

Hinweise zum Upgrade auf LabVIEW™ 2020

In diesem Dokument wird beschrieben, wie Sie LabVIEW für Windows, macOS und Linux auf LabVIEW 2020 aktualisieren. Bevor Sie auf die neue LabVIEW-Version umsteigen, informieren Sie sich in diesem Dokument zu den folgenden Themen:

- die empfohlene Vorgehensweise für das Upgrade von LabVIEW
- Hinweise zu möglichen Kompatibilitätsproblemen beim Laden von VIs, die mit älteren Versionen von LabVIEW erstellt wurden
- Änderungen und Neuerungen in LabVIEW 2020

Inhalt

Aktualisieren auf LabVIEW 2020.....	1
Überblick über den empfohlenen Upgrade-Prozess.....	2
1. Sichern der VIs und der Computerkonfiguration.....	2
2. Prüfen der aktuellen Funktionsweise der VIs.....	3
3. Installieren von LabVIEW sowie von Zusatzpaketen und Gerätetreibern.....	4
4. Umwandeln der VIs und Beseitigen von Funktionsunterschieden.....	5
Fehlersuche beim Upgrade.....	7
Hinweise zur Kompatibilität beim Upgrade.....	7
Upgrade von LabVIEW 2016.....	7
Upgrade von LabVIEW 2017.....	8
Upgrade von LabVIEW 2018.....	9
Upgrade von LabVIEW 2019.....	12
Änderungen und Neuerungen in LabVIEW 2020.....	13
Erhöhte Programmcode-Flexibilität durch LabVIEW-Schnittstellen.....	13
Verbesserungen an LabVIEW-Webdiensten.....	15
Verbesserungen am Application Builder.....	16
Verbesserungen an der Programmoberfläche.....	17
Verbesserungen am DLL-Importassistenten.....	21
Verbesserungen am Blockdiagramm.....	21
Neue und geänderte VIs und Funktionen.....	24
Neuerungen und Änderungen an Eigenschaften, Methoden und Ereignissen.....	26
Änderungen und Neuerungen in vorherigen LabVIEW-Versionen.....	28

Aktualisieren auf LabVIEW 2020

Zum Umstellen kleinerer Anwendungen auf eine neue LabVIEW-Version genügt es in der Regel, die VIs in der neuen LabVIEW-Version zu laden. Um durch die Umstellung verursachte Probleme so effizient wie möglich zu erkennen und zu beheben, wird jedoch eine systematischere Vorgehensweise empfohlen.



Tip Große LabVIEW-Anwendungen zur Steuerung und Überwachung von kritischen Operationen sowie Anwendungen, deren Ausfall kostspielig ist, profitieren insbesondere von der systematischen Umstellung. Das Gleiche gilt für Anwendungen, die mit mehreren Modulen und Toolkits arbeiten oder die in einer nicht unterstützten LabVIEW-Version gespeichert sind. Informationen zu LabVIEW-Versionen, zu denen Standard-Support erhältlich ist, finden Sie auf der NI-Webseite unter ni.com/info nach Eingabe des Infocodes *lifecycle*.

Überblick über den empfohlenen Upgrade-Prozess

Sichern der VIs und der Computerkonfiguration	Sichern Sie die VIs und die Konfiguration des Entwicklungscomputers, so dass diese bei Problemen mit dem Upgrade wiederhergestellt werden können.
Prüfen der aktuellen Funktionsweise der VIs	Ermitteln Sie die genaue Funktionsweise Ihrer VIs in der bestehenden LabVIEW-Version, um spätere Funktionsunterschiede erkennen zu können.
Installieren von LabVIEW, Zusatzpaketen und Gerätetreibern	Aktualisieren Sie Ihre gesamte NI-Software auf einmal, damit die aktualisierten VIs in der neuen LabVIEW-Version die erforderlichen SubVIs, Paletten und Eigenschaftsknoten finden.
Umwandeln der VIs und Beseitigen von Funktionsunterschieden	Nehmen Sie die Umstellung und das Testen Ihrer VIs direkt nach dem Aktualisieren vor, so dass Sie Funktionsunterschiede gegenüber Ihrer bisherigen Version sofort beseitigen können.



Hinweis Zum Umstellen Ihrer VIs von einer LabVIEW-Version bis einschließlich 5.1 müssen Sie die VIs zuerst auf eine Zwischenversion von LabVIEW aktualisieren. Weitere Informationen zum Upgrade einer älteren LabVIEW-Version erhalten Sie auf der Website ni.com/info nach Eingabe des Infocodes *upgradeOld*.

1. Sichern der VIs und der Computerkonfiguration

Durch Sichern einer Kopie der VIs und – sofern möglich – der Konfiguration des Computers vor dem Upgrade auf LabVIEW 2020 wird gewährleistet, dass Sie den Upgrade-Prozess bei Bedarf erneut beginnen können.

a. Sichern von VIs

Wenn Sie Ihre VIs vor dem LabVIEW-Upgrade sichern, können Sie die alte Version problemlos wiederherstellen. Ohne die Sicherungskopie können Sie die aktualisierten VIs

nicht mehr in älteren LabVIEW-Versionen öffnen. Die VIs müssen dann erst in der neuen Version für eine ältere Version gespeichert werden.

Zum Sichern mehrerer VIs können Sie auch mit einem Versionsverwaltungssystem arbeiten. Auf diese Weise können Sie bei Bedarf die vorherige VI-Version wiederherstellen.

Weitere Informationen zur Verwendung von Versionsverwaltungssystemen mit LabVIEW finden Sie in der *LabVIEW-Hilfe* unter **Grundlagen»Arbeiten mit Projekten und Zielsystemen»Allgemeines»Verwendung der Versionsverwaltung in LabVIEW**.

b. Sichern der Computerkonfiguration

Beim Upgrade von LabVIEW werden versionsübergreifend verwendete gemeinsame Dateien verändert. Das kann in Ausnahmen dazu führen, dass Ihre VIs in älteren LabVIEW-Versionen nicht mehr erwartungsgemäß funktionieren. Die vorherige Version der gemeinsam genutzten Dateien kann nach dem Installieren der neuen LabVIEW-Version nicht ohne Weiteres wiederhergestellt werden. Daher sollten Sie die Konfiguration der NI-Software auf Ihrem Entwicklungscomputer mit einer der nachfolgend aufgeführten Methoden sichern. Das gilt insbesondere, wenn für Ihre bisherige LabVIEW-Version keine Unterstützung mehr angeboten wird oder ein Ausfall Ihrer Anwendungen kostspielig wäre.

