

Hinweise zum Upgrade von LabVIEW™

In dieser Broschüre wird beschrieben, wie LabVIEW unter Windows, Mac OS und Linux auf Version 8.6 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 früheren 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 Version bis einschließlich LabVIEW 7.1 gearbeitet haben, empfehlen wir Ihnen zusätzlich zur vorliegenden Broschüre die folgenden Dokumente zu lesen, in denen die wichtigsten Änderungen und Neuerungen zwischen LabVIEW 7.1 und LabVIEW 8.6 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 `rdg81v` 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 `udgup82` 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.

In der *LabVIEW-Hilfe* finden Sie auch weitere Informationen zu den Funktionen von LabVIEW 8.6, 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**.

Inhaltsverzeichnis

Upgrade auf LabVIEW 8.6	2
Konvertieren von VIs	3
Upgrade von Modulen, Toolkits und Gerätetreibern	4
Upgrade zusätzlicher Software von National Instruments	6
Upgrade älterer LabVIEW-Versionen	6
Hinweise zur Kompatibilität beim Upgrade	7
Upgrade von LabVIEW 8.5	8
Erweitern der Lizenzen für Netzwerk-Frontpanel (Windows)	10
Gemeinsame Komponenten des Application Builders	11
Upgrade von LabVIEW 8.2	11
Upgrade von LabVIEW 8.0	16
Upgrade von LabVIEW 7.x	21
Upgrade von LabVIEW 6.x	36
Upgrade von LabVIEW 5.x oder älteren Versionen	39
Änderungen und Neuerungen in LabVIEW 8.6	39
LabVIEW installieren	39
LabVIEW-Dokumentation	40
Neue Beispiel-VIs	40
Verbesserungen am Blockdiagramm	40
Verbesserungen am Frontpanel	42
Optische Verbesserungen	42
Neue und verbesserte VIs, Funktionen und Knoten	45
Neue Klasse, Eigenschaften, Methoden und Ereignisse	49
Blockdiagramm automatisch aufräumen	49
Objekte über Schnell Tasten einfügen	50
Eigenschaften mehrerer Objekte bearbeiten	50
Verbesserungen an LabVIEW-Webdiensten (Windows, Nicht im Basispaket enthalten)	50
NI-DSM	50
Verbesserungen am LabVIEW-MathScript (Nicht im Basispaket enthalten)	51
Verwalten von Überlauf für Festkommazahlen	53
LLBs vereinigen (Nicht im Basispaket)	54

Upgrade auf LabVIEW 8.6

Bei einem LabVIEW-Upgrade lesen Sie zunächst die Abschnitte *Upgrade auf LabVIEW 8.6* und *Upgrade von LabVIEW x.x* im Abschnitt *Hinweise zur Kompatibilität beim Upgrade* dieses Dokuments, wobei *x.x* für Ihre Version von LabVIEW steht.

Im Folgenden wird beschrieben, wie das Upgrade auf LabVIEW 8.6 durchzuführen ist und welche Dokumente gelesen werden sollten. National Instruments empfiehlt, dass Sie sich zunächst die *LabVIEW-Versionshinweise* und das vorliegende Dokument durchlesen.

1. Um sicherzustellen, dass Ihnen alle Kompatibilitätsprobleme bekannt sind, sollten Sie vor der Installation der neuen LabVIEW-Version folgende Abschnitte lesen:
 - **Upgrade auf LabVIEW 8.6**—Darin finden Sie Hinweise zum Deinstallieren von Toolkits und Modulen, Kopieren von Umgebungseinstellungen und `user.lib`-Dateien aus vorhergehenden LabVIEW-Versionen und zum Umwandeln von VIs für LabVIEW 8.6.
 - **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 `exqmex` heruntergeladen werden.

- (Optional) **Änderungen und Neuerungen in LabVIEW 8.6**—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 finden Sie in folgenden Abschnitten der *LabVIEW-Versionshinweise*:
 - *Systemvoraussetzungen*
 - *Installation von LabVIEW 8.6* und die entsprechenden Abschnitte für Ihre Plattform.
 - *Installation von LabVIEW-Zusatzpaketen*, wenn Sie LabVIEW-Toolkits oder Module von einer CD und nicht von 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. Lesen Sie die *LabVIEW Readme* für Hinweise zu Problemen, die in der neuen Version behoben wurden, zu bekannten Problemen in der neuen Version und zusätzlicher Dokumentation, die nicht in der *LabVIEW-Hilfe* zu finden ist. Die Datei `readme.html` finden Sie im Verzeichnis `labview\readme`.
 5. Kopieren Sie die Umgebungseinstellungen einer älteren LabVIEW-Version. Weitere Informationen zum Kopieren von Umgebungseinstellungen finden Sie im Abschnitt *Kopieren der Einstellungen einer älteren LabVIEW-Version* in diesem Dokument.
 6. Kopieren Sie die `user.lib`-Dateien einer älteren LabVIEW-Version. Im Abschnitt *Kopieren der user.lib-Dateien einer älteren LabVIEW-Version* dieses Dokuments finden Sie weitere Informationen zu `user.lib`-Dateien.
 7. Konvertieren Sie VIs für LabVIEW 8.6. Informationen zum Konvertieren von VIs, die in einer älteren Version von LabVIEW gespeichert wurden, finden Sie im Abschnitt *Konvertieren von VIs* in diesem Dokument.

Konvertieren von VIs

Wenden Sie sich an einen Vertreter von National Instruments, um VIs zu öffnen, die in einer Version vor LabVIEW 6.0 gespeichert wurden. Von diesem erhalten Sie eine Konvertierungssoftware zur Aktualisierung Ihres Programmcodes auf ein Format, das mit LabVIEW 8.6 kompatibel ist. Wenn Sie ein VI öffnen, das mit einer LabVIEW-Version ab 6.0 gespeichert wurde, wird es in LabVIEW 8.6 automatisch konvertiert 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 8.6 gespeichert werden. VIs mit ungespeicherten Änderungen (dazu zählt unter anderem auch eine Neukompilierung) können merklich langsamer sein als gespeicherte VIs.



Hinweis In LabVIEW 8.6 gespeicherte VIs lassen sich nicht in älteren Versionen öffnen. Um VIs in LabVIEW 8.5, 8.2 oder 8.0 ausführen zu können, wählen Sie im Menü **Datei»Für vorige Version speichern**. Vor dem Speichern von VIs in LabVIEW 8.6 (nach der Konvertierung) empfiehlt es sich, eine Sicherheitskopie der VIs anzulegen, die für die Verwendung in LabVIEW 8.5, 8.2 oder 8.0 vorgesehen sind.

Wenn Ihr Rechner nicht genug Speicherplatz hat, um alle VIs auf einmal zu konvertieren, gehen Sie bei der Umwandlung etappenweise vor. Werfen Sie unter Umständen auch einen Blick auf die Hierarchie der VIs, 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 alphabetischer Reihenfolge kompiliert. 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 früheren 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 8.6 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.

NI-Module und Toolkits

In der folgenden Tabelle finden Sie Informationen zu für die Installation zu verwendenden Datenträgern.

Betriebssystem	Datenträger	Wichtige Hinweise
Windows	DVD	Installieren Sie LabVIEW 8.6 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 8.6 installiert, ohne dass Sie vorige Versionen deinstallieren oder modifizieren müssen. Informationen zur Installation von LabVIEW, Modulen und Toolkits finden Sie in den <i>LabVIEW-Versionshinweisen</i> .

Betriebssystem	Datenträger	Wichtige Hinweise
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 Module und Toolkits im Verzeichnis von LabVIEW 8.6. Massenkompilieren Sie alle VIs, die in Vorgängerversionen von LabVIEW gespeichert wurden. Weitere Informationen dazu finden Sie im Abschnitt Massenkompilierung in LabVIEW .
Mac/Linux		



Hinweis Einige Versionen von Toolkits können nicht mit LabVIEW 8.6 verwendet werden. Wenn Sie ein inkompatibles Toolkit installieren, funktionieren einige Funktionen des Toolkits oder von LabVIEW nicht ordnungsgemäß. 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 Installation von LabVIEW 8.6 durch ein inkompatibles Toolkit Fehler aufweist, deinstallieren Sie mithilfe der Funktion **Programme ändern oder entfernen** zunächst das Toolkit und reparieren Sie dann die LabVIEW-Installation.

Gerätetreiber

Für die Steuerung von und Kommunikation mit jeglicher Hardware sind aktuelle Gerätetreiber erforderlich. Wenn ein Gerätetreiber mit einer früheren Version von LabVIEW installiert wurde, muss dieser mit LabVIEW 8.6 erneut installiert werden. Folgende Methoden stehen dafür zur Auswahl:

- **Modulare NI-Gerätetreiber**—Installieren Sie modulare NI-Gerätetreiber mithilfe der NI-Gerätetreiber-DVD oder -CD.
- **Plug-and-Play-Gerätetreiber**—Mit der Suchmaschine für NI-Gerätetreiber können Sie nach LabVIEW-Treibern für Plug-and-Play-Geräte suchen und diese installieren, ohne LabVIEW verlassen zu müssen.
- **IVI-Treiber oder nicht geprüfte Gerätetreiber**—Suchen Sie im [Netzwerk für Gerätetreiber](#) auf der Website von National Instruments nach IVI- oder nicht geprüften Gerätetreibern.



Hinweis Wenn Sie Treiber mit Hilfe der Suchmaschine für NI-Gerätetreiber neu installieren, empfiehlt National Instruments eine Massenkompilierung des Verzeichnisses `labview\instr.lib`.

Zusatzpakete von Drittanbietern

Wenden Sie sich an den Anbieter des Zusatzpakets für LabVIEW, um festzustellen, ob das Produkt mit LabVIEW 8.6 und Ihrem Betriebssystem kompatibel ist. 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

VI, 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. Wenn Sie LabVIEW-Module und Toolkits installieren, die nicht auf den LabVIEW-Plattform-DVDs enthalten sind oder von Zusatzpaketen von Drittanbietern stammen, wird empfohlen, für alle VIs eine Massenkompilierung durchzuführen, die damit installiert werden.

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

Upgrade zusätzlicher Software von National Instruments

Für LabVIEW 8.x wird NI TestStand ab 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 NI TestStand 4.0 geben einen Fehler aus, wenn Sie versuchen, die folgenden Express-VIs aus LabVIEW 8.6 zu konfigurieren:

- “Datenspeicher schließen”
- “Formel”
- “Eigenschaften lesen”
- “Datenspeicher öffnen”
- “Daten lesen”
- “Eigenschaften festlegen”
- “Spektrum ermitteln”
- “Daten schreiben”

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 `Readme.html` Ihrer aktuellen NI TestStand-Version auf der NI TestStand-CD oder im Verzeichnis `<TestStand>\Doc`.

Für LabVIEW 8.x wird NI Spy ab Version 2.3 benötigt. NI Spy 2.5 befindet sich auf der Treiber-CD von National Instruments.

LabVIEW 8.6 unterstützt Measurement Studio ab Version 8.0. Wie Measurement Studio 8.0 oder später bestellt werden kann, erfahren Sie auf unserer Website ni.com/info nach Eingabe des Infocodes `exd8yy`.

Upgrade älterer LabVIEW-Versionen

LabVIEW 8.6 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.x` gespeichert (`x.x` steht für die Version).

