginn zeitgesteuerter Strukturen auf mehreren Computern synchronisieren.

Allgemeine Verbesserungen am Blockdiagramm

In LabVIEW 2009 gibt es folgende allgemeine Verbesserungen im Blockdiagramm:

- Bei einem Rechtsklick auf einen Umgebungsvariablenknoten und Auswahl von **Variable in Projekt anzeigen** wird der Name der betreffenden Umgebungsvariablen im **Projekt-Explorer** hervorgehoben.
- Benutzerdefinierte Sonden stehen im LabVIEW Base Package zur Verfügung.

Verbesserungen am Frontpanel

In LabVIEW 2009 gibt es folgende Verbesserungen am Frontpanel und dazugehörigen Funktionen:

Verbesserungen an 2D-Graphen

Die Palette **Graph** enthält folgende neue 2D-Graphen:

- 2D-Kompassplot
- 2D-Fehlerplot
- 2D-Federplot
- XY-Plotmatrix

Verbesserungen an 3D-Graphen

Die Palette **3D-Graph** enthält folgende neue 3D-Graphen:

- 3D-Balkenplot
- 3D-Kometenplot
- 3D-Konturplot
- 3D-Netzplot
- 3D-Tortenplot
- 3D-Köcherplot
- 3D-Bandplot
- 3D-Streuplot
- 3D-Stabplot
- 3D-Oberflächenplot
- 3D-Wasserfallplot

Allgemeine Verbesserungen am Frontpanel

In LabVIEW 2009 gibt es folgende allgemeine Verbesserungen am Frontpanel:

- In einem digitalen Signalverlaufsgraphen werden automatisch die Namen der Leitungen und Busse angezeigt, die Sie für die digitalen Signale festgelegt haben.
- Der Name des 3D-Bildelements hat sich in 3D-Bild geändert.
- Der Name des Bildanzeigeelements hat sich in 2D-Bild geändert. 2D-Bildelemente sind nun auch im LabVIEW Base Package enthalten.
- Unterpanel-Elemente und Baumstrukturelemente sind nun ebenfalls im LabVIEW Base Package enthalten.
- Bei einem Rechtsklick auf ein Frontpanel-Element, das mit einer Umgebungsvariable verbunden ist, und Auswahl von **Variable in Projekt anzeigen** aus dem Kontextmenü wird die betreffende Umgebungsvariable im **Projekt-Explorer** hervorgehoben.
- In LabVIEW-Versionen bis einschließlich 8.6 konnte der Benutzer sowohl im Bearbeitungs- als auch im Ausführungsmodus um ein Viertel der Fensterlänge und -breite an sichtbaren Objekten vorbei scrollen. Bei LabVIEW 2009 ist der Bildlaufbereich im Bearbeitungs- und im Ausführungsmodus unterschiedlich groß. So können Sie nun im Ausführungsmodus zehn Pixel am Rahmen der sichtbaren Frontpanel-Objekte vorbei scrollen.

Optische Verbesserungen

In LabVIEW 2009 gibt es folgende Verbesserungen an der LabVIEW-Umgebung:

Verbesserungen an Zugriffsbereichen

Für Bibliotheken, LabVIEW-Klassen und XControls wurde der Zugriffsbereich "Gruppe" hinzugefügt. Auf eine Bibliothek mit dem Zugriffsbereich "Gruppe" können nur Objekte der Bibliothek oder Freund-VIs bzw. Freundbibliotheken zugreifen. Der Zugriffsbereich für die Objekte einer Bibliothek kann auf der Seite **Objekteinstellungen** des Dialogfelds **Eigenschaften für Projektbibliothek** festgelegt

werden. Auf der Seite **Freundobjekte** des Dialogfelds **Eigenschaften für Projektbibliothek** können Sie die Freundobjekte einer LabVIEW-Projektbibliothek festlegen.

Bei der Auswahl eines VIs als Freundobjekt einer Bibliothek kann das VI mit allen Bestandteilen der Bibliothek arbeiten, deren Zugriffsbereich "Gruppe" lautet. Auch Bibliotheken können als Freundobjekte einer Bibliothek festgelegt werden.

Klassen und XControls werden in den Dialogfeldern **Klasseneigenschaften** und **Eigenschaften für XControls** auf den Zugriffsbereich "Gruppe" eingestellt. In diesen Dialogfeldern können auch Freundobjekte festgelegt werden.

Verbesserungen am Assistenten zum Erstellen von Gerätetreiberprojekten

Der Assistent **Neues Gerätetreiberprojekt erstellen** wurde um folgende Vorlagen ergänzt:

- Zähler
- Leistungsmesser

Verbesserungen am Domänen-Kontomanager (Windows)

In LabVIEW 2009 gibt es folgende Verbesserungen am Dialogfeld **Domänen-Kontomanager**:

- Mit Hilfe des **Domänen-Kontomanagers** können Sie eine lokale Domäne exportieren und diese in einen Rechner importieren, mit dem Sie LabVIEW-Applikationen verteilen. Beim Import einer Domäne prüft der **Domänen-Kontomanager**, ob bereits eine lokale Domäne auf dem Rechner vorhanden ist. Ist bereits eine Domäne vorhanden, werden Sie vom **Domänen-Kontomanager** aufgefordert, vor dem Import der neuen Domäne eine Kopie der vorhandenen lokalen Domäne zu speichern. Mit dem **Domänen-Kontomanager** ist es nicht möglich, eine Domäne zu importieren, wenn bereits eine solche im Netzwerk vorhanden ist.
- Der Menüpunkt **Lookout 4.x Sicherheitsdatei importieren** wurde umbenannt in **Lookout Sicherheitsdatei importieren** und ist nicht mehr über **Datei»Neu** im **Domänen-Kontomanager** zugänglich. Wählen Sie stattdessen zum Import eines Lookout-Kontos **Datei»Importieren und Exportieren»Lookout Sicherheitsdatei importieren** oder klicken Sie die Domänenliste mit der rechten Maustaste an und wählen Sie **Importieren und Exportieren»Lookout Sicherheitsdatei importieren** aus dem Kontextmenü.

Verbesserungen am Startfenster

Im **Startfenster** von LabVIEW 2009 finden Sie unter **Neuigkeiten von ni.com** fortlaufend aktualisierte Neuigkeiten, technische Informationen, Beispielprogramme und Schulungsdaten von der Website ni.com. Diesen Kategorien werden dynamisch Artikel über LabVIEW und installierten Modulen und Toolkits hinzugefügt. Wenn neue Artikel hinzukommen, wird neben dem Namen der betreffenden Kategorie die Anzahl der ungelesenen Artikel angezeigt.



Hinweis In den nicht englischen Ausgaben von LabVIEW ist die Suche nach Neuigkeiten auf ni.com per Voreinstellung deaktiviert. Nur die englische LabVIEW-Ausgabe unterstützt die automatische Suche nach Neuigkeiten auf ni.com. In den übersetzten Ausgaben von LabVIEW führen die Links in dieser Kategorie stets zu den gleichen Seiten auf ni.com.

Verbesserungen an der LabVIEW-Klassenhierarchie

In LabVIEW 2009 gibt es folgende Verbesserungen am Fenster **LabVIEW-Klassenhierarchie**:

- Die Farbe des Hintergrunds, der Verbindungslinien und der VI-Ränder in der **VI-Klassenhierarchie** wurde geändert, um die Darstellung der Objekte benutzerfreundlicher zu gestalten.
- In der **VI-Klassenhierarchie** werden Knoten entsprechend ihrer Bibliothekszugehörigkeit in Gruppen zusammengefasst. Knoten einer Bibliothek werden mit der gleichen Randfarbe dargestellt.