- Erstellen Sie ein Abbild der Computerkonfiguration—Erstellen Sie vor dem Umstellen auf die neue Version mit Hilfe von *Disk-Imaging-Software* ein Festplattenabbild des Computers, das die installierte Software sowie sämtliche Dateien und Benutzereinstellungen enthält. Mit dem Festplattenabbild können Sie den Computer nach dem Upgrade wieder in den Originalzustand zurückversetzen.
- Testen Sie den Upgrade-Prozess auf einem Testcomputer—Nehmen Sie den Upgrade-Prozesses zuerst auf einem Testcomputer (üblicherweise einer virtuellen Maschine) vor. Diese Vorgehensweise wird – obwohl sie zeitaufwändiger als das Erstellen eines Festplattenabbilds ist – für alle ausfallkritischen Systeme in Produktionsprozessen dringend empfohlen. Nach dem Beheben eventueller Upgrade-Probleme auf dem Testcomputer können Sie den Produktionscomputer entweder durch den Testcomputer ersetzen oder den Upgrade-Prozess auf dem Produktionscomputer wiederholen.



Tipp Damit VIs auf dem Testcomputer genauso funktionieren wie auf dem später genutzten Computer, sollten beide u. a. hinsichtlich CPU, RAM, Betriebssystem und Softwareversionen ähnlich konfiguriert sein.

2. Prüfen der aktuellen Funktionsweise der VIs

Aufgrund der Weiterentwicklung von LabVIEW kann es gelegentlich vorkommen, dass VIs in LabVIEW 2020 anders arbeiten als in einer Vorgängerversion. Vergleichen Sie daher, wie ein VI in der bisherigen Version von LabVIEW und in der neuen Version von LabVIEW funktioniert. Führen Sie dazu folgende Tests durch:

- Erstellen Sie Massenkompilierungsprotokolle zum Erkennen fehlerhafter VIs.

Eine Massenkompilierung von VIs vor dem Upgrade ist besonders nützlich, wenn mehrere Personen an der VI-Entwicklung beteiligt sind oder wenn Sie vermuten, dass VIs lange nicht mehr kompiliert worden sind. Damit ein Massenkompilierungsprotokoll erstellt wird, aktivieren Sie die Option **Ergebnisse aufzeichnen** im Dialogfeld **Massenkompilierung**. Weitere Informationen dazu finden Sie in der *LabVIEW-Hilfe* unter **Grundlagen»Erstellen von VIs und SubVIs»Anleitung»Speichern von VIs»Massenkompilierung von VIs**.

- Prüfen Sie mit Modultests, ob einzelne VIs ihre vorgesehenen Funktionen erfüllen.
- Prüfen Sie mit Integrationstests, ob SubVIs in einem Projekt oder einer Gruppe erwartungsgemäß zusammenarbeiten.
- Prüfen Sie mit Einsatztests, ob die VIs nach dem Übertragen auf ein Zielsystem (z. B. einen Desktop-Computer oder FPGA-Chip) erwartungsgemäß funktionieren.
- Prüfen Sie mit Leistungstests die CPU-Auslastung, den Speicherbedarf und die Ausführungsgeschwindigkeit des Programmcodes.

Im Fenster **Profil - Leistung und Speicher** finden Sie Schätzwerte für die durchschnittliche Ausführungsgeschwindigkeit der VIs.

- Prüfen Sie mit Stresstests, wie die VIs mit unerwarteten Daten umgehen.



Hinweis Wenn sich VIs nach der Massenkompilierung oder nach Tests geändert haben, sollten Sie zunächst die neuen Versionen der VIs sichern.

Weitere Informationen zum Testen der VIs finden Sie in der *LabVIEW-Hilfe* unter **Grundlagen»Entwicklungsrichtlinien für Anwendungen»Allgemeines»Entwicklung großer Anwendungen»Phasen von Entwicklungsmodellen»Testen von Anwendungen**.

3. Installieren von LabVIEW sowie von Zusatzpaketen und Gerätetreibern

a. Installieren von LabVIEW einschließlich Modulen, Toolkits und Treibern

Beim Umsteigen auf eine neue LabVIEW-Version müssen Sie neben dem LabVIEW Development System auch Module, Toolkits und Treiber für die neue LabVIEW-Version installieren.

b. Kopieren von user.lib-Dateien

Damit alle benutzerdefinierten Elemente und VIs aus der bisherigen LabVIEW-Version auch für die VIs in LabVIEW 2020 verfügbar sind, kopieren Sie das Verzeichnis `labview\user.lib` Ihrer bisherigen Version in das Verzeichnis `labview\user.lib` von LabVIEW 2020.

c. Erneutes Installieren von VI-Paketen

Wenn Sie VI-Pakete in Ihrer bisherigen Version von LabVIEW mit dem VI Package Manager (VIPM) von JKI installiert haben, rufen Sie VIPM auf und installieren Sie alle Pakete erneut mit LabVIEW 2020.

4. Umwandeln der VIs und Beseitigen von Funktionsunterschieden

VIs werden durch eine Massenkompilierung auf LabVIEW 2020 umgestellt. Das dabei erzeugte Fehlerprotokoll enthält Angaben zu fehlerhaften VIs. Werten Sie diese Angaben unter Zuhilfenahme der [Hinweise zur Kompatibilität beim Upgrade](#) aus, um durch das Upgrade verursachte Funktionsunterschiede zu erkennen.