Ersetzen einer bestehenden LabVIEW-Version

Deinstallieren Sie die bestehende Version von LabVIEW, starten Sie das Installationsprogramm von LabVIEW 8.6 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 8.6 ersetzt werden. Dateien, die im `labview`-Verzeichnis erstellt wurden, bleiben vom Deinstallationsprogramm unberührt.



Hinweis Bei der Deinstallation oder Neuinstallation von LabVIEW werden alle `*.lib`-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 8.6.

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

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

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

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 8.6 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.

Weitere Informationen zu bekannten Problemen der neuen LabVIEW-Version, zusätzlichen Kompatibilitätsproblemen und Informationen zu zuletzt hinzugefügten Funktionen in LabVIEW 8.6 finden Sie in der Datei `readme.html` im `labview`-Verzeichnis.

Upgrade von LabVIEW 8.5

Beim Upgrade von LabVIEW 8.5 auf LabVIEW 8.6 können folgende Kompatibilitätsprobleme auftreten:



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 `upnote85`.

Unterstützte Plattformen

Im Hinblick auf die Plattformkompatibilität gibt es bei LabVIEW 8.6 folgende Änderungen:

- LabVIEW 8.6 unterstützt keine Macintosh-Computer mit PowerPC-Prozessoren.

Systemvoraussetzungen

(Windows) Für LabVIEW 8.6 ist ein Speicherplatz von mindestens 1,6 GB erforderlich.

(Mac OS) Für die Minimalversion von LabVIEW 8.6 sind mindestens 262 MB Festplattenspeicher und für die vollständige Installation 828 MB erforderlich.

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

Gedruckte Dokumentation

Die folgenden Dokumente haben sich in LabVIEW 8.6 nicht geändert. Daher sind für LabVIEW 8.6 relevante Informationen möglicherweise nicht darin enthalten.

- Handbuch *Erste Schritte mit LabVIEW*
- **(LabVIEW 8.2, 8.5 und 8.6)** *LabVIEW-Grundlagen*



Hinweis Da das Handbuch *LabVIEW-Grundlagen* Teil des Buchs **Grundlagen** der *LabVIEW-Hilfe* ist, finden Sie aktuelle Informationen im Buch **Grundlagen** auf der Registerkarte **Inhalt** der *LabVIEW-Hilfe*.

Änderungen im Verhalten von VIs und Funktionen

Das Verhalten der nachfolgenden VIs und Funktionen hat sich in LabVIEW 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, Anzeige- und Bedienelemente 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.

Externer Code (DLLs und CINS)

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

Verschiedene Änderungen im Verhalten von VIs und Funktionen

In LabVIEW 8.6 gibt es folgende Änderungen an VIs und 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`.
- LabVIEW setzt Einzelprozess-Umgebungsvariablen automatisch auf systemvariabel. 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.

Veraltete VIs und Funktionen

Folgende VIs und Funktionen werden von LabVIEW 8.6 nicht unterstützt:

- **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 in LabVIEW nicht automatisch umgewandelt werden zur Verwendung des VIs “Semaphor-Referenz anfordern”, müssen VIs, die in einer früheren Version von LabVIEW gespeichert wurden, manuell aktualisiert werden.
- **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 in LabVIEW nicht automatisch umgewandelt werden zur Verwendung des VIs “Semaphor-Referenz freigeben”, müssen VIs, die in einer früheren 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 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**.

Veraltete Eigenschaften, Methoden und Ereignisse

Die folgenden Eigenschaften, Methoden und Ereignisse werden von LabVIEW-Versionen ab 8.6 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” des Klasse “Eigenschaften für VI (ActiveX)”. Verwenden Sie stattdessen die Eigenschaft “VIModificationBitSet2”.
- Die Eigenschaft “Modifikationen:VI-Modifikationen” der Klasse “VI-Eigenschaften”. 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 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
VI	Elementwert:Alle ermitteln [Variant]	Elementwert:Alle ermitteln	Methode

Erweitern der Lizenzen für Netzwerk-Frontpanel (Windows)

Clients können eine Applikation oder ein Frontpanel mit Hilfe von LabVIEW auf einem Netzwerkrechner sehen. LabVIEW unterstützt Lizenzen für 5, 20, 50 oder eine unbegrenzte Anzahl von 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 die Lizenz für Netzwerk-Frontpanel für LabVIEW 8.5.1 oder frühere Versionen 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 von LabVIEW 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 8.6 angewiesen ist. Zum Beheben dieses Problems installieren Sie LabVIEW 8.6 erneut.

Upgrade von LabVIEW 8.2

Beim Upgrade von LabVIEW 8.2 auf LabVIEW 8.6 können nachfolgend beschriebene Kompatibilitätsprobleme auftreten. Sonstige potentielle Probleme beim Upgrade von LabVIEW sind im Abschnitt *Upgrade von LabVIEW 8.5* beschrieben.



Hinweis Zusätzliche Informationen zu weiteren Problemen, die beim Upgrade von LabVIEW 8.2 auftreten können, erhalten Sie auf der Website von National Instruments ni.com/info nach Eingabe des Infocodes `ex5d8c`.

Unterstützte Plattformen

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

- LabVIEW-Versionen ab 8.5 unterstützen Windows Vista x86 und x64.
- LabVIEW-Versionen ab 8.5 unterstützen Macintosh-Rechner mit Prozessoren von Intel und PowerPC.

Systemvoraussetzungen

(Windows) Für LabVIEW 8.5 ist ein Speicherplatz von mindestens 1,2 GB erforderlich. Für LabVIEW 8.6 ist ein Speicherplatz 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. LabVIEW 8.6 benötigt für die Minimalversion einen Festplattenspeicherplatz von mindestens 262 MB und für die vollständige Installation 828 MB.

(Linux) Für die Minimalversion von LabVIEW 8.5 sind mindestens 450 MB Festplattenspeicher und für die vollständige Installation 640 MB erforderlich. LabVIEW 8.6 benötigt für die Minimalversion einen Festplattenspeicherplatz von mindestens 365 MB und für die vollständige Installation 651 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 x86)** Die VI-Komponenten werden zwar ordnungsgemäß installiert, erscheinen aber im Windows Defender als nicht signiert. Die VIs werden ordnungsgemäß ausgeführt.
- **(Windows Vista x64)** Diese VIs geben den “Fehler -4850” aus.

Änderungen im Verhalten von VIs und Funktionen

Das Verhalten 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 früheren 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 `exigr`.

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 hyperbolische Funktionen haben sich in LabVIEW-Versionen ab 8.5 geändert:

- Die Funktion “Areacoscinus Hyperbolicus” gibt NaN aus, wenn der Eingangswert eine reelle Zahl ist, die sich außerhalb des Bereichs der Funktion befindet.
- Die Funktion “Areascans” gibt NaN aus, wenn der Eingangswert eine reelle Zahl ist, die sich außerhalb des Bereichs der Funktion befindet.

VIs 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.

Funktion “VI-Referenz öffnen”

Wenn der Name des VIs vom Eingang **VI-Pfad** in LabVIEW 8.0 und 8.2 mit dem Namen eines VIs im Speicher auf dem Zielsystem übereinstimmt, gibt LabVIEW eine Referenz auf das VI im Speicher aus und LabVIEW lädt das VI nicht, das am Eingang **VI-Pfad** festgelegt wurde. Wenn der Name des VIs vom Eingang **VI-Pfad** in LabVIEW-Versionen ab 8.5 mit dem Namen eines VIs im Speicher auf diesem Zielsystem übereinstimmt, die Pfade aber unterschiedlich sind, gibt die Funktion “VI-Referenz öffnen” einen Fehler aus.

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.

Verschiedene Änderungen im Verhalten von VIs und Funktionen

In LabVIEW-Versionen ab 8.5 gibt es folgende Änderungen an VIs und Funktionen:

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

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: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 “Signalverlaufgraph”. 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 “Szenobjekt”. Verwenden Sie stattdessen die Methode “Als darstellbar 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, Numerisch	Aktion für Werte außerhalb des Bereichs	Antwort auf Wert außerhalb der Grenzen	Eigenschaft
Element für absolute Zeit, Numerisch	Bei Bereichsüberschreitung: Inkrement	Antwort auf Wert außerhalb der Grenzen: Inkrement	Eigenschaft
Element für absolute Zeit, Numerisch	Bei Bereichsüberschreitung: Maximum	Antwort auf Wert außerhalb der Grenzen: Maximum	Eigenschaft
Element für absolute Zeit, Numerisch	Bei Bereichsüberschreitung: Minimum	Antwort auf Wert 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

Klasse	Name in LabVIEW 8.2	Name in LabVIEW 8.5	Typ
Applikation	Szene:Geometrie:Neue Kugel	Szene:Darstellbares Objekt:Geometrie:Neue Kugel	Methode
Application (ActiveX)	LibraryGetProjectLibFileVersion	LibraryGetFileLVVersion	Methode
Digital, Numerischer-Text und Skalierung	Format und Genauigkeit	Anzeigeformat	Eigenschaft
Digital, Numerischer-Text und Skalierung	Format und Genauigkeit:Format	Anzeigeformat:Format	Eigenschaft
Digital, Numerischer-Text 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	Als darstellbar festlegen	Methode
VI	Anschlussfeld	Anschlussfeld:Festlegen	Eigenschaft

Änderungen im Verhalten 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`
 - `Pfad`
 - `rmpath`

LabVIEW stellt nach Schließen des **LabVIEW-MathScript-Fensters** oder nach Beendigung des VIs mit dem MathScript-Knoten die Standardliste und das Standardarbeitsverzeichnis wieder her.
- 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 Klassenelements in ein Blockdiagramm angezeigt werden soll, muss das Klassensymbol verkleinert werden, sodass die Klassenmaske das Symbol nicht verdeckt. Das Bild sollte nicht größer sein als 32 Pixel breit und 19 Pixel hoch.

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 Fenster **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.

Upgrade von LabVIEW 8.0

Beim Upgrade von LabVIEW 8.0 auf LabVIEW 8.6 können nachfolgend beschriebene Kompatibilitätsprobleme auftreten. Alle weiteren potentiellen Probleme beim Upgrade von LabVIEW sind in den Abschnitten [Upgrade von LabVIEW 8.2](#) und [Upgrade von LabVIEW 8.5](#) beschrieben.

Weitere Informationen zu neuen Eigenschaften und Änderungen in jeder Version sind in den *Hinweisen zum Upgrade von LabVIEW* der einzelnen LabVIEW-Versionen von 8.0 bis 8.5 beschrieben, die Sie unter ni.com/manuals finden.

Unterstützte Plattformen

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

- Versionen ab LabVIEW 8.2 arbeiten nicht mit Windows XP x64.
- Versionen ab LabVIEW 8.2 arbeiten weder mit Mac OS X 10.3.8 noch mit Vorgängerversionen davon.
- LabVIEW 8.2 arbeitet mit Macintosh-Rechnern mit Prozessoren von Intel. Weitere Informationen zur Unterstützung von Macintosh-Rechnern finden Sie auf der Website von National Instruments ni.com/info nach Eingabe des Infocodes `macintel`. LabVIEW 8.5 unterstützt Macintosh-Rechner mit Intel- und PowerPC-Prozessoren.