- Das Menü **Ansicht** im Fenster **LabVIEW-Klassenhierarchie** enthält folgende neue Menüpunkte: **Immer Beschriftungen anzeigen**, **Hineinzoomen**, **Herauszoomen**, **Tatsächliche Größe** und **An Fenster anpassen**.

Verbesserungen der Ausführungseffizienz

LabVIEW 2009 enthält eine aktualisierte Version der Math Kernel Library (MKL) 10.0 für Windows und Linux. MKL ist ein Programm eines anderen Herstellers, mit dem in LabVIEW die Ausführung von VIs zur linearen Algebra verbessert wird. Weitere Informationen zu MKL finden Sie im Abschnitt *Copyright der LabVIEW-Hilfe*.

Verbesserungen an der Suche

In den Dialogfeldern **Schnelleinfügeliste** und **Paletten durchsuchen** wird ein neuer Suchalgorithmus verwendet, mit dem alle Elemente, VIs und Funktionen nach Stichwörtern durchsucht und die Suchergebnisse nach Relevanz statt alphabetisch angezeigt werden.

Verbesserungen am Fenster "VI-Hierarchie"

In LabVIEW 2009 gibt es folgende Verbesserungen am Fenster **VI-Hierarchie**:

- Die Farbe des Hintergrunds, der Verbindungslinien und der VI-Ränder in der **VI-Hierarchie** wurde geändert, um die Darstellung der Objekte benutzerfreundlicher zu gestalten.
- Das **VI-Hierarchie**-Fenster zeigt Umgebungsvariablen an.
- Das **VI-Hierarchie**-Fenster zeigt rekursive Beziehungen an, indem miteinander in Verbindung stehende VIs mit einer gestrichelten Linie verbunden werden.
- In der **VI-Hierarchie** werden Knoten entsprechend ihrer Bibliothekszugehörigkeit in Gruppen zusammengefasst. Knoten einer Bibliothek werden mit der gleichen Randfarbe dargestellt.
- Das Menü **Ansicht** im Fenster **VI-Hierarchie** enthält folgende neue Menüpunkte: **Immer Beschriftungen anzeigen**, **Hineinzoomen**, **Herauszoomen**, **Tatsächliche Größe** und **An Fenster anpassen**.

Verbesserungen an Dialogfeldern

In LabVIEW 2009 wurden folgende Verbesserungen an Dialogfeldern durchgeführt:

Verbesserungen am Dialogfeld "Klasseneigenschaften"

In LabVIEW 2009 gibt es folgende Änderungen am Dialogfeld **Klasseneigenschaften**:

- Aktivieren Sie auf der Seite **Objekteinstellungen** die Option **Beim Überschreiben dieses dynamischen Dispatch-VIs immer Elternmethode aufrufen**, wenn alle dynamischen Dispatch-VIs, die das angegebene VI überschreiben, mit dem Knoten "Elternmethode aufrufen" arbeiten sollen.
- Aktivieren Sie die Option **Abgeleitete Klasse muss dieses dynamische Dispatch-VI überschreiben** auf der Seite **Objekteinstellungen**, wenn alle Kindklassen ein Überschreib-Member-VI des ausgewählten VIs definieren sollen.
- Aktivieren Sie die Option **Alle Anforderungen für "Muss überschreiben?" an Nachfolger übertragen** auf der Seite **Vererbung**, wenn eine Klasse in der Lage sein soll, alle Überschreibobligationen für dynamische Dispatch-VIs an Nachfolgerklassen zu übertragen.
- Aktivieren Sie die Option **Referenzen dieses Klassentyps auf Member-VIs dieser Klasse beschränken** auf der Seite **Vererbung**, wenn Datenwertreferenzen auf eine Klasse nur von Member-VIs erstellt oder gelöscht werden sollen.
- Aktivieren Sie auf der Seite **Vererbung** die Option **Referenzen von Nachfolgerklassentypen auf Member-VIs dieser Klasse beschränken**, wenn Datenwertreferenzen auf eine Nachfolgerklasse nur von Member-VIs erstellt oder gelöscht werden sollen.

Verbessertes Dialogfeld "Optionen"

In LabVIEW 2009 gibt es folgende Verbesserungen am Dialogfeld **Optionen**:

- Die Seiten des Dialogfelds **Optionen** sind zur Verbesserung der Benutzerfreundlichkeit in Abschnitte gegliedert.
- Die Seiten **Farben**, **Schriftarten** und **Fehlersuche** sind nun Abschnitte der Seite **Umgebung**.
- Die Optionen auf der Seite **Ausrichtungsgitter** befinden sich jetzt auf den Seiten **Frontpanel** und **Blockdiagramm**.
- Die **VI-Server**-Seiten wurden zu einer Seite zusammengefasst.
- Die **Webserver**-Seiten wurden zu einer Seite zusammengefasst.
- Selten genutzte Optionen wurden aus dem Dialogfeld **Optionen** entfernt. In der [KnowledgeBase](#) finden Sie eine Liste mit diesen Optionen und Informationen zu alternativen Vorgehensweisen.
- Mit Hilfe der Option **Zahlen in Symbolen neuer VIs verwenden (1 bis 9)** auf der Seite **Frontpanel** können Sie festlegen, ob die Symbole der ersten neun nach dem Start von LabVIEW erstellten VIs automatisch durchnummeriert werden sollen. Ausgenommen sind VIs, die anhand von Vorlagen erstellt wurden.
- Mit Hilfe der Option **Im Startfenster Suche nach NI-Neuigkeiten deaktivieren** auf der Seite **Umgebung** können Sie die automatische Suche nach den neuesten Artikeln auf [ni.com](#) deaktivieren.
- Die Voreinstellung für die Option **Zulässige Rückgängig-Schritte pro VI** auf der Seite **Umgebung** wurde von 9 auf 30 geändert.
- Mit der Option **Höchstanzahl zuletzt geöffneter Dateien** auf der Seite **Umgebung** können Sie die Anzahl der Dateien festlegen, die bei der Auswahl von **Datei»Zuletzt geöffnete Dateien** aufgelistet werden.
- Auf der Seite **Webserver** können Sie unter **Erweiterte Einstellungen** festlegen, dass der nicht als Standardexemplar genutzte LabVIEW-Webserver mit SSL-Verschlüsselung arbeiten soll.

Verbesserungen am Dialogfeld "Schnelleinfügeliste"

In LabVIEW 2009 gibt es folgende Verbesserungen am Dialogfeld **Schnelleinfügeliste**:

- In der **Schnelleinfügeliste** werden Objekte aktiver LabVIEW-Projekte nach Namen sortiert angezeigt. Sie können ohne Verwendung des **Projekt-Explorers** nach Objekten suchen.
- Bei angezeigter **Schnelleinfügeliste** können Sie folgende Tastenkombinationen verwenden:
 - <Strg + D>—Erstellt an allen offenen Ein- und Ausgängen der markierten Blockdiagrammobjekte automatisch Bedien- und Anzeigeelemente.
 - <Strg + Shift + D>—Erstellt an allen offenen Eingängen der markierten Blockdiagrammobjekte automatisch Konstanten.
 - <Strg + R>—Entfernt die markierten Blockdiagrammobjekte einschließlich der Verbindungsstücke und Konstanten, die mit ihren Ein- und Ausgängen verbunden sind. Wenn die gelöschten Objekte mit einem Verbindungsstück eines bestimmten Datentyps verbunden waren, wird dieses automatisch durch ein Verbindungsstück des gleichen Datentyps ersetzt.
 - <Strg + T>—Verschiebt die Beschriftungen aller Bedienelemente auf die linke Seite des Blockdiagrammsymbols und die Beschriftungen aller Anzeigeelemente auf die rechte Seite.