Hinweis Es wird empfohlen, zum Sichern von VIs und zum Nachverfolgen von Änderungen an VIs die Versionsverwaltung in LabVIEW zu verwenden. Auf diese Weise können Sie bei Bedarf die vorherige VI-Version wiederherstellen.

a. Massenkompilierung von VIs in der neuen Version von LabVIEW

Bei der Massenkompilierung werden die VIs für die Verwendung in LabVIEW 2020 umgewandelt und gespeichert. Danach können die VIs jedoch nicht mehr in einer Vorgängerversion von LabVIEW geöffnet werden. Zum Öffnen der VIs in einer Vorgängerversion muss für jedes VI oder Projekt die Option **Datei»Für vorige Version speichern** ausgewählt werden. Daher sollten VIs nur dann massenkompiliert werden, wenn sie auf die neue Version umgestellt werden sollen. Zum Erkennen von aktualisierungsbezogenen Problemen führen Sie eine Massenkompilierung mit Fehlerprotokoll durch, indem Sie im Dialogfeld **Massenkompilierung** die Option **Ergebnisse aufzeichnen** aktivieren.



Hinweis Bei der Massenkompilierung von VIs, die FPGA- oder Real-Time-Ressourcen enthalten, werden die VIs im Dialogfeld **Massenkompilierung** unter Umständen als nicht ausführbar angezeigt. Um die VIs auf Fehler zu prüfen, müssen Sie sie in einem LabVIEW-Projekt mit den erforderlichen FPGA- oder Real-Time-Ressourcen unter dem FPGA- oder Real-Time-Zielsystem öffnen.

Informationen zur Massenkompilierung finden Sie in den folgenden Abschnitten der *LabVIEW-Hilfe*:

- **Grundlagen»Erstellen von VIs und SubVIs»Anleitung»Speichern von VIs»Massenkompilierung von VIs**
- **Grundlagen»Erstellen von VIs und SubVIs»Anleitung»Speichern von VIs»Gängige Fehlermeldungen bei der Massenkompilierung**

b. Korrigieren von Fehlern in VIs

Unterschiede zwischen Ihrer bisherigen LabVIEW-Version und LabVIEW 2020 können gelegentlich zu Fehlern in VIs führen. Zum schnellen Erkennen und Korrigieren fehlerhafter VIs in LabVIEW 2020 gehen Sie wie folgt vor:

1. Um durch das Upgrade verursachte VI-Fehler zu erkennen, vergleichen Sie das im vorherigen Schritt erzeugte Massenkompilierungsprotokoll mit dem Protokoll, das Sie beim Prüfen der aktuellen Funktionsweise der VIs erzeugt haben.
2. Um festzustellen, ob die VI-Fehler durch LabVIEW-Updates verursacht worden sind, lesen Sie die [Hinweise zur Kompatibilität beim Upgrade](#).

c. Erkennen und Korrigieren von Funktionsunterschieden

Obwohl NI erhebliche Anstrengungen zur Wahrung der Rückwärtskompatibilität Ihrer VIs unternimmt, kommt es dennoch aufgrund von Verbesserungen oder Fehlerkorrekturen hin und wieder zu Funktionsunterschieden. Mit den folgenden Tools kann auf einfache Weise ermittelt werden, ob Ihre VIs von Änderungen an der neuen LabVIEW-Version betroffen sind:

- Führen Sie die Upgrade Tests des VI-Analyzers durch—Mit Hilfe dieser Tests werden viele der durch die Aktualisierung verursachten Funktionsunterschiede in einer Vielzahl von VIs auf effiziente Weise erkannt. Führen Sie zum Herunterladen und Verwenden dieser Tests folgende Schritte aus:
 1. Laden Sie die Upgrade Tests des VI-Analyzers für alle Nachfolgeversionen Ihrer aktuellen LabVIEW-Version herunter. Zur Download-Seite für diese Tests gelangen Sie, indem Sie ni.com/info besuchen und den Infocode [analyzevi](#) eingeben.
 2. Klicken Sie zum Starten der Tests auf **Werkzeuge»VI-Analyseprogramm»VIs analysieren** und starten Sie eine neue VI-Analyse-Aufgabe. Zur Analyse eines gesamten Projekts wählen Sie den entsprechenden Menüpunkt aus dem **Projekt-Explorer** aus.
 3. Zum Beheben der Probleme lesen Sie die [Hinweise zur Kompatibilität beim Upgrade](#) für die LabVIEW-Version, in der die Tests fehlgeschlagen sind.
- Lesen Sie die Dokumentation zum Upgrade.
 - [Hinweise zur Kompatibilität beim Upgrade](#)—Listet Änderungen auf, die sich auf die Funktion Ihrer VIs auswirken können. Lesen Sie den Unterabschnitt für jede LabVIEW-Version beginnend bei Ihrer bisherigen Version.



Tipp Zum schnellen Auffinden von Objekten, die im Abschnitt [Hinweise zur Kompatibilität beim Upgrade](#) erwähnt werden, öffnen Sie Ihre aktualisierten VIs und wählen Sie **Bearbeiten»Suchen und Ersetzen** aus.

- Bekannte Probleme in LabVIEW 2020—Führt alle Programmfehler auf, die kurz vor der Veröffentlichung von LabVIEW 2020 erkannt wurden. Zu dieser Liste gelangen Sie, indem Sie ni.com/info besuchen und den Infocode [lv2020ki](#) eingeben. Unter [Upgrade - Behavior Change](#) und [Upgrade - Migration](#) (falls vorhanden) können Sie sehen, welche Behelfslösungen für die einzelnen Probleme vorgeschlagen werden.

- Dokumentation für Module und Toolkits—Listet Upgrade-Probleme bei bestimmten Modulen und Toolkits wie LabVIEW FPGA oder LabVIEW Real-Time auf.
- Readme-Dateien zu Treibern—Enthalten Listen mit möglichen Upgrade-Problemen. Zum Anzeigen der Readme-Datei für Ihren Treiber gehen Sie zu ni.com/manuals.



Tipp Ob Funktionsunterschiede durch ein Treiber-Upgrade oder das LabVIEW-Upgrade auf LabVIEW 2020 verursacht wurden, erfahren Sie durch Testen der VIs in Ihrer bisherigen LabVIEW-Version.