Systemvoraussetzungen

(Windows) Für LabVIEW 8.2 ist ein Speicherplatz von mindestens 1,2 GB erforderlich. Für LabVIEW 8.6 ist ein Speicherplatz 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. LabVIEW 8.6 benötigt für die Minimalversion einen Festplattenspeicherplatz von mindestens 262 MB und für die vollständige Installation 828 MB.

(Linux) Für die Minimalversion von LabVIEW 8.2 sind mindestens 430 MB Festplattenspeicher und für die vollständige Installation 620 MB erforderlich. LabVIEW 8.6 benötigt für die Minimalversion einen Festplattenspeicherplatz von mindestens 365 MB und für die vollständige Installation 651 MB.

Gedruckte Dokumentation

Die folgenden Dokumente haben sich in LabVIEW 8.2 nicht geändert. Daher sind für LabVIEW 8.2 relevante Informationen möglicherweise nicht darin enthalten.

- *LabVIEW-Schnellübersicht*
- *LabVIEW-Grundlagen*—Da die *LabVIEW-Grundlagen* Teil des Buchs **Grundlagen** in der *LabVIEW-Hilfe* sind, finden Sie aktuelle Informationen im Buch **Grundlagen** auf der Registerkarte **Inhalt** der *LabVIEW-Hilfe*.

Änderungen im Verhalten von VIs und Funktionen

Das Verhalten 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 erzeugen”, “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 spätere 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 unabhängig von Eingangsfehlern.

Histogramm-VI

Der Standardwert des Eingangs **Intervalle** des VIs “Histogramm” wurde auf 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 Audio-Task 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 Triggererkennung**—In beiden Instanzen dieses VIs wurde der Eingang **Steigung in Triggerflanke** 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)”. Am Eingang “Format von Start/Dauer” gibt es die Option **Absolute Zeit** nicht mehr. Der Eingang “Start” wurde in “Start-Samples/Zeit” und der Ausgang “Tatsächlicher Start” in “Tatsächliche Start-Samples/Zeit” geändert.
- **Zeit-Array für Signalverlauf lesen**—Der Datentyp des Ausgangs “X-Array” wurde von “numerischer Fließkommawert doppelter Genauigkeit” in “Zeitstempel” geändert.
- **Y-Wert abfragen**—Dieses polymorphe VI und dessen Instanzen wurden 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 Start” 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 Näherung” und “Zeit der Näherung” wurde von numerischem Fließkommawert doppelter Genauigkeit in Zeitstempel geändert.
- **Signalverlauf - Minimum und Maximum**—Der Datentyp der Ausgänge “Minimum-Zeit” und “Maximum-Zeit” wurde von numerischem Fließkommawert doppelter Genauigkeit in Zeitstempel geändert.
- **Signalverlauf zu XY-Paaren**—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.

- Die Voreinstellung für den Eingang **Optionen** der ActiveX-Methode “GetVIReference” ist jetzt, den Benutzer zum Suchen nach fehlenden SubVIs aufzufordern. Es gibt jedoch eine neue Einstellung, 0x20, 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.
- .NET-Methoden in LabVIEW 8.0, die Array-Datentypen anhand von Referenzen verarbeiten, leiten alle Daten als Refnum-Datentyp weiter. .NET-Methoden in Versionen ab LabVIEW 8.2, die Array-Datentypen mittels Referenzen verarbeiten, leiten alle Daten mit ihrem individuellen Datentyp weiter.
- Die Eigenschaft “Position bearbeiten” der Klassen “Digitalwerttabelle”, “mehrspalzigesListefeld”, “Tabelle” und “Baumstruktur” gibt die Werte (-2, -2) aus. Das bedeutet, dass Benutzer keine Änderungen am Text des Elements vornehmen. Die Eigenschaft “Zeile bearbeiten” der Klasse “Listefeld” 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 Versionen ab LabVIEW 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:

- Versionen ab LabVIEW 8.2 unterstützen die Eigenschaft “Anschlussfeld” 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	Bibliotheks-Tag löschen	Bibliotheks-Tag:Löschen	Methode
Bibliothek	Symbol ermitteln	Symbol:Ermitteln	Methode
Bibliothek	Bibliotheks-Tag ermitteln	Bibliotheks-Tag:Ermitteln	Methode
Bibliothek	Bibliotheks-Tagnamen 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:Setzen	Methode
Bibliothek	Bibliotheks-Tag festlegen	Bibliotheks-Tag:Setzen	Methode
Bibliothek	Sperrzustand festlegen	Sperrzustand:Festlegen	Methode
Bibliothek	Quellbereich festlegen	Quellbereich:Festlegen	Methode

Klasse	Name in LabVIEW 8.0	Name in LabVIEW-Versionen ab 8.2	Typ
Listenfeld, mehrspaltigesListenfeld und Baumstruktur	Ziehen/Ablegen:Verschieben von Objekten möglich	Ziehen/Ablegen:Ziehen zulassen	Eigenschaft
Projektelement	Tag löschen	Tag:Löschen	Eigenschaft
Projektelement	Tag lesen	Tag:Tag ermitteln	Eigenschaft
Projektelement	Tag-Namen ermitteln	Tag:Namen ermitteln	Eigenschaft
Projektelement	XML-Tag lesen	Tag:XML-Tag ermitteln	Eigenschaft
Projektelement	Tag festlegen	Tag:Tag setzen	Eigenschaft
Projektelement	XML-Tag festlegen	Tag:XML-Tag festlegen	Eigenschaft
Projektelement	Bibliothekselementtyp-String	Bibliothekselementtyp:String	Eigenschaft
Projektelement	Bibliothekselementtyp	Bibliothekselement: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 `.11b` 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 8.6 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](#) und [Upgrade von LabVIEW 8.5](#) beschrieben.

Alle Unterschiede zwischen den einzelnen Versionen ab 7.x sind in den *Hinweisen zum Upgrade von LabVIEW* jeder Version beschrieben, die Sie unter ni.com/manuals finden.



Hinweis Das Handbuch *LabVIEW-Grundlagen* hat sich für LabVIEW 8.6 nicht geändert. Die PDF-Versionen dieser Dokumente finden Sie im Verzeichnis `labview\manuals`. Da das Handbuch *LabVIEW-Grundlagen* Teil des Buchs **Grundlagen** der *LabVIEW-Hilfe* ist, finden Sie aktuelle Informationen im Buch **Grundlagen** auf der Registerkarte **Inhalt** der *LabVIEW-Hilfe*.

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 wird ein Arbeitsspeicher von mindestens 128 MB benötigt. Empfohlen werden jedoch 256 MB. Für LabVIEW 8.5 ist ein Arbeitsspeicher von mindestens 256 MB erforderlich (empfohlen wird 1 GB).

Für LabVIEW 7.x ist eine Bildschirmauflösung von 800 × 600 Pixeln erforderlich. Empfohlen werden jedoch 1.024 × 768 Pixel. Für LabVIEW 8.x 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. Die Mindestanforderung für LabVIEW 8.x ist ein Pentium III, ein Celeron-Prozessor mit 866 MHz oder ein gleichwertiges Produkt. Es wird jedoch ein Pentium 4/M oder ein gleichwertiger Prozessor empfohlen.

LabVIEW 7.x benötigt für die Minimalversion einen Festplattenspeicherplatz von mindestens 130 MB und für die vollständige Installation 550 MB. LabVIEW 8.6 benötigt für eine vollständige Installation 1,6 GB Festplattenspeicherplatz.

Mac OS

LabVIEW 7.x benötigt für die Minimalversion einen Festplattenspeicherplatz von mindestens 280 MB und für die vollständige Installation 350 MB. LabVIEW 8.6 benötigt für die Minimalversion einen Festplattenspeicherplatz von mindestens 262 MB und für die vollständige Installation 828 MB.

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. Die Mindestanforderung für LabVIEW 8.x ist ein Pentium III, ein Celeron-Prozessor mit 866 MHz oder ein gleichwertiges Produkt. Es wird jedoch ein Pentium 4/M oder ein gleichwertiger Prozessor empfohlen.

LabVIEW 7.x benötigt für die Minimalversion einen Festplattenspeicherplatz von mindestens 200 MB und für die vollständige Installation 300 MB. LabVIEW 8.6 benötigt für die Minimalversion einen Festplattenspeicherplatz von mindestens 365 MB und für die vollständige Installation 651 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 8.x ist die GNU C Library ab Version 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 8.x arbeitet nur mit Red Hat Enterprise Linux WS ab 3, Mandrake-Linux/Mandriva ab 10.0 oder SuSE Linux ab 9.1.

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 im Verhalten von VIs und Funktionen

Das Verhalten der nachfolgenden VIs und Funktionen hat sich in LabVIEW 7.1 oder 8.0 geändert.

.NET-VIs und -Applikationen

Für die .NET-Funktionen und -Anwendungen 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 Assembly 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 Versionen ab LabVIEW 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 gleich dem einer Datei im Speicher des Zielsystems ist.

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 **max|X|** als **0** und **Y[i]=X[i]/Max|X|** oder **Yij=Xij/Max|X|** als 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 **Y[i]=X[i]/Max|X|** bzw. **Yij=Xij/Max|X|** als 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” bzw. “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)/scale** 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 aktualisiert, um den Funktionsumfang von LabVIEW 7.x zu erhalten.

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 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.

VI “Einheitsvektor”

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

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

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

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

wobei X der Vektor (Eingang), $\|X\|$ die Norm und y 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 Anwendung 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 für 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:

- **(Mac OS)** Die PPC-VIs werden bei LabVIEW 7.1 und später 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” bzw. “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” bzw. 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 **Anstiegsgeschwindigkeit**, **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)” bzw. “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-Ressourcename” und “Logischer IVI-Name” nicht mehr auf der Palette **Funktionen**. Der VISA-Ressourcename 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 Programmtypen ermitteln” bzw. “Datei- und Programmtypen 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 das Verhalten 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 sie sich 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 das Verhalten wieder zurück geändert auf das Verhalten von LabVIEW 7.1.

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, passen Sie Ihre VIs der neuen Funktionsweise entsprechend an.

Eigenschaften für Listefeld

Wenn Sie in LabVIEW 7.x oder Vorgängerversionen die Eigenschaft “Erste Zeile” in einem Listefeld auf eine Zeile einstellen, die sich unter dem untersten Element im Listefeld 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 Listfelder 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 Referenz, die von der Eigenschaft “Übergeordnetes VI” ausgegeben wird, das VI nicht im Speicher. Wenn das übergeordnete VI nicht mehr im Arbeitsspeicher ist, wird die Referenz ungültig. Eine Referenz für ein VI, mit der das VI so lange im Speicher verbleibt, bis die Referenz per Hand geschlossen wird, kann mit der Funktion “VI-Referenz öffnen” erzeugt werden.

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.

Eigenschaften für Baumstruktur-Element

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 aus früheren LabVIEW-Versionen mit Hilfe der Methode “VI-Strings exportieren” exportieren, 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 Listenfelder, 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 numerischesElementMitText

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 Frontpanel

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 "Panel skalieren" nicht mehr unterstützt. Verwenden Sie stattdessen die Methode "Skalierungsmodus einstellen" 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 setzen” 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 rdlv21 ein.

Mit den **Build-Spezifikationen** im **Projekt-Explorer** können Sie Build-Spezifikationen für Applikationen (EXE-Dateien), DLLs und Zip-Dateien und diese selbst erstellen. (**Windows**) Mit den **Build-Spezifikationen** können auch Spezifikationen für Installer sowie diese selbst erstellt werden. 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.