Allgemeine Verbesserungen an Dialogfeldern

In LabVIEW 2009 gibt es folgende allgemeine Verbesserungen an Dialogfeldern:

- Die Fenster **Navigation** und **Pufferzuweisungen anzeigen** sind im LabVIEW Base Package verfügbar.
- Die Seite **Server** des Dialogfelds **Eigenschaften für Webdienst** wurde in **Diensteinstellungen** umbenannt. Das Dialogfeld **Eigenschaften für Webdienst** enthält Einstellungen zu Webservern, Dienst-Aliasnamen und Standarddokumenten/-Webmethoden für Webdienste.

- Auf der Seite **Fortgeschritten** des Dialogfelds **Eigenschaften für Webdienst** können Sie die Fehlersuche eines Webdienstes über das Netzwerk aktivieren und den Build-Prozess des Webdienstes protokollieren.
- Die Seite **Scan Engine**, die in den bisherigen LabVIEW-Versionen nur bei installiertem LabVIEW Real-Time Module angezeigt wurde, ist jetzt im LabVIEW Base Package im Dialogfeld **Mein Computer: Eigenschaften** enthalten.
- Im Dialogfeld **Variablen suchen** werden Umgebungsvariablen angezeigt, die Sie in einem Element oder einer Konstante für Variablenreferenzen verwenden können, um programmatisch Daten mit einer Variablen auszutauschen oder Variablen programmatisch zu suchen.
- Auf der Seite **Konfiguration** des Dialogfelds **Eigenschaften für Variablenreferenz** enthält Optionen zur Konfiguration des Datentyps, Zugriffs und Zeitstempels für Elemente oder Konstanten für Variablenreferenzen.
- Das Dialogfeld **Ereignisse bearbeiten** wurde benutzerfreundlicher gestaltet.
- Das Dialogfeld **Softwarepaket suchen** wurde abgeschafft. Um erforderliche Programmkomponenten auf den Computer zu kopieren, aktivieren Sie die Option **Ausgewählte Programme auf diesen Computer kopieren** auf der Seite **Weitere Installationsprogramme** des Dialogfelds **Eigenschaften für Installer**.
- Wenn Sie sowohl die 32- als auch die 64-Bit-Version von LabVIEW installiert haben, unterstützt das Dialogfeld **Elemente- und Funktionenpalette bearbeiten** beide Versionen. Änderungen an den Paletten werden in das Verzeichnis `x\Palette\menus` im Standarddatenverzeichnis gespeichert, wobei `x` für die LabVIEW-Version steht. Die Versionsnummer wird entweder durch `(32-bit)` oder `(64-bit)` ergänzt. Wenn Sie beispielsweise eine Palette in der 32-Bit-Version von LabVIEW bearbeiten, werden die Änderungen im Verzeichnis `2009\ (32-bit)\Palettes\menus` gespeichert.

Verbesserungen an der Elementepalette

Die Paletten von LabVIEW 2009 wurden folgendermaßen verbessert:

- Auf der Palette **System** gibt es die folgenden Unterpaletten:
 - Numerisch
 - Boolesch
 - String & Pfad
 - Ring & Enum
 - Container
 - Liste, Tabelle & Baumstruktur
 - Gestaltungselemente
- Die Palette **Liste & Tabelle** wurde in **Liste, Tabelle & Baumstruktur** umbenannt.
- Die Palette **Container** enthält ein Unterpanel-Element.

Allgemeine Verbesserungen an der Programmoberfläche

In LabVIEW 2009 gibt es folgende allgemeine Verbesserungen an der Programmoberfläche:

- Das Menü **Ansicht** wurde um den Menüpunkt **Dieses VI im Projekt** ergänzt.
- Das Menü **Fenster** enthält den neuen Menüpunkt **Projekt anzeigen**.
- Der Menüpunkt **Werkzeuge»Instrumentierung»Fortgeschrittene Entwicklung»Icon Art Glossary anzeigen** ist nicht mehr vorhanden. Verwenden Sie stattdessen die Seite **Grafiken** des **Symbol-Editors**. Diese enthält alle Grafiken der Icon Library auf ni.com.

- In LabVIEW erzeugte Grafikdateien werden mit ihrem vollständigen qualifizierten Namen aufgeführt.

Verbesserungen am LabVIEW Application Builder

In LabVIEW 2009 wurden die nachfolgend genannten Verbesserungen am LabVIEW Application Builder vorgenommen, der im **Projekt-Explorer** durch einen Rechtsklick auf **Build-Spezifikationen** und Auswahl der zu erstellenden Software gestartet wird.

- Der Application Builder verschiebt Dateien, deren Namen in den Build-Spezifikationen Konflikte verursachen, nicht mehr in Ordner außerhalb der Applikation, DLL oder des Webdienstes. In LabVIEW-Versionen bis einschließlich 8.6 wurden die VIs und Bibliotheken in Form einer einfachen Liste zusammen mit der Applikation gespeichert und alle Dateien mit identischen Namen wurden in Verzeichnisse aus der Applikation verschoben. Wenn Sie mit LabVIEW 2009 eine eigenständige Applikation oder DLL erstellen, speichert der Application Builder alle Quelldateien in der Applikation, wobei die Dateien in einer ähnlichen Struktur wie auf dem Datenträger angeordnet sind. In der folgenden Tabelle werden die relativen Pfade für ein Haupt-VI mit dem Namen `muster.vi` aufgeführt, das die SubVIs `a.vi` und `b.vi` aufruft. `C:\...\Application.exe` steht für den Pfad zur Applikation.

Pfad zu den Quelldateien	Pfad zu den Dateien der Applikation
<code>C:\Quelldateien\muster.vi</code>	<code>C:\...\Application.exe\muster.vi</code>
<code>C:\Quelldateien\xxx\a.vi</code>	<code>C:\...\Application.exe\xxx\a.vi</code>
<code>C:\Quelldateien\yyy\b.vi</code>	<code>C:\...\Application.exe\yyy\b.vi</code>

Wenn Sie lieber mit dem bisherigen Datei-Layout arbeiten möchten, aktivieren Sie auf der Seite **Fortgeschritten** des Dialogfelds **Eigenschaften für Applikation**, **Eigenschaften für DLL** oder **Eigenschaften für Webdienst** die Option **Datei-Layout von LabVIEW 8.x verwenden**. Bei Build-Spezifikationen aus einer älteren LabVIEW-Version ist diese Option standardmäßig aktiviert.

- **(Windows)** Mit Hilfe der Build-Spezifikation für .NET-Interop-Assemblys lassen sich VIs in einem LabVIEW-Projekt erstellen und als .NET-Interop-Assemblys verteilen. Klicken Sie zur Anzeige des Dialogfelds **Eigenschaften für .NET-Interop-Assembly** mit der rechten Maustaste im **Projekt-Explorer** auf **Build-Spezifikationen** und wählen Sie **Neu» .NET-Interop-Assembly** aus dem Kontextmenü aus.
- Mit dem Dialogfeld **Applikation von VI erstellen** können Sie das aktuell geöffnete VI in eine eigenständige Applikation umwandeln.