- Führen Sie Ihre eigenen Tests durch—Führen Sie in LabVIEW 2020 die gleichen Tests wie in der bisherigen LabVIEW-Version durch. Wenn Ihnen Funktionsunterschiede auffallen, ermitteln Sie anhand der Upgrade-Dokumentation die Ursache dafür.

Fehlersuche beim Upgrade

Im Dokument `troubleshooting_guide.html`, das im Verzeichnis `labview\manuals` installiert ist, erhalten Sie weitere Hinweise zu folgenden Themen:

- Suchen nach fehlenden Modul- oder Toolkit-Funktionen
- Suchen nach fehlenden SubVIs, Paletten und Eigenschaftsknoten
- Feststellen, warum sich VIs aus älteren Versionen in LabVIEW 2020 nicht öffnen lassen
- Ermitteln der Versionen installierter Softwareprodukte von National Instruments
- Zurücksetzen von VIs auf eine Vorgängerversion von LabVIEW

Hinweise zur Kompatibilität beim Upgrade

In den folgenden Abschnitten werden versionsspezifische Änderungen an LabVIEW beschrieben, die in bestehenden VIs zu Fehlern oder Funktionsunterschieden führen können.

Bekannte Funktionsprobleme, Kompatibilitätsprobleme sowie kurz vor Veröffentlichung von LabVIEW 2020 hinzugefügte Funktionen, die nicht anderweitig dokumentiert sind, werden in der Datei `readme.html` im `labview`-Verzeichnis beschrieben.

Upgrade von LabVIEW 2016

Beim Upgrade von LabVIEW 2016 auf LabVIEW 2020 können die nachfolgend aufgeführten Kompatibilitätsprobleme auftreten.

Änderungen in der Funktionsweise bei Akteur-Framework-VIs

In LabVIEW-Versionen bis 2016 gibt der Unterakteur für einen fehlgeschlagenen Start aufgrund eines Fehlers in der Methode "Init vor Start" eine Fehlermeldung aus und sendet die Nachricht "Letzte Bestätigung" an den aufrufenden Akteur. In LabVIEW 2017 und neueren Versionen gibt der Unterakteur einen Fehler aus, ohne die Meldung "Letzte Bestätigung" an den aufrufenden Akteur zu senden.

Upgrade von LabVIEW 2017

Beim Upgrade von LabVIEW 2017 auf LabVIEW 2020 können die nachfolgend aufgeführten Kompatibilitätsprobleme auftreten.

Rückwärtskompatibilität der LabVIEW-Runtime-Engine

Beginnend mit LabVIEW 2017 sind die LabVIEW-Runtime-Engines rückwärtskompatibel. Binärdateien und VIs aus älteren Versionen von LabVIEW können ohne erneute Massenkompilierung bzw. erneutes Erzeugen der entsprechenden Binärdateien in der aktuellen Version von LabVIEW geladen und ausgeführt werden. Alle in künftigen LabVIEW-Versionen erzeugten Binärdateien und VIs können dann beispielsweise in LabVIEW 2017 ohne vorherige Kompilierung geladen werden. Diese Neuerung betrifft EXE-Dateien, DLLs und komprimierte Projektbibliotheken.

Um die Rückwärtskompatibilität für Binärdateien zu aktivieren, markieren Sie das folgende Auswahlfeld auf der Seite "Fortgeschritten" im jeweiligen Dialogfeld (abhängig von Ihren Build-Spezifikationen):

Build-Spezifikation	Dialogfeld	Auswahlfeld
Eigenständige Anwendung (EXE)	Eigenschaften für Anwendung	Künftigen LabVIEW-Runtime-Versionen das Ausführen dieser Anwendung gestatten
Komprimierte Projektbibliothek	Eigenschaften für komprimierte Bibliothek	Künftigen LabVIEW-Versionen das Laden dieser komprimierten Bibliothek gestatten
DLL	Eigenschaften für DLL	Künftigen LabVIEW-Runtime-Versionen das Laden dieser DLL gestatten

Bei Build-Spezifikationen, die in LabVIEW 2017 oder einer neueren Version erzeugt werden, sind diese Optionen standardmäßig aktiviert. Sie können diese Optionen deaktivieren, wenn Sie eine Build-Spezifikation an eine bestimmte Version von LabVIEW binden möchten. Das Deaktivieren dieser Optionen verhindert Änderungen an den Leistungsprofilen, wodurch unerwartete aus Compiler-Updates resultierende Probleme vermieden werden. Bei Real-Time ist diese Funktion standardmäßig aktiviert, die Optionen werden jedoch nicht angezeigt.

Änderungen in der Funktionsweise bei VI zur Protokollerstellung

In LabVIEW 2018 unterstützen die Protokoll-VIs nicht mehr die Protokollerstellung im Standardformat. Protokolle können nur noch im HTML-, Word- oder Excel-Format erstellt werden. Folgende VIs sind von der Änderung betroffen:

- Einfaches Drucken von VI-Panels oder Dokumentationen—Dieses VI ist veraltet. Es wurde durch das VI "VI-Panel oder Dokumentation drucken" ersetzt.
- Einfaches Textprotokoll—Dieses VI ist veraltet. Es wurde durch das VI "Einfaches Textprotokoll erstellen" ersetzt.
- Protokolltyp ermitteln—Dieses VI ist veraltet. Es wurde durch das VI "Protokolltyp" ersetzt.
- Neues Protokoll—Dieses VI ist veraltet. Es wurde durch das VI "Protokoll erstellen" ersetzt.
- Tabulatorbreite für Protokoll festlegen—Dieses VI ist veraltet.

Veraltete VIs, Funktionen und Knoten

LabVIEW 2018 und neuere LabVIEW-Versionen unterstützen das VI "Zahl nach Enum" nicht. Verwenden Sie stattdessen die Funktion "Typ erzwingen".