Sie können eine * .bld-Datei in einem neuen Projekt in eine Build-Spezifikation umwandeln. 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 Anwendungsmenüpunkte

In der folgenden Tabelle finden Sie Tags für Standardmenüpunkte, die nicht mehr in LabVIEW verwendet werden, weil der Punkt nicht mehr verfügbar ist oder mit einem anderen Punkt verknüpft 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_IIST
- 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 Standardmenüpunkte 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.

Fenster “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 Integern. 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 Listenfeldern oder mehrspaltigen Listenfeldern.

In LabVIEW 8.x enthält jede Zeile einer Exportdatei nicht mehr als zwei Tags für Privat- oder Standarddaten. Des Weiteren werden in LabVIEW 8.x werden die Elemente auch auf jeder Unterebene 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 bzw. 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 zukünftigen Versionen nicht mehr unterstützt.

Umstellung vom integrierten LabVIEW-Versionsverwaltungs-Provider

Den Versionsverwaltungs-Provider 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 Versionsverwaltungs-Provider von einem Drittanbieter auswählen. Wenn Sie bisher den integrierten Provider verwendet haben, müssen Sie die Dateien auf einen anderen Versionsverwaltungs-Provider in LabVIEW umstellen. Die aktuelle Liste der Versionsverwaltungs-Provider 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 Versionsverwaltungs-Provider 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 Versionsverwaltungs-Provider auf einen Versionsverwaltungs-Provider von einem Drittanbieter zu übertragen:

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

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 integrierten Versionsverwaltungs-Provider 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 bzw. 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 Bearbeitung von Meldungen des Betriebssystems verzögert, bis die Aufrufe der DLL-Funktionen beendet sind bzw. bis Sie ein modales Dialogfeld von der DLL aus geladen haben. Durch diese Verzögerung werden DLL-Funktionen ohne Unterbrechung ausgeführt. LabVIEW ruft auch nicht die gleiche DLL auf, während eine DLL-Funktion ausgeführt wird, was zu einem Sperrstatus 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 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 7.x und frühere Versionen bieten undokumentierte Eigenschaften, mit denen Macintosh-Ressourcendateien gelesen und geschrieben werden können. Diese Methoden gibt es in LabVIEW 8.x jedoch 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 `labvieww.lib`. Weitere Informationen zum Verknüpfen von DLLs mit `labvieww.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 8.6 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* und *Upgrade von LabVIEW 8.5* beschrieben.

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 späteren Versionen wandelt das VI “Signalverlauf aus Datei lesen” das alte Signalverlaufsdattentyp-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 und spätere Versionen zu beachten gibt.

VIs für serielle Kompatibilität

In LabVIEW 7.x und späteren 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 7.x und später 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 Anwendungen zur Steuerung von Geräten mit serieller Schnittstelle die VIs und Funktionen der VISA-Palette. Alle VIs, die in Vorgängerversionen von LabVIEW enthalten waren und mit seriellen VIs gearbeitet haben, sind auch in LabVIEW 7.1 und späteren Versionen funktionstüchtig.

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 Eingangsarray verwendet werden. Jedes Element im Array sollte einen Port darstellen. Wenn Sie beispielsweise Port 0 das VISA-Alias “MySerialPort” zugewiesen haben, geben Sie in das Array **VISA Aliases** als erstes Element `MySerialPort` ein. Vor dem VI “Set Serial Alias Ports” muss immer 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\smplser1.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 Tunneltyp ausgegeben.

Lizenzen für netzwerkgesteuerte 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*.

Gerätetreiber

Bei LabVIEW-Versionen ab LabVIEW 7.x ist die LabVIEW-Instrumententreiber-CD nicht im Lieferumfang enthalten. Im Instrument Driver Network 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 späteren 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 und späteren Versionen:

Einheit	Definition (6.1)	7.x und neuer (Definition)
Astronomische Einheit (AU)	149.498.845.000 m	149.597.900.000 m
British Thermal Unit (Mittelwert)	1055,79 J	1055,87 J

Einheit	Definition (6.1)	7.x und neuer (Definition)
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.
Gallone (imperial)	4,54596 l	4,54609 l
Lichtjahr	9,4605 Pm	9,46073 Pm
Pound Force	4,448 N	4,448222 N
Rod	502,92 cm	5,029210 m
Slug	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 Elemente, die sich inzwischen geändert haben, erst wieder aktualisiert, wenn der Wert FALSE an die Eigenschaft übermittelt wird. Wenn diese Eigenschaft in LabVIEW ab Version 7.0 den Wert TRUE hat, werden alle Frontpanel-Objekte mit noch ausstehenden Änderungen neu angezeigt 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 einen Mindestwert von 0,00, einen Höchstwert von 0,00, einen Inkrement-Wert von 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 den LabVIEW-Versionen ab 6.1 sind in den Verbindungen Angaben über Typdefinitionen enthalten, so dass im Blockdiagramm mehr Typumwandlungspunkte vorliegen können. Wenn Sie ein Typ-Definitionselement mit einem VI oder einem Knoten verbinden, der kein Typ-Definitionsknoten ist, zeigt LabVIEW einen Typumwandlungspunkt an. Typumwandlungspunkte werden auch angezeigt, wenn ein Ausgangsanschluss, der eine Typdefinition hat, mit einer Anzeige verbunden wird, die keine Typdefinition hat. 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 `^` verwendet. In den LabVIEW-Versionen ab 6.1 ist eine Potenzierung mit dem Symbol `**` gekennzeichnet (zum Beispiel `x**y`). Das Symbol `^` 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

Informationen zum Upgrade von LabVIEW 5.x und Vorgängerversionen auf LabVIEW 8.6 finden Sie auf der Website von National Instruments ni.com/info nach Eingabe des Infocodes `ext8h9`. Auf dieser Website finden Sie Hinweise zu älteren Versionen der *Hinweise zum Upgrade von LabVIEW*, die Sie lesen sollten, und zum Umwandeln von VIs, sodass diese in LabVIEW 8.6 funktionieren.

Änderungen und Neuerungen in LabVIEW 8.6

In der *LabVIEW-Hilfe* finden Sie Informationen zu Funktionen in LabVIEW 8.6, 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 Bugs, zusätzlichen Kompatibilitätsproblemen und Informationen zu zuletzt hinzugefügten Funktionen in LabVIEW 8.6 finden Sie in der Datei `readme.html` im `labview`-Verzeichnis.

LabVIEW installieren

(Windows) Mit LabVIEW 8.6 können Sie LabVIEW installieren und Module und Toolkits von den LabVIEW-Plattform-DVDs auswählen. Weitere Informationen finden Sie im Abschnitt [Upgrade von Modulen, Toolkits und Gerätetreibern](#) in diesem Dokument oder im Abschnitt [Installieren von LabVIEW 8.6](#) in den *LabVIEW-Versionshinweisen*.

LabVIEW-Dokumentation

Die *LabVIEW-Schnellübersicht* wurde überarbeitet und enthält Tastenkombinationen, Online-Ressourcen und eine Anleitung zur Verwendung der Dokumentation.

Neue Beispiel-VIs

Im Ordner **Neue Beispiele für LabVIEW 8.x** auf der Registerkarte **Suchen** der NI-Beispielsuchmaschine finden Sie neue Beispiel-VIs und Beschreibungen dazu. Die Beispiele können direkt gestartet werden.

Verbesserungen am Blockdiagramm

In LabVIEW 8.6 gibt es folgende Verbesserungen am Blockdiagramm und dazugehörigen Funktionen:

Erstellen und Bearbeiten von Express-VIs

Wählen Sie **Werkzeuge»Fortgeschritten»Express-VI erstellen oder bearbeiten** zum Erstellen und Bearbeiten eines Express-VIs im Dialogfeld **Express-VI erstellen oder bearbeiten**. Sie können ein Express-VI von einem vorhandenen VI, von einem anderen Express-VI oder von einem leeren VI erstellen. In älteren LabVIEW-Versionen ist diese Funktion nur verfügbar, wenn Sie das LabVIEW Express VI Development Toolkit installiert haben.

Weitere Informationen zum Erstellen von Express-VIs in LabVIEW finden Sie im Buch **Grundlagen»Erstellen des Blockdiagramms»Allgemeines»Express-VIs** auf der Registerkarte **Inhalt** in der *LabVIEW-Hilfe*.

Verwalten von Haltepunkten mit dem Fenster “Haltepunkt-Manager”

Sie können zum Erstellen eines Haltepunktes ein Objekt auf dem Blockdiagramm mit der rechten Maustaste anklicken und **Haltepunkt»Haltepunkt setzen** aus dem Kontextmenü auswählen. Zum Deaktivieren eines Haltepunkts und Verhindern, dass ein VI am Haltepunkt anhält, ohne den Haltepunkt zu entfernen, klicken Sie mit der rechten Maustaste auf ein Objekt im Blockdiagramm und wählen Sie **Haltepunkt»Haltepunkt deaktivieren** aus dem Kontextmenü. Sie können einen deaktivierten Haltepunkt aktivieren, indem Sie ein Objekt mit einem deaktivierten Haltepunkt mit der rechten Maustaste anklicken und **Haltepunkt»Haltepunkt aktivieren** auswählen. Klicken Sie zum Entfernen eines Haltepunkts mit der rechten Maustaste auf ein Objekt mit Haltepunkt und wählen Sie **Haltepunkt»Haltepunkt entfernen** aus dem Kontextmenü.

Mit dem Fenster **Haltepunkt-Manager** lassen sich Haltepunkte in einem VI verwalten. In diesem Fenster lassen sich alle oder nur ausgewählte Haltepunkte in der VI-Hierarchie deaktivieren, aktivieren und löschen. Zum Anzeigen, wo sich ein bestimmter Haltepunkt im Blockdiagramm befindet, klicken Sie doppelt auf einen Haltepunkt im **Haltepunkt-Manager**.

Wählen Sie zur Anzeige des **Haltepunkt-Managers** den Menüpunkt **Ansicht»Haltepunkt-Manager** oder klicken Sie mit der rechten Maustaste auf ein Objekt und wählen Sie **Haltepunkt»Haltepunkt-Manager** aus dem Kontextmenü. Wählen Sie bestimmte Haltepunkte aus oder klicken Sie zur Auswahl aller Haltepunkte auf die Schaltfläche **Alle auswählen**. Klicken Sie zum gleichzeitigen Aktivieren oder Deaktivieren aller ausgewählten Haltepunkte auf **Aktivieren** oder **Deaktivieren**.