Verbesserungen am Dialogfeld "Application Builder"

In LabVIEW 2009 gibt es folgende Verbesserungen am LabVIEW Application Builder:

- Mit der Option **Build-Protokolldatei erzeugen** auf der Seite **Fortgeschritten** der Dialogfelder **Eigenschaften für Applikation**, **Eigenschaften für DLL** und **Eigenschaften für Webdienst** kann eine Protokolldatei zum Build erzeugt werden. Darin sind z. B. die Namen der Quelldateien, eventuelle Build-Fehler sowie der Anfangs- und Endzeitpunkt des Build-Prozesses aufgeführt.
- Auf der Seite **Übertragung von Umgebungsvariablen** des Dialogfelds **Eigenschaften für Applikation** können Sie festlegen, dass abhängige Bibliotheken mit Umgebungsvariablen während der Ausführung auf das Zielsystem übertragen werden sollen.
- Auf der Seite **Fortgeschritten** des Dialogfelds **Eigenschaften für Quellcodepaket** können Sie eine Protokolldatei zum Build erstellen. Diese enthält z. B. eine Liste der Quelldateien, eventuelle Build-Fehler und eine Zeitangabe für den Beginn und das Ende des Build-Prozesses.
- Mit der Seite **Windows-Sicherheit** der Dialogfelder **Eigenschaften für Applikation**, **Eigenschaften für DLL** und **Eigenschaften für .NET-Interop-Assembly** können Sie Sicherheitseinstellungen

zu Windows-Computern vornehmen. Hier können Sie z. B. eine digitale Signatur oder eine Manifestdatei in den Build einbinden.

Verbesserungen an LabVIEW-Projekten

In LabVIEW 2009 gibt es folgende Verbesserungen an LabVIEW-Projekten und dem damit verbundenen Funktionsumfang:

Allgemeine Verbesserungen an Projekten

In LabVIEW 2009 gibt es folgende allgemeine Verbesserungen an Projekten:

- Mit Hilfe des Dialogfelds **Warum befindet sich dieses Objekt in Abhängigkeiten?** können Sie alle Projektobjekte finden, die zwar nicht unter "Abhängigkeiten" aufgelistet, aber vom ausgewählten Objekt abhängig sind.
- Mit dem Dialogfeld **Freundobjekte suchen** können Sie alle Freundobjekte einer bestimmten Klasse im Projekt suchen.
- Mit dem Dialogfeld **Unterobjekte suchen** können Sie alle Unterobjekte einer bestimmten Klasse im Projekt suchen.
- Mit dem Dialogfeld **Aufrufer suchen** können Sie alle VIs im Projekt suchen, die über einen Umgebungsvariablenknoten mit einer bestimmten Umgebungsvariable arbeiten.
- Probleme in Bezug auf Umgebungsvariablen, die in der Vergangenheit beim Speichern von Kopien komplexer Projektbibliotheken auftraten, wurden beseitigt.

Neue und verbesserte VIs, Funktionen und Knoten

Die nachfolgend aufgeführten VIs, Funktionen und Knoten wurden in LabVIEW 2009 hinzugefügt oder verändert. Weitere Informationen zu VIs, Funktionen und Knoten finden Sie im Buch **VI- und Funktionenübersicht** auf der Registerkarte **Inhalt** in der *LabVIEW-Hilfe*.

Neue VIs und Funktionen

LabVIEW 2009 enthält folgende neue VIs und Funktionen:

VIs zur fortgeschrittenen Kurvenanpassung

Auf der Palette **Fortgeschrittene Kurvenanpassung** gibt es folgende neue VIs:

- "Koeffizienten der allgemeinen Polynomanpassung"
- "Intervalle der Polynomanpassung"

VIs und Funktionen zur Applikationssteuerung

Auf der Palette **Applikationssteuerung** gibt es die neue Unterpalette **CPU-Informationen** mit folgenden neuen VIs:

- "CPU-Informationen"
- "Daten-Cache-Größe"
- "Anzahl der Cache-Level"

VIs für Differentialgleichungen

Auf der Palette **Differentialgleichungen** gibt es die neue Unterpalette **Partielle Differentialgleichungen** mit folgenden neuen VIs:

- "PDE definieren"
- "PDE-Randbedingung definieren"
- "PDE-Bereich definieren"
- "PDE-Anfangsbedingung definieren"
- "PDE-Rendering"

- "PDE-Lösungssystem"

Datei-I/O-VIs und -Funktionen

Auf der Palette **Datei-I/O** sind folgende VIs hinzugekommen:

- "Systemverzeichnis ermitteln"
- "Applikationsverzeichnis"
- "TDMS: Format konvertieren"
- "TDMS: Erstellen von Skalierungsinformationen"

Festkommafunktionen

Auf der Palette **Festkomma** gibt es folgende neue Funktionen:

- "Festkomma in Integer umwandeln"
- "Integer in Festkomma umwandeln"

Matrixfunktionen

Die Paletten **Array** und **Lineare Algebra** enthalten die neue Palette **Matrix** mit folgenden Funktionen:

- "Matrix erstellen"
- "Matrixdiagonale lesen"
- "Matrizelemente lesen"
- "Untermatrix lesen"
- "Matrixgröße"
- "Matrixgröße ändern"
- "Matrixdiagonale festlegen"
- "Matrizelemente festlegen"
- "Untermatrix festlegen"
- "Transponierte Matrix"

VIs, Funktionen und Knoten zur Speicherverwendung

Auf der Palette **Speicherverwendung** gibt es folgende neue Funktionen:

- "Datenwertreferenz löschen"
- "Neue Datenwertreferenz"

Umgebungsvariablenknoten und -VI sowie Funktionen für Umgebungsvariablen

Die Palette **Umgebungsvariable** enthält folgende neue Funktionen:

- "Variablenverbindung beenden"
- "Variablenverbindung herstellen"
- "Variable lesen"
- "Variable schreiben"

Die Palette **Umgebungsvariable** wurde um die neue Unterpalette **I/O-Variable** erweitert, die folgende neue Funktionen enthält:

- "Variable direkt lesen"
- "Variable direkt schreiben"
- "Variable lesen (Scan)"
- "Variable schreiben (Scan)"

Die Palette **Umgebungsvariable** enthält die neue Palette **PSP-Variable** mit folgenden neuen Funktionen:

- "Variablenverbindung herstellen und prüfen"

- "Variablenverbindung im Hintergrund herstellen"

Express-VIs zur Signalverarbeitung

Auf der Palette **Signalverarbeitung** gibt es folgende neue Express-VIs:

- "Attribute für dynamische Daten ermitteln"
- "Digitalsignale gruppieren"
- "Attribute für dynamische Daten festlegen"

Utility-VIs

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

- "LV-Grafik nach PNG-Daten"
- "PNG-Daten nach LV-Grafik"

XML-Parser-VIs und -Funktionen

Auf der Palette **XML-Parser** gibt es folgende neue VIs:

- "Alle übereinstimmenden Knoten ermitteln"
- "Ersten übereinstimmenden Knoten ermitteln"

Verschiedene neue VIs und Funktionen

LabVIEW 2009 wurde um die folgenden weiteren VIs und Funktionen ergänzt:

- Zur Palette **Cluster, Klasse, Variant** wurde die Funktion "Laufzeitklasse erhalten" hinzugefügt.
- Zur Palette **Signaloperation** wurde der "Knoten in Z-Transformationsansicht" hinzugefügt. Dabei handelt es sich um den Rückkopplungsknoten in Z-Transformationsansicht.
- Zur Palette **Anpassung** wurde das VI "B-Spline-Anpassung" hinzugefügt.
- Zur Palette **Webdienste** wurde das VI "Info zu hochgeladenen Dateien lesen" hinzugefügt.