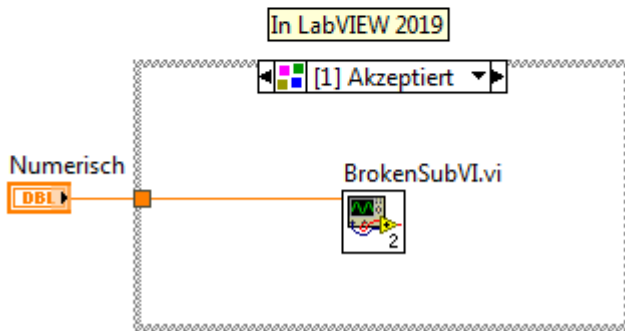
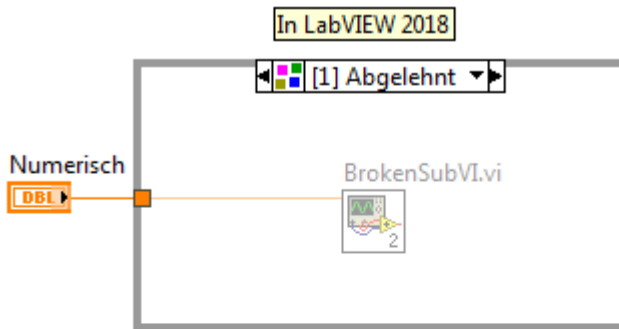
Upgrade von LabVIEW 2018

Beim Upgrade von LabVIEW 2018 auf LabVIEW 2020 kann das nachfolgend aufgeführte Kompatibilitätsproblem auftreten.

- Die Seite "**Paketattribute**" des Dialogfelds "Eigenschaften für Paket" wurde in "**Paket**" umbenannt.
- Die Palette "Array, Matrix & Cluster" wurde in "Daten-Container" umbenannt.
- (Nur Englisch) Die Anschlussnamen der höchsten Ebene der VIs auf der Palette "Datentyp-Parsing" haben sich von Großbuchstaben in Kleinbuchstaben geändert. Die Namen der untergeordneten Anschlüsse, z. B. die Namen von Cluster-Elementen, sind unverändert geblieben.

Änderungen an der Funktionsweise der Typspezialisierungsstruktur


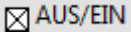
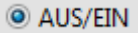
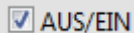
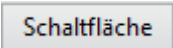
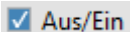
In LabVIEW 2019 hat sich die Syntaxfehlerprüfung der Typspezialisierungsstruktur geändert, mit der entschieden wird, ob ein Unterdiagramm verarbeitet wird. In LabVIEW 2018 und älteren Versionen wurden Fehler in einer Typspezialisierungsstruktur, wie unterbrochene Verbindungen, sowie Fehler in SubVIs oder anderen abhängigen Komponenten als Syntaxfehler betrachtet. In LabVIEW 2019 und neueren Versionen gelten nur noch Fehler innerhalb der Typspezialisierungsstruktur selbst als Syntaxfehler.



Weitere Informationen zur Typspezialisierungsstruktur finden Sie in der *LabVIEW-Hilfe* unter **VI- und Funktionenübersicht»VIs und Funktionen für die Programmierung»Strukturen»Typspezialisierungsstruktur**.

Änderungen beim Erstellen boolescher Bedien- und Anzeigeelemente in der Darstellungsart "Klassisch", "System" und "NXG"

Mit LabVIEW 2019 hat sich die Art und Weise geändert, mit der Bedien- und Anzeigeelemente von booleschen Anschlüssen aus erstellt werden, wenn ein VI für das Erstellen von Bedien- und Anzeigeelementen in der Darstellungsart "Klassisch", "System" oder "NXG" konfiguriert ist. In der folgenden Tabelle wird die Darstellung boolescher Bedien- und Anzeigeelemente, die in LabVIEW 2018 und älteren Versionen erstellt wurden, der Darstellung in LabVIEW 2019 und neuer gegenübergestellt

Darstellung	Typ	LabVIEW 2018 und älter	LabVIEW 2019 und neuer
Klassisch	Bedienelement		
System	Bedien-/ Anzeigeelement		
NXG	Bedienelement		



Hinweis Um die Darstellungsart für das Erstellen von Bedien- und Anzeigeelementen zu konfigurieren, wählen Sie **Datei»VI-Einstellungen**, klicken Sie im Pulldown-Menü **Kategorie** auf **Editor-Optionen** und wählen Sie aus der Liste **Elementdarstellung für "Bedien-/Anzeigeelement erstellen"** die gewünschte Darstellungsart aus.

Die geänderte Funktionsweise betrifft boolesche Bedien- und Anzeigeelemente, die wie folgt erstellt werden:

- Unter Verwendung der Methode "Bedienelement erstellen" oder "Anzeigeelement erstellen"
- Durch Rechtsklick auf einen booleschen Anschluss und Auswahl von **Bedienelement» erstellen** oder **Anzeigeelement» erstellen** aus dem Kontextmenü

Geänderte Funktionsweise bei der automatischen Fehlerbehandlung des Rahmenknotens "Datenwertreferenz lesen / schreiben"

Wenn Sie ein Rahmenknotenpaar des Typs "Datenwertreferenz lesen / schreiben" an einer Inplace-Elementstruktur platzieren, haben sowohl der linke als auch der rechte Rahmenknoten einen Anschluss mit dem Namen **Fehler (Ausgang)**. Wenn in LabVIEW 2018 ein Fehler auftritt und für das VI die automatische Fehlerbehandlung aktiviert ist, zeigt LabVIEW für jeden nicht verbundenen Anschluss des Typs **Fehler (Ausgang)** ein Fehlerdialogfeld an. In LabVIEW 2019 und neueren Versionen zeigt LabVIEW unabhängig davon, wie viele unverbundene Anschlüsse des Typs **Fehler (Ausgang)** vorhanden sind, nur ein Fehlerdialogfeld an.

Weitere Informationen zum Rahmenknoten "Datenwertreferenz lesen / schreiben" finden Sie in der *LabVIEW-Hilfe* unter **VI- und Funktionenübersicht»VIs und Funktionen für die Programmierung»Strukturen»Inplace-Elementstruktur»Datenwertreferenz lesen / schreiben**.