Vorbelegung von gemeinsam genutzten Kopien von ablaufinvarianten VIs in zeitgesteuerten Strukturen

Sie können LabVIEW so konfigurieren, dass Kopien eines ablaufinvarianten VIs in einer zeitgesteuerten Schleife oder Sequenz entweder vorbelegt oder gemeinsam genutzt werden. Sie können z. B. beim Einfügen eines ablaufinvarianten VIs mit gemeinsam genutzten Kopien in eine zeitgesteuerte Schleife oder Sequenz diese so einstellen, dass Kopien für Instanzen des ablaufinvarianten VIs, die in der Struktur aufgerufen werden, vorbelegt werden. Instanzen von ablaufinvarianten VIs, die außerhalb der Struktur

aufgerufen werden, verwenden Kopien weiterhin gemeinsam. LabVIEW belegt VI-Kopien für zeitgesteuerte Strukturen per Voreinstellung vor. Zum Verringern der Speicherauslastung bei gemeinsam genutzten Kopien von ablaufinvarianten VIs konfigurieren Sie die zeitgesteuerte Struktur, so dass VI-Kopien nach Bedarf zugewiesen werden. Zum Festlegen der Zuweisung eines VIs in einer zeitgesteuerten Schleife oder Struktur klicken Sie mit der rechten Maustaste auf die Struktur und wählen Sie **Zuweisung gemeinsam genutzter Kopien** aus dem Kontextmenü und dann eine der folgenden Optionen aus:

- **Automatisch**—LabVIEW entscheidet je nach Kontext der Struktur, ob die Einstellung **Vorbelegen** oder **Nach Bedarf zuweisen** lautet. LabVIEW hängt die Kontextmenüoption an, die für die ausgewählte Einstellung steht. Wenn z. B. **Vorbelegen** ausgewählt ist, heißt der Menüpunkt **Automatisch (Vorbelegen)**.
- **Vorbelegen**—LabVIEW erstellt eine VI-Kopie für jeden Aufruf des ablaufinvarianten VIs in der Struktur. In der zeitgesteuerten Schleife oder Struktur wird ein bestimmtes Symbol angezeigt, um zu verdeutlichen, dass Kopien für ablaufinvariante VIs in der Struktur vorbelegt werden.
- **Nach Bedarf zuweisen**—LabVIEW erstellt erst dann eine VI-Kopie, wenn das ablaufinvariante VI in der Struktur aufgerufen wird. In der zeitgesteuerten Schleife oder Struktur wird ein bestimmtes Symbol angezeigt, um zu verdeutlichen, dass Kopien für ablaufinvariante VIs in der Struktur gemeinsam genutzt werden.

Automatisches Verbinden von Tunneln in Strukturen mit mehreren Cases

LabVIEW kann Eingangs- und Ausgangstunnel in Strukturen mit mehreren Cases automatisch verbinden. Sie können einen Eingangs- und Ausgangstunnel so konfigurieren, dass Verbindungen automatisch erstellt werden, indem Sie den entsprechenden Ausgangstunnel mit der rechten Maustaste anklicken und eine der folgenden Optionen aus dem Kontextmenü **Verknüpfter Eingangstunnel** auswählen:

- **Offene Cases erstellen & verbinden**—LabVIEW verbindet Tunnel in allen vorhandenen Cases, die noch nicht verbunden sind, automatisch. Außerdem werden die Tunnel in allen neu erstellten Cases automatisch verbunden.
- **Erstellen**—LabVIEW verbindet Tunnel in allen neu erstellten Cases automatisch. Vorhandene Cases bleiben unverbunden.

Bei Auswahl einer der zuvor genannten Optionen wird zwischen Eingangs- und Ausgangstunnel eine Verbindung erstellt. Wenn die Tunnel verbunden sind, wird ein weißes Dreieckssymbol angezeigt. Klicken Sie mit der rechten Maustaste auf einen verbundenen Ausgangstunnel und wählen Sie **Verknüpfter Eingangstunnel**»Suchen oder **Verknüpfter Eingangstunnel**»Löschen aus dem Kontextmenü, um einen verbundenen Eingangstunnel zu suchen oder die Verbindung zwischen Tunneln zu entfernen.



Hinweis Bei gestapelten Sequenzstrukturen kann das automatische Verbinden von Tunneln nicht konfiguriert werden.

Verschiedene Verbesserungen am Blockdiagramm

In LabVIEW 8.6 gibt es folgende weitere Verbesserungen im Blockdiagramm:

- In LabVIEW 8.6 arbeiten alle Vergleichsfunktionen mit LabVIEW-Klassen.
- LabVIEW 8.6 erzeugt Prototypinformationen für boolesche Anzeige- und Bedienelemente im Anschlussfeld eines VIs in einer DLL. Wird die DLL in LabVIEW aufgerufen, behandelt LabVIEW das boolesche Element als vorzeichenlosen 8-Bit-Integer. LabVIEW gibt einen Fehler aus, wenn die Beschriftung des Elements `Boolesch` lautet. Bei `Boolesch` muss Groß- und Kleinschreibung nicht beachtet werden.

Verbesserungen am Frontpanel

In LabVIEW 8.6 gibt es folgende Verbesserungen am Frontpanel und dazugehörigen Funktionen:

Verbesserungen an 3D-Graphen

(Windows) Der 3D-Oberflächengraph, der parametrische 3D-Graph und der 3D-Kurvengraph aus LabVIEW 8.5 und früheren Versionen wurden umbenannt in ActiveX-3D-Oberflächengraph, ActiveX-3D-Graph (parametrisch) und ActiveX-3D-Kurvengraph. ActiveX-3D-Graphen arbeiten mit ActiveX und VIs zur 3D-Darstellung.

LabVIEW 8.6 enthält eine XControl-Version des 3D-Graphen, die auf allen Versionen zur Verfügung steht. Mit dem 3D-Oberflächengraphen, dem 3D-Graphen (parametrisch) und dem 3D-Liniengraphen lassen sich 3D-Daten an einem 3D-Plot darstellen. Mit Hilfe des Dialogfelds **Eigenschaften für 3D-Graph** lässt sich die Darstellung eines 3D-Graph-XControls anpassen. Sie können die allgemeine Darstellung des Graphen konfigurieren, die x-, y- und z-Plots formatieren, die Achsen des Graphen formatieren, Wertepaare hinzufügen und Cursor hinzufügen.

Verschiedene Verbesserungen am Frontpanel

Ein Privatdatenelement kann kein XControl und keine strikt typisierte XControl-Referenz enthalten.

Optische Verbesserungen

In LabVIEW 8.6 gibt es folgende Verbesserungen an der LabVIEW-Umgebung:

Verbesserungen am Fenster "Fehlerliste"

LabVIEW 8.6 enthält Verbesserungen an der Fehlermeldung in LabVIEW-Klassen. Dadurch werden möglicherweise weniger funktionstüchtige Objekte in der **Fehlerliste** angezeigt.

Verbesserungen der Ausführungseffizienz

Die Dateigröße der VIs wurde verringert.

Verbesserungen am Fenster "VI-Hierarchie"

In LabVIEW 8.6 gibt es folgende Verbesserungen am Fenster **VI Hierarchie**:

- In der **VI-Hierarchie** werden LabVIEW-Klassen, dynamische Member-VIs, XControls, DLLs und * .m-Dateien aufgeführt, auf die durch MathScript-Knoten, Express-VIs und statische VI-Referenzen verwiesen wird.
- Zum Hervorheben der Aufrufkette des VIs klicken Sie mit der rechten Maustaste im Fenster **VI-Hierarchie** auf das Symbol eines angehaltenen VIs und wählen Sie **Aufrufkette anzeigen** aus dem Kontextmenü. Die Aufrufkette ist die Kette der VI-Aufrufe vom aktuellen VI zum Haupt-VI. Wenn Sie die Hierarchieverbindung hervorheben, wird die Verbindung durch eine dicke, rote Linie gekennzeichnet.

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

Verbesserungen an Dialogfeldern

In LabVIEW 8.6 wurden folgende Verbesserungen an Dialogfeldern durchgeführt.

Verbesserungen am Dialogfeld "Warnungen"

Das Dialogfeld **Warnungen** wurde in die Dialogfelder **Ladewarnungen (Zusammenfassung)** und **Warnungen zum Speichern für ältere Version** umgewandelt. Klicken Sie in einem der beiden Dialogfelder auf die Schaltfläche **Details anzeigen**, um das Dialogfeld **Liste mit Lade- und Speicherwarnungen** zu öffnen und Details zu jeder Warnkategorie anzuzeigen. Sie können aber auch **Liste mit**

Lade- und Speicherwarnungen aus dem Menü **Ansicht** wählen, um das Dialogfeld **Liste mit Lade- und Speicherwarnungen** anzuzeigen.

Im Dialogfeld **Ladewarnungen (Zusammenfassung)** wird für das jeweils zu ladende Hauptobjekt eine Liste mit verschiedenen Warnkategorien angezeigt.

Im Dialogfeld **Warnungen zum Speichern für ältere Version** wird für das jeweils zu speichernde Hauptobjekt eine Liste mit verschiedenen Warnkategorien angezeigt.

Änderungen am Dialogfeld “Application Builder”

In LabVIEW 8.6 gibt es folgende Änderungen am LabVIEW Application Builder:

- Auf der Seite **Weitere Installationsprogramme** im Dialogfeld **Eigenschaften für Installer** wird im Textfeld **Installer-Speicherort** der Speicherort nur angezeigt. Der Installer-Speicherort kann nicht mehr geändert werden. Wenn LabVIEW die Installer-Quelle nicht automatisch finden kann, werden Sie später im Build-Prozess aufgefordert, die Quelle manuell einzugeben.
- Wenn Sie einen Installer erstellen, der weitere Installer oder Komponenten enthält, wird möglicherweise das Dialogfeld **Erforderliche Softwarepakete** angezeigt, wenn LabVIEW den ausgewählten Installer oder die Abhängigkeiten des Installers nicht finden kann.
- **(Windows)** Mit Hilfe der Build-Spezifikation für Webdienste lassen sich VIs in einem LabVIEW-Projekt erstellen und als Webdienste an andere Nutzer weitergeben. Klicken Sie zur Anzeige des Dialogfelds **Eigenschaften für Webdienst** mit der rechten Maustaste im **Projekt-Explorer** auf **Build-Spezifikationen** und wählen Sie **Neu»Webdienst (RESTful)** aus dem Kontextmenü aus. LabVIEW-Webdienste stehen nur im LabVIEW Full Development System und LabVIEW Professional Development System zur Verfügung.

Verbesserungen am Dialogfeld “Eigenschaften für Umgebungsvariable”

LabVIEW 8.6 enthält die folgenden Verbesserungen am Dialogfeld **Eigenschaften für Umgebungsvariable**:

- Auf der Seite **Variable** im Dialogfeld **Eigenschaften für Umgebungsvariable** wurde die Option **Mit Quelle verbinden** umbenannt in **Aliasing aktivieren**. Wenn Sie die Option **Aliasing aktivieren** aktivieren, kann die Zugriffsart der Umgebungsvariablen festgelegt werden.
- Die Optionen **Pufferung verwenden** und **Einzelne Datenquelle** der Seite **Variable** wurden auf die Seite **Netzwerk** des Dialogfelds **Eigenschaften für Umgebungsvariable** verschoben.
- Die Seite **Beschreibung** des Dialogfelds **Eigenschaften für Umgebungsvariable** wird ohne das DSC-Modul angezeigt.

Verbessertes Dialogfeld “Optionen”

In LabVIEW 8.6 gibt es folgende Verbesserungen an der Seite **Webserver: Konfiguration** im Dialogfeld **Optionen**:

- Sie können mehrere Schnittstellen für den LabVIEW-Webserver konfigurieren. Mit der Option **Serverbeschriftung** lässt sich der Name einer Serverschnittstelle benutzerspezifisch anpassen und im Listenfeld **Serverbeschriftungen (Ports)** werden alle derzeit konfigurierten Schnittstellen für den Webserver angezeigt.
- Die **Listener-IP-Adresse** legt die IP-Adresse für die ausgewählte Webserver-Schnittstelle fest. Im Pull-down-Menü finden Sie alle auf dem lokalen Rechner verfügbaren Adressen. Bei der Arbeit mit dem LabVIEW Real-Time Module passt sich dieses Feld nicht automatisch an.
- Mit den Optionen unter **Zugriff gewähren** lassen sich bestimmte Funktionen für die ausgewählte Webserver-Schnittstelle konfigurieren. Im Abschnitt **Zugriff gewähren** können Sie Netzwerk-Frontpanel, Skripte, Webdienste, statischen Inhalt und Momentaufnahmen aktivieren.