Geänderte VIs, Funktionen und Knoten

Folgende VIs, Funktionen und Knoten haben sich in LabVIEW 2009 geändert:

VIs für fortgeschrittene IIR-Filterung

Auf der Palette **Fortgeschrittene IIR-Filterung** wurden die nachfolgend beschriebenen VIs geändert.

Folgende VIs enthalten einen Eingang namens **Filterstrukturoption** zur Angabe der Ordnung des IIR-Kaskadenfilters:

- "Bessel-Koeffizienten"
- "Butterworth-Koeffizienten"
- "Chebyshev-Koeffizienten"
- "Koeffizienten für elliptischen Filter"
- "Inverse Chebyshev-Koeffizienten"

Konnektivitäts-VIs und -Funktionen (Windows)

Auf der Palette **Konnektivität** wurden folgende VIs geändert:

- Das VI "Registry: Schlüssel erzeugen" enthält einen Eingang mit dem Namen **Registry-Ansicht**, mit dem festgelegt wird, ob ein 32-Bit-Schlüssel oder 64-Bit-Schlüssel auf einem 64-Bit-Betriebssystem erstellt werden soll.



Hinweis Die folgenden VIs gibt es nur im LabVIEW Full und Professional Development System:

- **"ESP-Variable festlegen"**—Enthält polymorphe Instanzen für Strings, boolesche Werte und Zahlen.
- **"Session-Variable schreiben"**—Enthält polymorphe Instanzen für Strings, boolesche Werte und Zahlen.

VIs und Funktionen für digitale Signalverläufe

Auf der Palette **Digitaler Signalverlauf** wurden folgende VIs geändert:

- **"Digitale Signale anfügen"**—In der DWDT-Instanz werden die Namen der Digitalsignale kombiniert.
- **"Abschnitt des Digitalsignals"**—In der DWDT-Instanz werden die Namen der Signale im Abschnitt des digitalen Signalverlaufs ausgegeben, sofern die Signale benannt sind.

Datei-I/O-VIs und -Funktionen

Auf der Palette **Datei-I/O** wurden folgende VIs geändert:

- **"Konfigurationsdatei schließen"**—Der Eingang **Konfigurationsdaten schreiben?** wurde in **Bei Änderungen Datei schreiben** umbenannt.
- **"Schlüsselnamen lesen"**—Der Ausgang **Gefunden?** wurde in **Abschnitt vorhanden?** umbenannt.
- **"Konfigurationsdatei öffnen"**—Enthält den Ausgang **Datei erstellt?**, der beim Erstellen einer Datei auf TRUE wechselt. Der Eingang **Datei bei Bedarf erstellen?** wurde in **Datei bei Bedarf erstellen** umbenannt.
- **"Rekursive Dateiliste"**—Enthält einen Eingang des Typs **Bei Fehler Rekursionen fortsetzen?**, der angibt, ob rekursive Aufrufe von Verzeichnissen bei Auftreten eines Fehlers fortgesetzt werden sollen. Das VI enthält außerdem einen Eingang mit dem Namen **Auszuschließende Ordner**, mit dessen Hilfe Sie bestimmte Ordner aus der Rekursion ausschließen können.
- **"Abschnitt entfernen"**—Der Ausgang **Gefunden?** wurde in **Abschnitt vorhanden?** umbenannt.
- **"Messwerte aus Datei lesen"**—Der Eingang **Datei neu öffnen** wurde in **Zurücksetzen** umbenannt.
- **"Schlüssel schreiben"**—Enthält den Ausgang **Gefunden?**, der auf TRUE wechselt, wenn das VI den Schlüssel im angegebenen **Abschnitt** gefunden hat.

VIs zur Anpassung

Auf der Palette **Anpassung** gibt es folgende geänderte VIs:

- **"Exponentialanpassung"**—Enthält den Eingang **Parameter Grenzen** mit den oberen und unteren Grenzwerten für **Amplitude**, **Dämpfung** und **Offset**. Des Weiteren gibt es den Ausgang **Offset**, der den Offset des angepassten Modells ausgibt. Der Eingang **Verfeinern?** wurde entfernt.
- **"Gaußsche Spitzenwertanpassung"**—Enthält den Eingang **Parameter Grenzen** mit den oberen und unteren Grenzwerten für **Amplitude**, **Mitte**, **Standardabweichung** und **Offset**. Des Weiteren gibt es den Ausgang **Offset**, der den Offset des angepassten Modells ausgibt.
- **"Allgemeine lineare Anpassung"**—Enthält den Eingang **Toleranz**, mit dem bestimmt wird, wann die iterative Anpassung der **Koeffizienten** beendet werden soll. Des Weiteren gibt es den Eingang **Methode**, mit dem das Anpassungsverfahren festgelegt wird. Der Name des VIs wurde von "Allgemeine LS-Linearanpassung" in "Allgemeine lineare Anpassung" geändert.

- **"Allgemeine Polynomanpassung"**—Enthält den Eingang **Toleranz**, mit dem bestimmt wird, wann die iterative Anpassung der **Polynomkoeffizienten** beendet werden soll. Des Weiteren gibt es den Eingang **Methode**, mit dem das Anpassungsverfahren festgelegt wird.
- **"Lineare Anpassung"**—Enthält den Eingang **Parametergrenzen** mit den oberen und unteren Grenzwerten für **Steigung** und **Achsenabschnitt**.
- **"Logarithmische Anpassung"**—Enthält den Eingang **Parametergrenzen** mit den oberen und unteren Grenzwerten für **Amplitude** und **Skalierung**.
- **"Potenzanpassung"**—Enthält den Eingang **Parametergrenzen** mit den oberen und unteren Grenzwerten für **Amplitude**, **Potenz** und **Offset**. Des Weiteren gibt es den Ausgang **Offset**, der den Offset des angepassten Modells ausgibt.

Mathematik-VIs

Auf der Palette **Mathematik** wurden folgende VIs geändert:

- **"Integral x(t)"**—Enthält den Eingang **Integrationsverfahren**, mit dem die Art der numerischen Integration festgelegt wird. Des Weiteren haben sich die Eingänge **Anfangsbedingung** und **Endbedingung** geändert, um die Anfangs- und Endbedingungen von **X** bei der Integration basierend auf dem **Integrationsverfahren** anzugeben. Der Datentyp von **Anfangsbedingung** und **Endbedingung** hat sich von "Fließkommazahl doppelter Genauigkeit" in "1D-Array aus Fließkommazahlen doppelter Genauigkeit" geändert.
- **"Charakteristisches Polynom der Matrix"**—Ist ein polymorphes VI mit den Instanzen "Charakteristisches Polynom von reellen Zahlen" und "Charakteristisches Polynom von komplexen Zahlen".
- **"Numerische Integration"**—Der Datentyp des Eingangs **Integrationsverfahren** wurde in allen drei Instanzen des VIs in "32-Bit-Integer mit Vorzeichen" geändert.
- **"Quadratur"**—Enthält folgende neue Instanzen: "2D-Quadratur (VI)", "2D-Quadratur (Formel)", "3D-Quadratur (VI)" und "3D-Quadratur (Formel)".
- **"Riemannsche Zeta-Funktion"**—Der Datentyp des Eingangs **x** und des Ausgangs **z(x)** ist jetzt eine komplexe Zahl mit doppelter Genauigkeit.
- **"Matrixtyp testen"**—Ermöglicht am Eingang **Matrixtyp** die Einstellung **Symmetrisch** oder **Hermiteisch**.