Änderungen beim Aufrufen von Dateien mit öffentlichem Zugriffsbereich in einer passwortgeschützten Bibliothek

Wenn in LabVIEW 2018 ein Freund-VI eine Datei mit öffentlichem Zugriffsbereich in einer passwortgeschützten Bibliothek aufruft, muss beim erstmaligen Bearbeiten oder Ausführen des Freund-VIs das Passwort der Bibliothek angegeben werden. In LabVIEW und neuer können Sie das Freund-VI bearbeiten oder ausführen, ohne das Passwort der Bibliothek angeben zu müssen.

Anzeigen von Textüberlauf in Bedien- und Anzeigeelementen sowie in Konstanten

In LabVIEW 2019 wird per Standardeinstellung durch einen Verblassen-Effekt mit einem Pfeil angezeigt, wenn Text in Strings, numerischen Elementen, Zeitstempeln, Textringen, Enums und Kombinationsfeldern abgeschnitten und demzufolge nicht sichtbar ist. Um die Funktion zum Anzeigen von Textüberlauf in LabVIEW 2018 und älteren Versionen zu aktivieren, klicken Sie mit der rechten Maustaste auf ein Bedien- oder Anzeigeelement oder auf eine Konstante in einem Cluster oder Cluster-Array und wählen Sie **Sichtbare Objekte»Textüberlauf** aus.

Änderungen an den Anschlussnamen der VIs für das Parsen von Datentypen

(Nur Englisch) Die Anschlussnamen der höchsten Ebene der VIs auf der Palette "Datentyp-Parsing" haben sich von Großbuchstaben in Kleinbuchstaben geändert. Die Namen der untergeordneten Anschlüsse, z. B. die Namen von Cluster-Elementen, sind unverändert geblieben.

Weitere Informationen zu den VIs zum Parsen von Datentypen finden Sie in der *LabVIEW-Hilfe* unter **VI- und Funktionenübersicht»VIs und Funktionen für die Programmierung»VIs und Funktionen der Palette "Cluster, Klasse, Variant"»VIs und Funktionen für Variant»Datentyp-Parsing-VIs**.

Upgrade von LabVIEW 2019

Beim Upgrade von LabVIEW 2019 auf LabVIEW 2020 kann das nachfolgend aufgeführte Kompatibilitätsproblem auftreten.

Der MathScript-Funktionsumfang wird nicht mehr empfohlen

LabVIEW MathScript wird für das Entwickeln neuer Projekte nicht mehr empfohlen. Weitere Informationen und Hinweise zu empfohlenen Alternativen finden Sie unter www.ni.com/migratemathscript.

Die SSL-Verschlüsselung ist für Netzwerk-Frontpanels veraltet

In LabVIEW 2020 wird die SSL-Verschlüsselung (Secure Sockets Layer) nicht mehr unterstützt, wenn Sie auf der Seite **Webserver** des Dialogfelds **Optionen** das Kontrollkästchen **SSL** im Abschnitt **Netzwerkpanel-Server** aktivieren. LabVIEW gibt einen Fehler aus, wenn Sie versuchen, eine sichere Verbindung zu einem Netzwerk-Frontpanel-Server herzustellen.


Veraltete VIs, Funktionen und Knoten

Das VI "MD5-Prüfsummendatei" wird in LabVIEW 2020 und Folgeversionen nicht mehr unterstützt. Es wurde durch das VI "Dateiprüfsumme" ersetzt.

Knoten "Elternmethode aufrufen" umbenannt

Der Knoten "Elternmethode aufrufen" heißt fortan "Methode der übergeordneten Klasse aufrufen".

Änderungen und Neuerungen in LabVIEW 2020

Mit dem Symbol  gekennzeichnete Neuerungen stammen von Vorschlägen aus dem [NI Idea Exchange](#) auf [ni.com](#).

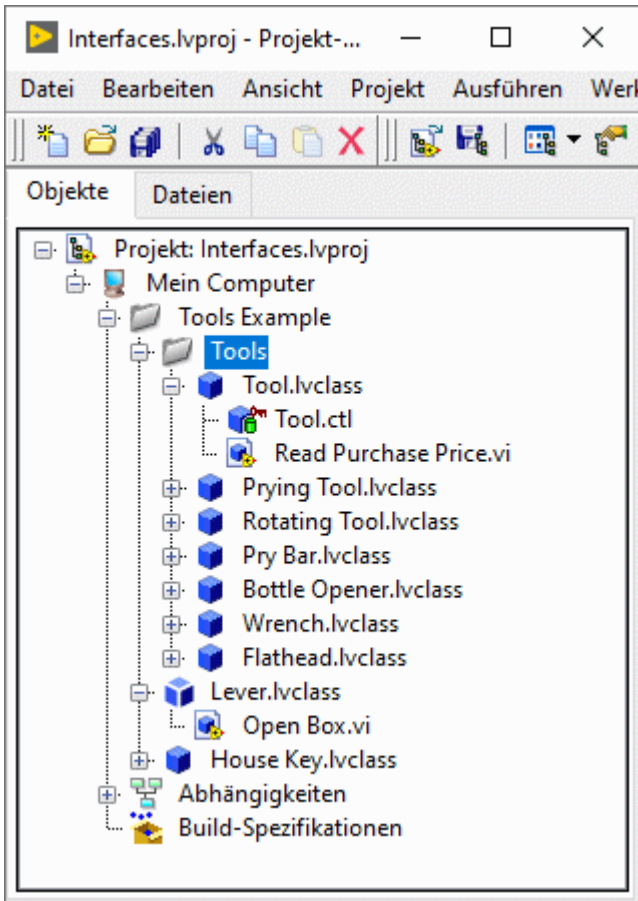
Informationen zu möglichen Problemen beim Upgrade auf bestimmte Versionen von LabVIEW finden Sie im Abschnitt [Hinweise zur Kompatibilität beim Upgrade](#). Bekannte Funktionsprobleme, Kompatibilitätsprobleme, Fehlerkorrekturen sowie kurz vor Veröffentlichung von LabVIEW 2020 hinzugefügte Funktionen werden in der Datei `readme.html` im Verzeichnis `labview\readme` beschrieben.