LabVIEW 8.6 enthält im Dialogfeld **Optionen** die neue Seite **Webdienste: Sicherheit**, auf der Sicherheitseinstellungen für LabVIEW-Webdienste konfiguriert werden können.

Verschiedene Verbesserungen an Dialogfeldern

In LabVIEW 8.6 gibt es folgende weitere Verbesserungen an Dialogfeldern:

- Im Dialogfeld **Konfliktlösung** gibt es Optionen zum Lösen von Konflikten, die während der Weitergabe von Anwendungen auftreten. Das Dialogfeld wird beispielsweise angezeigt, wenn Sie ein VI auf ein Zielsystem übertragen, das ausgeschaltet ist. Wenn das Problem beseitigt werden kann, wird in der Spalte **Konfliktlösung** ein Pulldown-Menü eingeblendet. Zum Lösen des Problems wählen Sie eine Option aus der Liste aus. Das Dialogfeld enthält auch die beiden Schaltflächen **Vorheriger** und **Nächster** zum Hervorheben des vorherigen oder nächsten Objekts mit einem Konflikt in der Liste **Konflikte**.
- Die Seite **Datentyp** des Dialogfelds **Erforderliche Softwarepakete** und des Dialogfelds **Eigenschaften für numerische Konstante** sowie die Seite **Ausgangskonfiguration** des Dialogfelds **Eigenschaften für numerischen Knoten** enthalten die Option **Überlaufzustand einschließen**, mit der festgelegt werden kann, ob LabVIEW einen Überlaufzustand bei Festkommazahlen einschließen soll.
- Mit dem Dialogfeld **Fehlende Objekte suchen** können Sie nach allen Objekten in einem LabVIEW-Projekt suchen, die auf eine Datei auf dem Datenträger verweisen, die von LabVIEW nicht gefunden wurde. Klicken Sie zur Anzeige dieses Dialogfelds mit der rechten Maustaste auf die höchste Ebene im **Projekt-Explorer** und wählen Sie **Fehlende Objekte suchen**.

Verbesserungen an den Paletten

In LabVIEW 8.6 gibt es folgende Verbesserungen an den Paletten:

Programmatisches Bearbeiten der Elemente- und Funktionenpalette

Mit Hilfe der VIs zur Palettenbearbeitung lassen sich Paletten programmatisch erstellen und bearbeiten. Verwenden Sie die VIs zur Palettenbearbeitung, wenn Sie viele Paletten bearbeiten, neue Paletten erstellen oder die Darstellung einer Palette nach der Bearbeitung prüfen möchten.

Beispiele zum programmatischen Bearbeiten der **Elemente-** und **Funktionen-**Palette finden Sie im Verzeichnis `labview\examples\Palette API`.

Weitere Informationen zum programmatischen Bearbeiten von Paletten finden Sie im Buch **VI- und Funktionenübersicht»VIs und Funktionen zur Programmierung»VIs und Funktionen zur Applikationssteuerung»Palettenbearbeitungs-VIs** auf der Registerkarte **Inhalt** in der *LabVIEW-Hilfe*.

Verschiedene Verbesserungen an den Paletten

In LabVIEW 8.6 gibt es folgende weitere Verbesserungen an den Paletten:

- Die VIs und Funktionen der Palette **XML** werden nun auf der Palette **LabVIEW-Schema** angezeigt. Die Palette **LabVIEW-Schema** ist eine Unterpalette der Palette **XML**.
- Beim Aktualisieren von LabVIEW versucht das Programm, das Palettenformat der zuvor installierten LabVIEW-Version wiederherzustellen.

Verschiedene Verbesserungen an der Umgebung

In LabVIEW 8.6 gibt es folgende weitere Verbesserungen an der Entwicklungsumgebung:

- Sie können einen LabVIEW-Built-Installer und die Dateien, die damit installiert werden, im Hintergrund deinstallieren. Geben Sie dazu folgenden Befehl in das Kommandozeilenfeld ein:

```
uninst.exe /qb /xProduktname, wobei Produktname der Name des Produkts ist, das deinstalliert werden soll.
```

Stimmt der eingegebene Produktname nicht genau mit dem Namen des Produkts überein, schlägt die Deinstallation fehl ohne Nachricht über den aufgetretenen Fehler. Die Datei `uninst.exe` befindet sich im Verzeichnis `C:\Programme\National Instruments\Shared\NIUninstaller`. Ist das Verzeichnis `C:\Programme\National Instruments\Shared\NIUninstaller` nicht das Arbeitsverzeichnis, geben Sie den kompletten Pfad auf dem Datenträger zu `uninst.exe` ein.

- Mit Hilfe der Bibliothek `fileVersionInfo.llb` aus dem Verzeichnis `labview\vi.lib\Platform` lassen sich Angaben zur Version einer Applikation oder DLL programmatisch abrufen. Dieses VI ist nicht auf der Palette **Funktionen** verfügbar.
- Unter dem Menüpunkt **Ausführen** gibt es die Option **Haltepunkte** nicht mehr. Wählen Sie stattdessen zum Verwalten von Haltepunkten in einer VI-Hierarchie den Menüpunkt **Ansicht» Haltepunkt-Manager**.
- Befindet sich eine Projektbibliothek in einem Ordner, gelten die Operationen der Versionsverwaltung nun für den Inhalt der Bibliothek.
- Die Bibliothek `_goopsup.llb` wurde in das Verzeichnis `labview\vi.lib\Utility` verschoben.

Neue und verbesserte VIs, Funktionen und Knoten

In LabVIEW 8.6 gibt es folgende neue und veränderte VIs und Funktionen. Weitere Informationen zu VIs, Funktionen und Knoten finden Sie im Buch **VI- und Funktionsreferenz** auf der Registerkarte **Inhalt** in der *LabVIEW-Hilfe*.

Neue VIs und Funktionen

LabVIEW 8.6 enthält folgende neue VIs und Funktionen:

VIs und Funktionen der Palette “Auf Meldung warten (Fortgeschritten)”

Auf der Palette **Auf Meldung warten (Fortgeschritten)** gibt es folgende neue Funktionen:

- “Auf Meldung von mehreren mit Melderhistorie warten”
- “Auf Meldung mit Melderhistorie warten”

VIs und Funktionen zur Applikationssteuerung

Auf der Palette **Applikationssteuerung** gibt es die neue Unterpalette **Palettenbearbeitung** mit folgenden neuen VIs:

- “Palette lesen”
- “Paletten aktualisieren”
- “Palette schreiben”

Vergleichsfunktionen

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

- “Festkommaüberlauf?”

VIs und Funktionen für ActiveX und .NET (Windows)

Auf der Palette **Konnektivität** gibt es die neue Unterpalette **Webdienste** mit folgenden neuen VIs:



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

- “Existenz der Session prüfen”
- “Session erstellen”
- “Session-Variable löschen”
- “Session entfernen”

- “Ausgabe löschen”
- “Session-ID-Cookie ermitteln”
- “Alle Formulardaten lesen”
- “Alle Abfragevariablen lesen”
- “Alle Session-Variablen lesen”
- “Formulardaten lesen”
- “POST-Daten lesen”
- “Anfragevariable lesen”
- “Session-Variable lesen”
- “ESP-Vorlage rendern”
- “ESP-Variable festlegen”
- “HTTP-Header festlegen”
- “HTTP-Umleitung festlegen”
- “HTTP-Antwort-Code festlegen”
- “HTTP-Antwort-MIME-Typ festlegen”
- “Antwort schreiben”
- “Session-Variable schreiben”

Auf der Palette **Webdienste** gibt es die neue Unterpalette **Utilities** mit folgenden neuen VIs:

- “HTTP-URL konvertieren”
- “HTTP-URL zurückkonvertieren”

Festkommfunktionen

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

- “Festkomma-Überlaufzustand löschen”
- “Festkomma-Überlaufzustand einschließen”
- “Festkomma-Überlaufzustand entfernen”

Hilfs-VIs

Die Palette **Helfer** enthält im LabVIEW Full Development System und im LabVIEW Professional Development System das Express-VI “Sensordarstellung”. Mithilfe des Express-VIs “Sensordarstellung” lassen sich reale Daten auf ein 3D-Modell anwenden.

Queue-Funktion

Auf der Palette **Queue-Funktionen** gibt es folgende neue Funktion:

- “Element einfügen (verlustbehaftet)”

Semaphor-VIs

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

- “Semaphor-Referenz anfordern”
- “Semaphor-Referenz freigeben”

XML-VIs und -Funktionen

Auf der Palette **XML** gibt es die neue Unterpalette **XML-Parser** mit folgenden neuen VIs und Funktionen:

- “Schließen”
- “Erstes Unterobjekt ermitteln (kein Text)”
- “Nächstes Geschwisterobjekt ermitteln (kein Text)”
- “Knotentextinhalt ermitteln”
- “Methodenknoten (XML)”
- “Laden”
- “Neu”
- “Eigenschaftsknoten (XML)”
- “Speichern”

VISA: Fortgeschrittene VIs und Funktionen

Auf der Palette **VISA: Fortgeschritten** gibt es folgenden neuen Knoten:

- “Eigenschaftsknoten (VISA)”

Veränderte VIs, Funktionen und Knoten

Folgende VIs, Funktionen und Knoten haben sich in LabVIEW 8.6 geändert.

Fortgeschrittene Datei-VIs und -Funktionen

Auf der Palette **Dateifunktionen (Fortgeschritten)** gibt es folgende geänderte Funktion:

Die Funktion “Datenträger-Info lesen” enthält den Ausgang **Sektorgröße (Byte)**, mit dem die Größe der kleinsten physikalischen Speichereinheit auf der Festplatte angezeigt wird. Wenn Sie eine Datei ohne Pufferung öffnen, muss die Dateigröße ein Vielfaches dieses Werts sein. Zum Deaktivieren der Pufferung setzen Sie den Eingang **Pufferung deaktivieren** der Funktion “Öffnen/Erstellen/Ersetzen einer Datei” auf TRUE.

Vergleichsfunktionen

Auf der Palette **Vergleich** gibt es folgende geänderte Funktionen:

- **Wertebereich prüfen und erzwingen**—Unterstützt den Festkomma-Datentyp.
- **Max & Min**—Unterstützt den Festkomma-Datentyp.

Datei-I/O-VIs und -Funktionen

Auf der Palette **Datei-I/O** haben sich folgende Funktionen geändert:

Die Funktion “Öffnen/Erstellen/Ersetzen einer Datei” enthält den Eingang **Pufferung deaktivieren**, mit dem festgelegt werden kann, ob die Datei mit oder ohne Pufferung geöffnet, erstellt oder ersetzt wird.