VIs, Funktionen und Knoten zur Speicherverwendung

Auf der Palette **Speicherverwendung** hat sich folgende Struktur geändert:

Die Inplace-Elementstruktur wurde um die Randknoten "Array teilen / Teil-Arrays ersetzen" und "Element zum Lesen / Schreiben der Datenwertreferenz" ergänzt.

Protokoll-VIs

Auf der Palette **Protokollerstellung** gab es folgende Änderungen an VIs:

- **"Einfaches Textprotokoll"**—Der Eingang **Drucken oder speichern? (Word/Excel)** wurde in **Drucken oder speichern? (Word/Excel/HTML)** umbenannt.
- **"Protokoll schließen"**—Der Eingang **Änderungen speichern?** ist optional.

Signalerzeugungs-VIs

Auf der Palette **Signalerzeugung** haben sich die nachfolgend beschriebenen VIs geändert.

Die nachfolgenden VIs enthalten den Eingang **Initialisieren?**, mit dem die Verwendung neuer Startwerte für den Rauschgenerator bestimmt wird. **Startwert** legt jetzt auch fest, wie der interne Startwertzustand erzeugt werden soll, wenn **Initialisieren?** TRUE ist.

- **"Bernoulli-Rauschen"**

- "Binäre MLS"
- "Binomisches Rauschen"
- "Gamma-Rauschen"
- "Gaußsches weißes Rauschen"
- "Poisson-Rauschen"
- "Gleichverteiltes weißes Rauschen"

Sound-VIs (Mac OS)

Die **Sound**-VIs unter Mac OS wurden folgendermaßen geändert:



Hinweis Die API von LabVIEW 2009 ist unter Windows, Mac OS und Linux identisch.

- Die VIs unterstützen Mono- und Stereo-Ton.
- Die Daten werden als Signalverlauf dargestellt. Die Daten des **Y**-Arrays können in Form von vorzeichenlosen 8-Bit-Integern, vorzeichenbehafteten 16-Bit-Integern oder vorzeichenbehafteten 32-Bit-Integern sowie mit Daten einfacher und doppelter Genauigkeit dargestellt werden. Jeder Signalverlauf definiert einen Kanal.
- Die Daten sind pulscodemoduliert.
- Die VIs können kontinuierlich Daten ausgeben.
- Bei Wave-Dateien ist eine Streaming-Anzeige möglich.
- Die VIs enthalten eine verbesserte Fehlerprüfung.

TDM-Streaming-VIs und -Funktionen

Auf der Palette **TDM-Streaming** gibt es folgende geänderte Funktionen:

- **"TDMS: Öffnen"**—Enthält den Eingang **"Pufferung deaktivieren"**, der angibt, ob *.tdms-Dateien ohne System-Pufferung geöffnet, erzeugt oder ersetzt werden sollen. Diese Funktion enthält auch den Eingang **"Byte-Reihenfolge"**, mit dem das Endian-Format der Daten in einer *.tdms-Datei festgelegt werden kann.

Diese Funktion enthält außerdem den Eingang **"Dateiformatversion"** zum Festlegen der Version einer *.tdms-Datei. Das Dateiformat 2.0 wurde gegenüber der Version 1.0 folgendermaßen erweitert:

- Sie können Interleave-Daten in eine *.tdms-Datei schreiben.
- *.tdms-Daten können in verschiedenen Endian-Formaten (Byte-Reihenfolgen) geschrieben werden.

Mit Hilfe des VIs **"TDMS: Format konvertieren"** lässt sich die Dateiformatversion einer *.tdms-Datei von 1.0 in 2.0 und umgekehrt konvertieren.

- **"TDMS: Schreiben"**—Wurde um den Eingang **"Daten-Layout"** ergänzt. Mit diesem Eingang wird festgelegt, wie die Streaming-Daten beim Übertragen in die *.tdms-Datei strukturiert werden sollen.

VIs für Signalverlaufsmessungen

Auf der Palette **Signalverlaufsmessungen** gibt es folgende geänderte VIs:

- **"Charakteristiken mehrerer Schwingungen ermitteln"**—Enthält folgende neue Instanzen: "Charakteristiken mehrerer Schwingungen ermitteln (1 Kanal, CDB)", "Charakteristiken mehrerer Schwingungen ermitteln (N Kanäle, CDB)" und "Charakteristiken mehrerer Schwingungen ermitteln (N Kanäle - N Spezifikationen)".

- **"Charakteristiken einer Schwingung ermitteln"**—Enthält folgende neue Instanzen: "Charakteristiken einer Schwingung ermitteln (1 Kanal, CDB)" und "Charakteristiken einer Schwingung ermitteln (N Kanäle, CDB)".

VI zur Nullstellenbestimmung

Auf der Palette **Nullstellen** gibt es folgende geänderte VIs:

- **"Alle Nullstellen von f(x) suchen"**—Ist ein polymorphes VI mit den Instanzen "Alle Nullstellen von f(x) suchen (Formel)" und "Alle Nullstellen von f(x) suchen (VI)". Die Instanz "Alle Nullstellen von f(x) suchen (Formel)" entspricht dem VI "Alle Nullstellen von f(x) suchen" in LabVIEW 8.6.
- **"nD Einzelne Lösung für nichtlineares System"**—Ist ein polymorphes VI mit den Instanzen "nD Einzelne Lösung für nichtlineares System (Formel)" und "nD Einzelne Lösung für nichtlineares System (VI)". Die Instanz "nD Einzelne Lösung für nichtlineares System (Formel)" entspricht dem VI "Einzelne Lösung für nichtlineares System" in LabVIEW 8.6.
- **"nD Nichtlineares System lösen"**—Ist ein polymorphes VI mit den Instanzen "nD Nichtlineares System lösen (Formel)" und "nD Nichtlineares System lösen (VI)". Die Instanz "nD Nichtlineares System lösen (Formel)" entspricht dem VI "nD Nichtlineares System lösen" in LabVIEW 8.6.
- **"Nullstellen nach Newton-Raphson suchen"**—Ist ein polymorphes VI mit den Instanzen "Nullstellen nach Newton-Raphson suchen (Formel)" und "Nullstellen nach Newton-Raphson suchen (VI)". Die Instanz "Nullstellen nach Newton-Raphson suchen (Formel)" entspricht dem VI "Nullstellen nach Newton-Raphson suchen" in LabVIEW 8.6.
- **"Nullstellen nach Ridders suchen"**—Ist ein polymorphes VI mit den Instanzen "Nullstellen nach Ridders suchen (Formel)" und "Nullstellen nach Ridders suchen (VI)". Die Instanz "Nullstellen nach Ridders suchen (Formel)" entspricht dem VI "Nullstellen nach Ridders suchen" in LabVIEW 8.6.