Erhöhte Programmcode-Flexibilität durch LabVIEW-Schnittstellen

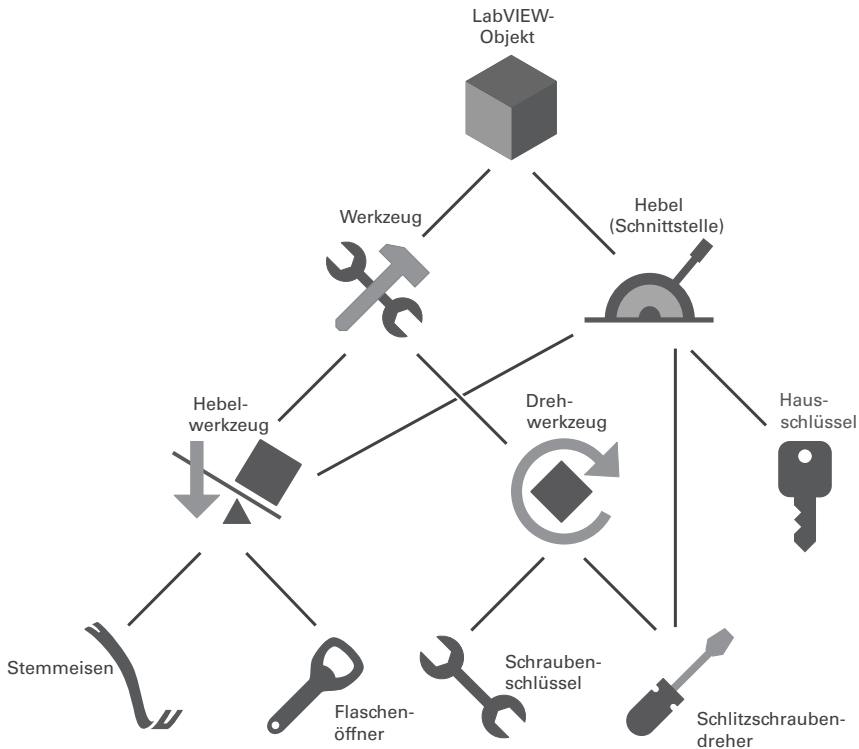
Eine wichtige Neuerung von LabVIEW 2020 sind Schnittstellen. Eine *Schnittstelle* kann als Klasse ohne Privatdatenelement betrachtet werden, wodurch Schnittstellen in Softwarearchitekturen andere Zwecke erfüllen können als Klassen. Das liegt daran, dass Schnittstellen von mehreren übergeordneten Objekten ableitbar sind.

Eine Schnittstelle gibt die Rolle vor, die ein Objekt spielen kann, ohne zu definieren, wie diese Rolle ausgeführt wird. Eine von einer Schnittstelle abgeleitete Klasse gibt ihren Objekten die Rolle vor, die diese zu erfüllen haben, und ist darüber hinaus dafür verantwortlich, die genaue Umsetzung dieser Rolle zu definieren. Wenn eine Klasse von mehreren Schnittstellen abgeleitet ist, können ihre Objekte an mehrere Softwaremodule übergeben werden, die für unterschiedliche Rollen erforderlich sind.

Das folgende Projekt enthält die Klasse "Werkzeug" mit mehreren untergeordneten Klassen für verschiedenartige Werkzeuge. Zu dem Projekt gehört auch eine Schnittstelle mit dem Namen "Hebel". Wie Sie sehen, enthält die Klasse "Werkzeug" ein Bedienelement zum Definieren der Daten der Klasse, das bei der Schnittstelle "Hebel" fehlt. Eine Klasse wird durch einen Würfel dargestellt (🎲). Eine Schnittstelle wird durch die Flächen eines Würfels dargestellt (🎲). Schnittstellen und Klassen haben beide die Dateierweiterung `.lvclass`.



Die folgende Abbildung zeigt die hierarchische Beziehung zwischen der Klasse "Werkzeug", der Klasse "Hausschlüssel" und der Schnittstelle "Hebel". Die Klasse "Werkzeug" und die Schnittstelle "Hebel" sind beide von "LabVIEW-Objekt" abgeleitet. Die Klasse "Hebelwerkzeug" und die Klasse "Schlitzschraubendreher" haben zusätzlich zu ihren eigenen Methoden und den Methoden ihrer übergeordneten Klassen auch Merkmale von der Schnittstelle "Hebel", da beide als Hebel verwendbar sind. Sie enthalten also auch Methoden der Schnittstelle "Hebel". Diese Art der Ableitung ist zulässig, da "Hebel" eine Schnittstelle und keine Klasse ist. Die Klassen "Hebelwerkzeug", "Schlitzschraubendreher" und "Hausschlüssel", die in der Klassenhierarchie nicht miteinander in Beziehung stehen, sind alle von einem übergeordneten Objekt ("Hebel") anstatt von "LabVIEW-Objekt" abgeleitet.



Schnittstellen lassen sich nach einer der folgenden Vorgehensweisen erstellen:

- Klicken Sie mit der rechten Maustaste im **Projekt-Explorer** auf **Mein Computer** und wählen Sie **Neu »Schnittstelle** aus.
- Wählen Sie **Datei »Neu** und klicken Sie in der Liste **Neu erstellen** auf **Andere Dateien » Schnittstelle**.
- Verwenden Sie die Methode "LabVIEW-Klasse:Schnittstelle erstellen".

Unter *Neuerungen und Änderungen an Eigenschaften, Methoden und Ereignissen* finden Sie weitere Informationen zu den neuen Eigenschaften und Methoden für Schnittstellen.

Sehen Sie sich die folgenden Beispielprojekte zur Arbeit mit Schnittstellen an:

- `labview\examples\Object-Oriented Programming\Basic Interfaces\Basic Interfaces.lvproj`
- `labview\examples\Object-Oriented Programming\Actors and Interfaces\Actors and Interfaces.lvproj`

Verbesserungen an LabVIEW-Webdiensten

LabVIEW 2020 umfasst Funktionen zum Entwickeln von LabVIEW-Webdiensten und zum Veröffentlichen von Webdiensten auf dem NI-Webserver sowie dem weiterhin unterstützten Webserver von Anwendungen.