Deaktivieren Sie die Pufferung, um die Datenübertragung in bestimmten Situationen zu beschleunigen. Deaktivieren Sie die Pufferung z. B. beim Streaming großer Datenmengen oder beim Speichern von Daten auf RAIDs (logische Laufwerke aus mehreren physischen Festplatten). Auf diese Weise werden die Daten direkt auf die Festplatte geschrieben.

LabVIEW-Schema

Auf der Palette **LabVIEW-Schema** gibt es folgende geänderte Funktionen:

- **In XML konvertieren**—Unterstützt LabVIEW-Klassen und den Festkomma-Datentyp.
- **Aus XML zurückkonvertieren**—Unterstützt LabVIEW-Klassen und den Festkomma-Datentyp.

Numerische Funktionen

Auf der Palette **Numerisch** gibt es folgende geänderte Funktionen:

- **Dividieren**—Unterstützt den “Festkomma”-Datentyp.
- **Kehrwert**—Unterstützt den Festkomma-Datentyp.
- **Quadratwurzel**—Unterstützt den Festkomma-Datentyp.

Protokoll-VIs

Die Palette **Protokoll-VIs** enthält VIs mit Ein- und Ausgängen für das LabVIEW Report Generation Toolkit. Diese Ein- und Ausgänge werden ignoriert, falls das Report Generation Toolkit nicht installiert ist. Auf der Palette **Protokoll** haben sich folgende VIs geändert:

- **Horizontale Linie in Protokoll einfügen**—Der Standardwert des Eingangs **Breite** wurde von Pixel in Prozent geändert.
- **Protokolltext anfügen**—Enthält den Eingang **Format-String**, mit dem die Formatierung für Zahlen festgelegt wird, wenn diese in Zeichen umgewandelt werden.
- **VI-Blockdiagramm an Protokoll anfügen**—Alle Instanzen enthalten den Eingang **Ausrichtung**, mit dem die Ausrichtung von Grafiken in einem Protokoll festgelegt wird.
- **VI-Hierarchie an Protokoll anfügen**—Alle Instanzen enthalten den Eingang **Ausrichtung**, mit dem die Ausrichtung von Grafiken in einem Protokoll festgelegt wird.
- **VI-Symbol an Protokoll anfügen**—Alle Instanzen enthalten den Eingang **Ausrichtung**, mit dem die Ausrichtung von Grafiken in einem Protokoll festgelegt wird.
- **Liste der VI-Elemente an Protokoll anfügen**—Alle Instanzen enthalten den Eingang **Ausrichtung**, mit dem die Ausrichtung von Grafiken in einem Protokoll festgelegt wird.
- **Liste der SubVIs an Protokoll anfügen**—Alle Instanzen enthalten den Eingang **Ausrichtung**, mit dem die Ausrichtung von Grafiken in einem Protokoll festgelegt wird.
- **Neue Protokollzeile**—Enthält einen Eingang **Zeilenanzahl**, mit der die Anzahl der Zeilen festgelegt wird, die zum Protokoll hinzugefügt werden.
- **Protokoll in Datei speichern**—Enthält den Eingang **Eingabeaufforderung zum Ersetzen?**, mit dem festgelegt werden kann, ob ein Dialogfeld angezeigt werden soll, wenn Sie einen bereits vorhandenen Dateinamen angeben.

Spektrumanalyse-VIs (Nicht im Basispaket)

Auf der Palette **Spektrumanalyse** haben sich folgende VIs geändert:

Das VI “STFT-Spektrogramm” enthält den Eingang **Zeit-Freq-Konfig**, mit dem die Konfiguration der **Frequenzunterteilungen** und die Anzahl der Spalten in **STFT-Spektrogramm {X}** festgelegt werden. Ein weiterer Eingang des VIs ist **Energieeinsparung?**. Damit wird festgelegt, ob **STFT-Spektrogramm {X}** skaliert wird, so dass die Energie im Zeit-Frequenz-Bereich der Energie im Zeitbereich entspricht.

Knoten zum Aufruf externer Bibliotheken

Das Dialogfeld der **Funktion zum Aufruf externer Bibliotheken** enthält folgende Änderungen:

- Das Pulldown-Menü **Datentyp** auf der Registerkarte **Parameter** enthält die numerischen Datentypen **Vorzeichenbehafteter zeigergroßer Integer** und **Vorzeichenloser zeigergroßer Integer**. Bei Verwendung dieser beiden Datentypen passt sich der Knoten zum Aufruf externer Bibliotheken an das Betriebssystem an, auf dem er ausgeführt wird und gibt Daten der entsprechenden Größe an die DLL-Funktion aus. Auf einer 64-Bit-Plattform werden diese numerischen Datentypen in 64-Bit-Integer umgewandelt. Auf einer 32-Bit-Plattform werden die numerischen Datentypen in 32-Bit-Integer umgewandelt.

- Der **Knoten zum Aufruf externer Bibliotheken** unterstützt die Verwendung von Platzhaltern (*), um die Referenz auf den **Namen oder Pfad der DLL** zwischen 32- und 64-Bit-Windows plattformunabhängig zu machen. Ein *-Platzhalter links der Dateierweiterung bedeutet 32 auf 32-Bit-Windows und 64 auf 64-Bit-Windows. So z. B. bedeutet `myLibrary*.dll` entweder `myLibrary32.dll` oder `myLibrary64.dll`. Zwei *-Platzhalter werden verwendet, wenn der Name der DLL vorher keine Referenz auf die Plattform hatte, aber der neue Name der DLL. So bedeutet z. B. `myLibrary**.dll` entweder `myLibrary.dll` auf einer 32-Bit-Plattform oder `myLibrary_64.dll` auf einer 64-Bit-Plattform.

Umgebungsvariablenknoten

Die Umgebungsvariable enthält die folgenden Änderungen:

- Enthält eine Option zum Aktivieren des Bedienelements **Timeout in ms** und Anzeigeelements **Timeout?** für Netzwerk-Umgebungsvariablen, die zum Lesen von Daten konfiguriert sind.
- Die Ausführungseffizienz von Umgebungsvariablen bei der Arbeit mit vielen Umgebungsvariablen in einem Blockdiagramm wurde in LabVIEW 8.6 verbessert.
- Die zeitgetriggerte Variable, die bei Installation des LabVIEW Real-Time Modules verfügbar ist, wird nicht zusammen mit den Standardumgebungsvariablen angezeigt. Klicken Sie zum Erstellen einer zeitgesteuerten Variable mit der rechten Maustaste auf ein unterstütztes Zielsystem und wählen Sie **Neu»Zeitgetriggerte Variable** aus dem Kontextmenü.

Verschiedene Änderungen an VIs, Funktionen und Knoten

Nachfolgend eine lose Auflistung sonstiger Änderungen an VIs in LabVIEW 8.6:

Ein Wert des Eingangs **Methode** des VIs "Komplexe Zahlen sortieren" wurde umbenannt. **Betrag** wurde umbenannt in **Betrag, Reell, Imaginär**. Der Eingang **Methode** enthält außerdem den neuen Wert **Betrag, Phasenwinkel**. **Betrag, Reell, Imaginär** sortiert die Elemente zuerst entsprechend des Betrags. Elemente mit dem gleichen Betrag werden nach Realteil und anschließend nach Imaginärteil sortiert. **Betrag, Phasenwinkel** sortiert die Elemente zuerst entsprechend des Betrags. Elemente mit dem gleichen Betrag werden anschließend nach dem Phasenwinkel sortiert. Der Phasenwinkel liegt im Bereich von minus Pi bis Pi.

Neue Klasse, Eigenschaften, Methoden und Ereignisse

In LabVIEW 8.6 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 8.6»Neue VI-Server-Objekte** auf der Registerkarte **Inhalt** in der *LabVIEW-Hilfe*.

Eigenschaften und Methoden für XML-Parser

Eine Liste mit neuen Eigenschaften und Methoden, die mit den XML-Parser-VIs und -Funktionen verwendet werden können, finden Sie im Buch **Eigenschaften- und Methodenübersicht»XML-Parser** auf der Registerkarte **Inhalt** der *LabVIEW-Hilfe*.

Eigenschaften von 3DPC_OberflächePlot

Eine Liste mit neuen Eigenschaften, die mit einem 3D-Graphen verwendet werden können, finden Sie im Buch **Eigenschaften- und Methodenübersicht»3DPC_OberflächePlot** auf der Registerkarte **Inhalt** in der *LabVIEW-Hilfe*.

Blockdiagramm automatisch aufräumen

Wählen Sie zum automatischen Neuverbinden aller vorhandenen Verbindungen und zum Neuordnen der Blockdiagrammobjekte **Bearbeiten»Diagramm aufräumen**. Wählen Sie zum Konfigurieren der Optionen zum Aufräumen des Blockdiagramms den Punkt **Werkzeuge»Optionen** zur Anzeige des Dialogfelds **Optionen** und wählen Sie **Blockdiagramm: Aufräumen** aus der Liste **Kategorie** aus. Sie

können automatisch alle Bedienelemente auf die linke Seite und alle Anzeigeelemente auf die rechte Seite des Blockdiagramms schieben, eine bestimmte Anzahl an Pixeln zwischen Blockdiagrammobjekten und Verbindungen einfügen und das Blockdiagramm kompakt darstellen.

Objekte über Schnell Tasten einfügen

Mit Hilfe des Dialogfelds **Schnelleinfügeliste** können Sie Frontpanel- und Blockdiagrammobjekte nach Namen suchen und direkt ohne Verwendung der **Elemente-** oder **Funktionen-**Palette einfügen.

(Windows und Linux) Zur Anzeige des Dialogfelds **Schnelleinfügeliste** wählen Sie **Ansicht»Schnelleinfügeliste** oder drücken Sie die Tastenkombination <Strg + Leertaste>. Bei chinesischen Tastaturen drücken Sie <Strg + Shift + Leertaste>. **(Mac OS)** Drücken Sie zur Anzeige des Dialogfelds <Command + Shift + Leertaste>.

Hinweise zur Verwendung des Dialogfelds **Schnelleinfügeliste** finden Sie in der *LabVIEW-Hilfe* auf der Registerkarte **Inhalt** im Buch **Grundlagen»LabVIEW-Umgebung»Anleitung»Suchen von Objekten**.

Eigenschaften mehrerer Objekte bearbeiten

Sie können mehrere Objekte auf dem Frontpanel oder im Blockdiagramm auswählen und deren gemeinsame Eigenschaften bearbeiten. Zur Auswahl mehrerer Objekte ziehen Sie mit dem Positionierwerkzeug ein Rechteck um alle gewünschten Objekte auf oder halten Sie die <Shift>-Taste gedrückt, während Sie jedes Objekt anklicken. Klicken Sie mit der rechten Maustaste auf eines der markierten Objekte und wählen Sie **Eigenschaften** aus dem Kontextmenü aus, um zum Dialogfeld **Eigenschaften** zu gelangen. Im Dialogfeld **Eigenschaften** werden nur die Registerkarten und Eigenschaften angezeigt, die bei allen Objekten vorhanden sind. Zur Anzeige von mehr Registerkarten und Eigenschaften sollten daher ähnliche Objekte ausgewählt werden. Bei Auswahl von mehreren Objekten, die nicht über gemeinsame **Eigenschaften** verfügen, werden in diesem Dialogfeld keine Registerkarten oder Eigenschaften angezeigt.