Verschiedene Änderungen an VIs, Funktionen und Knoten

In LabVIEW 2009 gibt es folgende weitere Änderungen an VIs, Funktionen und Knoten:

- Der Eingang **n- oder n-1-dimens. Array** der Funktion "In Array einfügen" wurde dahingehend geändert, dass bei einem Array aus Referenzen an **n-dimens. Array** der Eingang **n- oder n-1-dimens. Array** eine Referenz oder ein Array aus Referenzen mit der gleichen Kindklasse wie **n-dimens. Array** sein muss.
- Die VIs der Paletten **Eigenschaften für 3D-Graph**, **Cursor**, **Grafikformate**, **Bildfunktionen**, **Bildplots** und **SMTP (E-Mail)** sind nun auch im LabVIEW Base Package enthalten.
- Die Symbole für die VIs "Neu" und "Schließen" haben sich geändert.
- Für den Rückkopplungsknoten wurden neue Konfigurationsoptionen hinzugefügt. Sie können den Knoten in Z-Transformationsansicht anzeigen, eine Verzögerung für den Knoten angeben, den Knoten aktivieren oder deaktivieren und den Zeitpunkt der Initialisierung des Knotens bestimmen.
- Die Funktionsweise der **Konfigurationsdatei**-VIs sowie die Art und Weise, in der diese Daten aus *.ini-Dateien für Abschnitts- und Schlüsseleingaben auslesen, hat sich geändert. Der eingegebene Schlüssel darf nicht mit einem Semikolon beginnen und jeder Schlüssel muss einen Wert enthalten. Außerdem kann als Abschnittsangabe keine geschlossene eckige Klammer verwendet werden und das VI gibt keine veränderten Kommentare mehr aus.
- Die VIs für die **NI Scan Engine**, die in bisherigen LabVIEW-Versionen nur bei installiertem LabVIEW Real-Time Module angezeigt wurden, sind im LabVIEW Base Package enthalten.

Neue und geänderte Klassen, Eigenschaften, Methoden und Ereignisse

In LabVIEW 2009 gibt es neue VI-Server-Klassen, -Eigenschaften, -Methoden und -Ereignisse. Eine Liste neuer Klassen, Eigenschaften, Methoden und Ereignisse finden Sie unter dem Thema **Änderungen**

und Neuerungen in LabVIEW 2009»Neue VI-Server-Objekte auf der Registerkarte **Inhalt** in der *LabVIEW-Hilfe*.

Eigenschaften und Methoden für 3DPC_OberflächePlot

Die Klasse "3DPC_OberflächePlot" wurde um die Eigenschaft "Cursor-Liste" ergänzt.

Eigenschaften und Methoden von mathematischen Plots

Eine Liste mit neuen Eigenschaften und Methoden, die mit den neuen 2D- und 3D-Graphen verwendet werden können, finden Sie im Buch **Eigenschaften- und Methodenübersicht»Mathematische Plots** auf der Registerkarte **Inhalt** der *LabVIEW-Hilfe*.

Eigenschaften und Methoden für Variablen

LabVIEW 2009 enthält neue Variableneigenschaften und -methoden, mit denen Sie Variablen programmatisch suchen und programmatisch Daten mit Variablen austauschen können.

Eigenschaften und Methoden des VI-Servers

In LabVIEW wurden folgende Änderungen an den VI-Server-Methoden vorgenommen:

- "VI-Version abrufen"—Gibt das LabVIEW-Dateiformat eines VIs aus. Diese Version kann von der LabVIEW-Version abweichen, in der das VI zuletzt gespeichert wurde. Zum Ermitteln der LabVIEW-Version, in der ein VI zuletzt gespeichert wurde, benötigen Sie die Methode "VI-Editor-Version abrufen".
- "Bibliothek:LabVIEW-Datei-Version ermitteln"—Gibt das LabVIEW-Dateiformat einer Bibliothek aus. Diese Version kann von der LabVIEW-Version abweichen, in der die Bibliothek gespeichert wurde.
- "Quellbereich:Festlegen"—Wurde um den Zugriffsbereich "Gruppe" ergänzt.

Eigenschaften und Methoden für XML-Parser

LabVIEW 2009 enthält folgende neue XML-Parser-Eigenschaften:

- Die Klasse "Knoten" wurde um die Methode "XML erfassen (Quelltextformatierung)" ergänzt
- Die Klasse "Dokument" wurde um die Methode "Datei speichern (Quelltextformatierung)" ergänzt

Verbesserungen an der objektorientierten Programmierung in LabVIEW

In LabVIEW 2009 können Sie VIs, die mit LabVIEW-Klassen oder VIs einer LabVIEW-Klasse aufrufen, auf RT-Systeme übertragen. Beim Übertragen eines VIs auf ein Zielsystem, in dem eine Klasse verwendet oder ein VI einer Klasse aufgerufen wird, überträgt LabVIEW nur die Klassen und VIs auf das System, mit denen die Applikation tatsächlich arbeitet. Wenn die auf das System übertragenen VIs auf ein dynamisches Dispatch-VI einer Klasse verweisen, werden darüber hinaus alle VIs auf das System kopiert, die dynamische Dispatch-VIs in untergeordneten Klassen überschreiben.

Verbesserungen an LabVIEW-Webdiensten (Windows, Nicht im Basispaket enthalten)

In LabVIEW 2009 gibt es folgende Verbesserungen an Webdiensten:

- Die Option **Dienst-Aliasnamen** auf der Seite **Diensteinstellungen** des Dialogfelds **Eigenschaften für Webdienst** bietet die Möglichkeit, den **Dienst** in der benutzerdefinierten Browser-URL durch einen Alias zu ersetzen.
- Beim Hinzufügen eines Webmethoden-VIs zur Liste **Dienst-VIs**, die sich auf der Seite **Quelldateien** des Dialogfelds **Eigenschaften für Webdienst** befindet, wird auf der Seite **URL-Mappings** im Dialogfeld **Eigenschaften für Webdienst** eine Standard-URL erzeugt. Sie können diese Standard-URL bei Bedarf ändern oder löschen.
- Webdienste können zusätzliche Datentypen ausgeben, z. B. "Cluster", "Array", "Signalverlauf" oder "Zeitstempel".

- Sie können Dateien auf einen Webdienst hochladen.
- Webdienste unterstützen das JSON-Ausgabeformat. Der Ausgabeformattyp kann im Dialogfeld **RESTful-VI konfigurieren** konfiguriert werden.
- Sie können eine benutzerdefinierte Browser-URL erstellen, die mehrere Teile der URL einem Eingangsanschluss eines Webmethoden-VIs zuweist. Dieses Vorgehen erweist sich beim Weitergeben von Dateipfaden oder anderen hierarchisch strukturierten Werten mit Hilfe einer benutzerdefinierten Browser-URL als nützlich.
- Sie können ein Standard-HTML-Dokument (`index.html`) konfigurieren, das von Ordnern für statische Dokumente geladen wird.
- Sie können die Fehlersuche für ein Webmethoden-VI auf Echtzeitsystemen aktivieren.
- Der NI Distributed System Manager (NI-DSM) enthält verbesserte Statistikfunktionen für verteilte Webdienste.
- Sie können ein Standard-Webmethoden-VI für einen Webdienst konfigurieren.

Weitere Informationen zur Verwendung von Webdiensten in LabVIEW erhalten Sie auf der Registerkarte **Inhalt** der *LabVIEW-Hilfe* im Buch **Grundlagen»LabVIEW-Webdienste**.

SSL-Unterstützung für den LabVIEW-Webdienst

Sie können die SSL-Verschlüsselung (Secure Sockets Layer) für den LabVIEW-Webserver aktivieren. Die SSL-Verschlüsselung ermöglicht sichere Verbindungen zum Austausch von Daten zwischen Clients und dem Webserver. SSL ist nützlich für Webserver-Funktionen wie Webdienste, Netzwerk-Frontpanel oder das Veröffentlichen von statischen Bildern und Dokumenten.

Mit Hilfe von SSL-Zertifikaten (X.509) werden verschlüsselte Verbindungen zwischen einem Client und dem Webserver hergestellt. Mit dem voreingestellten selbstsignierten Zertifikat von LabVIEW oder dem NI-DSM können Sie benutzerdefinierte selbstsignierte Zertifikate erstellen und Zertifikatsregistrierungsanforderungen (CSR) an eine Zertifizierungsstelle (CA) versenden.

Vergleichen von VIs über die Kommandozeile oder mit Versionsverwaltungsprovidern

Mit der Anwendung `LVCompare.exe` können zwei VIs miteinander verglichen werden. Zum Aufrufen von `LVCompare.exe` können Sie entweder die Kommandozeile oder einen beliebigen Versionsverwaltungsprovider verwenden. Verwenden Sie die folgende Kommandozeilensyntax:

```
<absoluter Pfad zu VI 1> <absoluter Pfad zu VI 2> [-lvpath <Pfad zu LabVIEW>][ -noattr ][ -nofp ][ -nofppos ][ -nobd ][ -nobdcsm ][ -nobdpos ].
```



Hinweis `LVCompare.exe` ist nur im LabVIEW Professional Development System verfügbar.

Weitere Informationen zum Erstellen von Symbolen erhalten Sie unter **Grundlagen»Entwicklungsrichtlinien»Anleitung»Vergleichen von VIs** auf der Registerkarte **Inhalt** der *LabVIEW-Hilfe*.

Fehlersuche mit dem Sondenüberwachungsfenster

Individuelle frei verschiebbare **Sonden** sind durch das **Sondenüberwachungsfenster** ersetzt worden, in dem alle Sonden der aktuellen Programmkopie von LabVIEW angezeigt und verwaltet werden.

Verbesserungen am Symbol-Editor

In LabVIEW 8.6 und Vorgängerversionen besteht der **Symbol-Editor** aus einfachen Bearbeitungswerkzeugen zum Erstellen von Symbolen in 256 oder 16 Farben oder Schwarz/Weiß. Jeder Symboltyp wird individuell erstellt.

Der **Symbol-Editor** in LabVIEW 2009 bietet verbesserte Bearbeitungswerkzeuge für die Symbolerstellung. Der **Symbol-Editor** enthält Symbolvorlagen, Symbole der Icon Library auf `ni.com`,

Optionen für das Hinzufügen und Formatieren von Symboltext und Unterstützung für die Arbeit mit mehreren Ebenen. Im **Symbol-Editor** erstellte Symbole können im 256-Farben- oder im Schwarz/Weiß-Format gespeichert werden. Sie können im **Symbol-Editor** auch Symbolvorlagen erstellen und für die spätere Verwendung speichern. Die Bilder werden als *.png-Dateien im 256-Farben-Format gespeichert.

Wenn Sie das Symbol eines VIs oder benutzerdefinierten Elements bearbeiten möchten, klicken Sie dieses in der rechten oberen Ecke des Frontpanels, Blockdiagramms oder Element-Editors doppelt an, um zum **Symbol-Editor** zu gelangen. Sie können auch mit der rechten Maustaste auf das Symbol klicken und **Symbol bearbeiten** zur Anzeige des **Symbol-Editors** wählen. Um das Symbol einer Projektbibliothek, eines Zustandsdiagramms, einer Klasse oder eines XControls im **Symbol-Editor** zu bearbeiten, klicken Sie auf der Seite **Allgemeine Einstellungen** des entsprechenden **Eigenschaften**-Dialogfelds auf die Schaltfläche **Symbol bearbeiten**.

Weitere Informationen zum Erstellen von Symbolen erhalten Sie unter **Grundlagen»Erstellen von VIs und SubVIs** auf der Registerkarte **Inhalt** in der *LabVIEW-Hilfe*.

LabVIEW 2009 (64 Bit)

LabVIEW 2009 ist das erste LabVIEW Development System, das auch als 64-Bit-Version erhältlich ist. Wenn Sie die 64-Bit-Version von LabVIEW auf Windows Vista (64 Bit) ausführen, steht mehr Arbeitsspeicher zur Verfügung, als bei einem 32-Bit-Betriebssystem oder der 32-Bit-Version von LabVIEW der Fall wäre. LabVIEW 2009 (64 Bit) enthält fast alle Funktionen der 32-Bit-Version von LabVIEW 2009, darunter auch den LabVIEW Application Builder.

Informationen zur Bestellung der 64-Bit-Version von LabVIEW 2009 finden Sie auf der Website ni.com/info nach Eingabe des Infocodes `lv64bit`.

Unterstützte Hardware

Die 64-Bit-Version von LabVIEW 2009 unterstützt eine Vielzahl von Geräten. So sind beispielsweise Treiber für DAQ-, VISA-, GPIB- oder Bilderfassungsgeräte erhältlich. Für GPIB-Geräte wird mindestens die Version 2.6 des NI-488.2-Treibers für Windows benötigt. Informationen zur Kompatibilität der individuellen Geräte mit der 64-Bit-Version von LabVIEW finden Sie in der Beschreibung zu Ihrem Gerät.

Unterstützte Module

Die 64-Bit-Version von LabVIEW 2009 arbeitet nur mit dem NI Vision Development Module. Weitere Informationen erhalten Sie in der Dokumentation zum NI Vision Development Module. Andere Module, Zusatzpakete oder Toolkits werden nicht unterstützt.

Rekursion

In LabVIEW 2009 kann Rekursion verwendet werden, wenn alle VIs der VI-Hierarchie ablaufinvariant sind und Kopien von mindestens einem VI in der Hierarchie gemeinsam genutzt werden. Wenn Sie ein VI als ablaufinvariant konfiguriert haben, kann es sich selbst als SubVI aufrufen. Dazu muss nur das VI-Symbol in das Blockdiagramm des VIs gezogen werden.

Wiederverwenden von Blockdiagrammabschnitten

Sie können Blockdiagrammabschnitte speichern und später wiederverwenden oder an andere Benutzer von LabVIEW 2009 weitergeben. Gespeicherte Blockdiagrammabschnitte werden in Form einer *.png-Datei gespeichert. Die Grafik enthält sowohl eine Abbildung des Blockdiagrammabschnitts als auch den eigentlichen Code.

Zum Speichern eines Blockdiagrammabschnitts markieren Sie die gewünschte Stelle im Blockdiagramm und wählen Sie dann **Bearbeiten»VI-Abschnitt von Auswahl erstellen**. Nach dem Speichern des

VI-Abschnitts als *.png-Datei kann diese vom Verzeichnis, in dem sie gespeichert wurde, direkt in ein Blockdiagramm verschoben werden. Des Weiteren können Sie die Datei an andere Benutzer von LabVIEW 2009 senden.

Verwenden des TDM-Excel-Zusatzpakets (Windows)

Mit Hilfe des TDM-Excel-Zusatzpakets lassen sich *.tdm- und *.tdms-Dateien in Microsoft Excel laden. Über eine Symbolleiste in Excel lassen sich die Eigenschaften auswählen, die für Dateien, Gruppen und Kanäle in Excel geladen werden sollen.

National Instruments, NI, ni.com und LabVIEW sind Marken der Firma National Instruments Corporation. Nähere Informationen zu den Marken von National Instruments finden Sie im Abschnitt *Terms of Use* unter ni.com/legal. Sonstige hierin erwähnte Produkt- und Firmenbezeichnungen sind Marken oder Handelsnamen der jeweiligen Unternehmen. Nähere Informationen über den Patentschutz von Produkten oder Technologien von National Instruments finden Sie unter **HilfexPatente** in Ihrer Software, in der Datei `patents.txt` auf Ihrem Datenträger oder in den *Patentinformationen von National Instruments* auf ni.com/patents. For copyright notices, conditions, and disclaimers, including information regarding certain third-party components used in LabVIEW, refer to the *Copyright* topic of the *LabVIEW Help*.