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

In LabVIEW 8.6 ist es möglich, ein VI als Webdienst zu veröffentlichen. Webdienste sind eine standardisierte Methode für Server zum Weitergeben von Applikationen, auf die jeder HTTP-Client zugreifen kann. LabVIEW-Webdienste unterstützen die wichtigsten Plattformen und Programmiersprachen und ermöglichen eine einfache Implementierung und Weitergabe von Webapplikationen in einem Netzwerk mithilfe von LabVIEW.

Der Funktionsumfang von Webdiensten lässt sich auf der Seite **Webserver: Konfiguration** im Dialogfeld **Optionen** aktivieren. Zum Erstellen und Weitergeben eines Webdiensts klicken Sie mit der rechten Maustaste im **Projekt-Explorer** auf **Build-Spezifikationen** und wählen Sie **Neu»Webdienst (REST-ful)** aus dem Kontextmenü aus, um das Dialogfeld **Eigenschaften für Webdienst** anzuzeigen.

LabVIEW-Webdienste stehen nur im LabVIEW Full Development System und LabVIEW Professional Development System zur Verfügung.

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

NI-DSM

(Windows) LabVIEW 8.6 enthält den NI-DSM (Distributed System Manager), der den Variablenmanager als Tool zum Verwalten von Umgebungsvariablen ablöst. Mit Hilfe des Systemmanagers lassen sich Variablen, Prozesse, I/O-Server und Webdienste erstellen und beobachten. Außerdem ermöglicht der Systemmanager die Arbeit mit der Engine für Umgebungsvariablen sowie die Verwaltung von Sicherheitseinstellungen und Aliasen.

Weitere Hinweise zum DSM finden Sie unter dem Thema **Grundlagen»Arbeit im Netzwerk in LabVIEW»Allgemeines»Übertragung aktueller Frontpanel-Werte mit Umgebungsvariablen** auf der Registerkarte **Inhalt** in der *LabVIEW-Hilfe*.

Verbesserungen am LabVIEW-MathScript (Nicht im Basispaket enthalten)

In LabVIEW 8.6 gibt es folgende Verbesserungen und Änderungen an MathScript:



Hinweis Wählen Sie **Werkzeuge»MathScript-Fenster** zur Anzeige des **LabVIEW-MathScript-Fensters**.

Neue MathScript-Funktionen

LabVIEW 8.6 enthält folgende neue MathScript-Funktionen. Diese Funktionen können im **LabVIEW MathScript-Fenster** oder im MathScript-Knoten verwendet werden.

MathScript-Klasse	Funktion
Fortgeschritten	sphbesselh, sphbesselj, sphbessely
Approximation	csaps, lsqcurvefit
Audio	auread
Befehle	keyboard
Filterdesign	gaussfir, yulewalk
Umsetzung von Filtern	cconv
Integration	quad
ODE	odeset, radau5
Plots	imagesc, xlim, ylim, zlim
Spektrumanalyse	buffer, cpsd, mscohere, pburg, pcov, peig, periodogram, pmcov, pmusic, pwelch, pyulear, rooteig, rootmusic, specgram, spectrogram, tfestimate
pde	pdeelliptic
Support	warning, xlsread, xlswrite
Signalverlaufserzeugung	stepfun
Fenster	taylorwin

LabVIEW-MathScript auf Mac OS und Linux

Das **LabVIEW-MathScript-Fenster** und der MathScript-Knoten sind unter Mac OS und Linux verfügbar. Einige Funktionen stehen auf diesen Betriebssystemen jedoch nicht zur Verfügung. In der folgenden Tabelle sehen Sie, welche MathScript-Funktionen es nicht unter Mac OS und Linux gibt. Mit **x** wird angezeigt, dass die Funktion auf dem entsprechenden Betriebssystem nicht zur Verfügung steht.



Hinweis Die Funktionen `getFileproperty`, `load`, `save` und `setfileproperty` gibt es unter Mac OS und Linux. Diese Funktionen können aber nicht mit Binärdateien für Messwerte (*.tdm oder *.tdms) verwendet werden.

MathScript-Funktion	Mac OS	Linux
<code>aich</code>	x	
<code>aiwf</code>	x	
<code>aoch</code>	x	
<code>aowf</code>	x	
<code>calllib</code>	x	x
<code>dioread</code>	x	
<code>diowrite</code>	x	
<code>dos</code>	x	x
<code>ginput</code>	x	x
<code>gtext</code>	x	x
<code>libfunctionsview</code>	x	x
<code>libisloadeded</code>	x	x
<code>loadlibrary</code>	x	x
<code>pause</code>	x	x
<code>system</code>	x	x
<code>unloadlibrary</code>	x	x
<code>waitforbuttonpress</code>	x	x

Verbesserungen bei der Fehlersuche in MathScript-Knoten

In LabVIEW 8.6 gibt es folgende Verbesserungen bei der Fehlersuche in MathScript-Knoten:

- In einem MathScript-Knoten sind Einzelschrittausführung, Highlight-Funktion und Haltepunkte zur Fehlersuche in einem Skript möglich.
- Im grauen Bereich links im MathScript-Knoten werden folgende Objekte angezeigt:
 - Rotes Fehlersymbol neben Skriptzeilen mit einem Fehler
 - Warnsymbole
 - Haltepunkte
- Wenn Sie einen MathScript-Knoten im Fenster **Fehlerliste** auswählen und auf die Schaltfläche **Fehler anzeigen** klicken, wird die Skriptzeile hervorgehoben, die den Fehler enthält.

Highlight-Funktion für Skript in MathScript-Knoten

Mit der Highlight-Funktion für Skripte werden verschiedene Bestandteile eines Skripts in einem MathScript-Knoten mit unterschiedlichen Farben markiert. Mit diesen Farben wird die Lesbarkeit des Skripts verbessert und die Fehlersuche in Skripten mit Fehlern und Skripten, die unerwartete Daten ausgeben, erleichtert. Mit der Highlight-Funktion wird z. B. angezeigt, wenn eine benutzerdefinierte Funktion oder eine Variable eine integrierte MathScript-Funktion überschreibt. Die Highlight-Funktion ist per Voreinstellung aktiviert außer in MathScript-Knoten, die zuletzt in einer früheren Version von LabVIEW gespeichert wurden, in der benutzerdefinierte Farben verwendet werden.

Zum Aktivieren oder Deaktivieren der Highlight-Funktion für einen bestimmten MathScript-Knoten klicken Sie mit der rechten Maustaste in den entsprechenden Knoten und wählen Sie **Highlight-Funktion für Skript aktivieren** oder **Highlight-Funktion für Skript deaktivieren** aus dem Kontextmenü. Zum Aktivieren oder Deaktivieren der Highlight-Funktion für alle MathScript-Knoten und zum Anpassen der Farben für MathScript-Knoten mit aktivierter Highlight-Funktion verwenden Sie die Seite **MathScript: Highlight-Funktion für Skript** des Dialogfelds **Optionen**.

Verschiedene Neuerungen und Änderungen an MathScript

In LabVIEW 8.6 gibt es folgende Verbesserungen und Änderungen am MathScript:

- Im Fenster **VI-Hierarchie** werden *.m-Dateien angezeigt, auf die durch MathScript-Knoten verwiesen wird.
- Die Funktion `delete` enthält den Eingang **obj**, über den Plotobjekte gelöscht und aus der **Variablenliste** in **LabVIEW-MathScript** entfernt werden.
- Die Funktion `filter` enthält einen Eingang **'direct'**, mit dem festgelegt wird, dass direkte Faltung statt Fourier-Transformation verwendet wird.
- Aufgrund der verbesserten Leistung von LabVIEW werden LabVIEW-MathScripts sowohl im **LabVIEW-MathScript-Fenster** als auch im MathScript-Knoten schneller kompiliert.
- Enthält ein VI einen MathScript-Knoten, werden alle Rückgängig-Informationen beim Speichern oder Ausführen des VIs gelöscht.

Weitere Informationen zu LabVIEW MathScript finden Sie im Buch **Grundlagen»Formeln und Gleichungen** auf der Registerkarte **Inhalt** in der *LabVIEW-Hilfe*.

Verwalten von Überlauf für Festkommazahlen

Überlauf kann auftreten, wenn arithmetische Operationen an Festkommawerten durchgeführt oder numerische Daten mithilfe der Funktion "Nach Festkomma" in Festkommatdaten umgewandelt werden. Um festzulegen, ob ein Überlauf stattfindet, können Sie eine Festkommazahl konfigurieren, so dass ein Überlaufstatus eingeschlossen wird. Beim Einschließen eines Überlaufzustands in eine Festkommazahl weist LabVIEW zusätzlichen Speicherplatz zu, um nachzuverfolgen, ob die Fließkommazahl das Ergebnis einer Operation mit Überlauf ist.

Nach dem Konfigurieren einer Festkommazahl zum Einschließen eines Überlaufzustands können Sie eine Überlaufzustands-LED an Festkomma-Elementen und -Konstanten anzeigen. Die LED leuchtet, wenn der Überlaufzustand der Festkommazahl TRUE ist. Sie können den Überlaufzustand einer Festkommazahl auch mit Hilfe der Funktion "Festkommaüberlauf?" bestimmen. Mit den Festkommafunktionen können Sie den Überlaufzustand eines Festkommawerts ändern.

Weitere Informationen zu Festkommazahlen finden Sie unter dem Thema **Grundlagen»Erstellen des Blockdiagramms»Numerische Daten** auf der Registerkarte **Inhalt** in der *LabVIEW-Hilfe*.

LLBs vereinigen (Nicht im Basispaket)

Mit dem Dialogfeld **Zu vereinigende LLBs auswählen** im LabVIEW Professional Development System lassen sich eine LLB und zwei Versionen der Original-LLB zusammenführen und Unterschiede untersuchen.

Wählen Sie **Werkzeuge»Vereinigen»LLBs vereinigen**, um das Dialogfeld **Zu vereinigende LLBs auswählen** zu öffnen. Geben Sie im Feld **Basis-LLB** die Original-LLB an. Geben Sie in den Feldern **Andere LLB** und **Eigene LLB** die beiden zu vereinigenden LLBs an. Klicken Sie auf **Vereinigen**, um die ausgewählten LLBs zu vereinigen und das Dialogfeld **LLBs vereinigen** zu öffnen. Lösen Sie bestehende Konflikte zwischen den Versionen der LLB und klicken Sie anschließend auf **Schließen** und **Speichern**, um die zusammengeführte LLB zu speichern.

Weitere Informationen zum Vereinigen von LLBs erhalten Sie unter **Grundlagen»Entwicklungsrichtlinien»Anleitung»Vereinigen von VIs und LLBs»Vereinigen von LLBs** auf der Registerkarte **Inhalt** in der *LabVIEW-Hilfe*.

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. MATLAB® ist eingetragene Marke der Firma The MathWorks, Inc. Sonstige hierin erwähnte Produkt- und Firmenbezeichnungen sind Marken oder Handelsnamen der jeweiligen Unternehmen. Nähere Informationen über Patente auf Produkte von National Instruments finden Sie unter **Hilfe»Patente** in Ihrer Software, in der Datei `patents.txt` auf Ihrer CD oder unter ni.com/patents. Für Urheberrechtshinweise, Nutzungsbedingungen und rechtliche Hinweise betreffen Komponenten, die in USI genutzt werden (Xerces C++, ICU, und HDF5), siehe `USICopyrights.chm`.