Der NI-Webserver ist ein leistungsfähiger Webserver, auf dem von Benutzern erstellte Dienste (wie LabVIEW-Webdienste) und von NI erstellte SystemLink™-Dienste gehostet werden können. Der NI-Webserver ermöglicht das Verwalten von Geräten, bietet Schutz vor Sicherheitsbedrohungen aus dem Web und lässt sich auf viele in Unternehmen gängige Datendienste erweitern.



Hinweis Der NI-Webserver arbeitet nur mit der 64-Bit-Version von Windows. Der Webserver von Anwendungen arbeitet mit der 32- und der 64-Bit-Version von Windows sowie mit Real-Time-Zielsystemen.

Ein Webdienst kann über eine eigenständige Anwendung, über ein Paket oder ein Paket-Installationsprogramm auf dem NI-Webserver veröffentlicht werden. Zum Sichern der Kommunikation zwischen Webclients und LabVIEW-Webdiensten können Sie auf dem NI-Webserver die SSL-Verschlüsselung aktivieren und jeder Benutzerrolle unterschiedliche Berechtigungen zuweisen.

Als Antwort an einen Webclient wird statt einer XML-Datei ein JSON-String gesendet. Wenn Sie die Daten eines Webdienstes auf einer Benutzeroberfläche darstellen und mit dem Webdienst über einen Webbrowser interagieren möchten, binden Sie Web-VIs in Webdienste ein.

Weitere Informationen zu Web-VIs finden Sie in der [Anleitung zum LabVIEW NXG Web Module](#).

Weitere Informationen zum Entwickeln, Hosten und Veröffentlichen von LabVIEW-Webdiensten finden Sie in der [LabVIEW-Hilfe](#) unter **Grundlagen »Übertragen von Daten » Datenübertragung über ein Netzwerk »Kommunikation zwischen Web-Clients und einer LabVIEW-Anwendung »Überblick: Webbasierte Kommunikation mit einer LabVIEW-Anwendung** .

Verbesserungen am Application Builder

In LabVIEW 2020 wurden folgende Verbesserungen am LabVIEW Application Builder und an den Build-Spezifikationen vorgenommen:

(Windows) Reparieren von Paketen

Bei Problemen mit der Installation eines Pakets können Sie das Paket im NI-Paketmanager reparieren. In LabVIEW können Sie festlegen, welche Reparaturaktion in einem solchen Fall ausgeführt werden soll. Klicken Sie dazu auf der Seite **"Fortgeschritten"** des Dialogfelds **"Eigenschaften für Paket"** mit der rechten Maustaste in die Tabelle **Benutzerdefinierte Aktionen** und wählen Sie **Reparaturaktion hinzufügen** aus. Es stehen folgende Optionen zur Auswahl:

- **Vor Reparatur**—Gibt an, welche Maßnahme vor dem Reparieren des Pakets ausgeführt werden soll.
- **Nach Reparatur**—Gibt an, welche Maßnahme nach dem Reparieren des Pakets ausgeführt werden soll.
- **Nach Reparatur (alle)**—Gibt an, welche Maßnahme nach dem Reparieren aller Pakete ausgeführt werden soll.

Allgemeine Verbesserungen am Application Builder

Die Seite **"Feed"** des Dialogfelds **"Eigenschaften für Paket"** enthält die folgenden neuen Optionen:

- **Alle Abhängigkeiten in Feed einbeziehen**—Gibt an, ob alle abhängigen Komponenten eines Pakets bei der Auswahl des Pakets zum Feed hinzugefügt werden sollen.
- **Paketversion an Feed-Speicherort anhängen**—Gibt an, ob die Paketversion in den Feed aufgenommen werden soll. Wenn ja, wird am Feed-Speicherort ein Unterordner mit der Paketversion erzeugt.
- **Alle Abhängigkeiten in den SystemLink-Feed einbeziehen**—Gibt an, ob alle abhängigen Komponenten eines Pakets oder nur das Paket selbst in den SystemLink-Feed aufgenommen werden sollen.

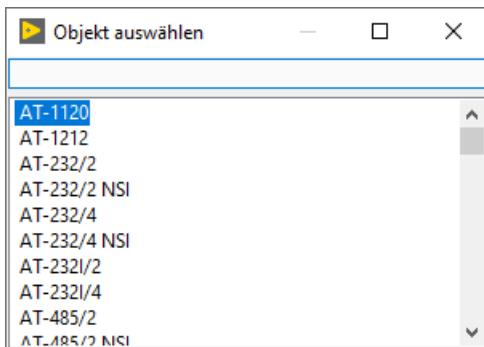
Diese Optionen sind nur unter Windows verfügbar.

Verbesserungen an der Programmoberfläche

In LabVIEW 2020 wurden folgende Verbesserungen an der Programmoberfläche vorgenommen:

Auswahl von Einträgen in Listenelementen

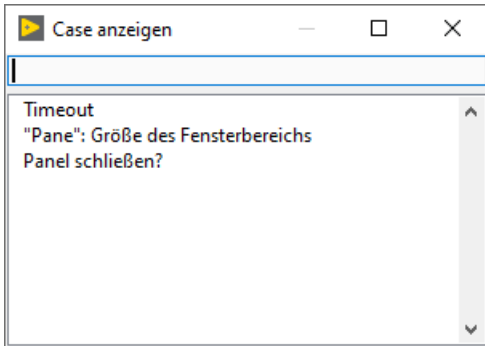
In LabVIEW 2020 können Einträge eines Ring-Bedienelements, einer Enum, eines I/O-Bedienelements oder einer I/O-Konstante im Dialogfeld **"Objekt auswählen"** bearbeitet werden. Klicken Sie das Element bzw. die Konstante auf dem Frontpanel oder das Element bzw. die Konstante auf dem Blockdiagramm mit der rechten Maustaste an und wählen Sie **Objekt auswählen**. Das Dialogfeld **Objekt auswählen** ersetzt den bisherigen gleichnamigen Kontextmenüpunkt.



Verschieben von Objekten zwischen Unterdiagrammen

In LabVIEW 2020 können Sie den sichtbaren Rahmen einer Case-Struktur, Ereignisstruktur oder einer bedingten Deaktivierungsstruktur über das Dialogfeld **"Case anzeigen"** ändern.

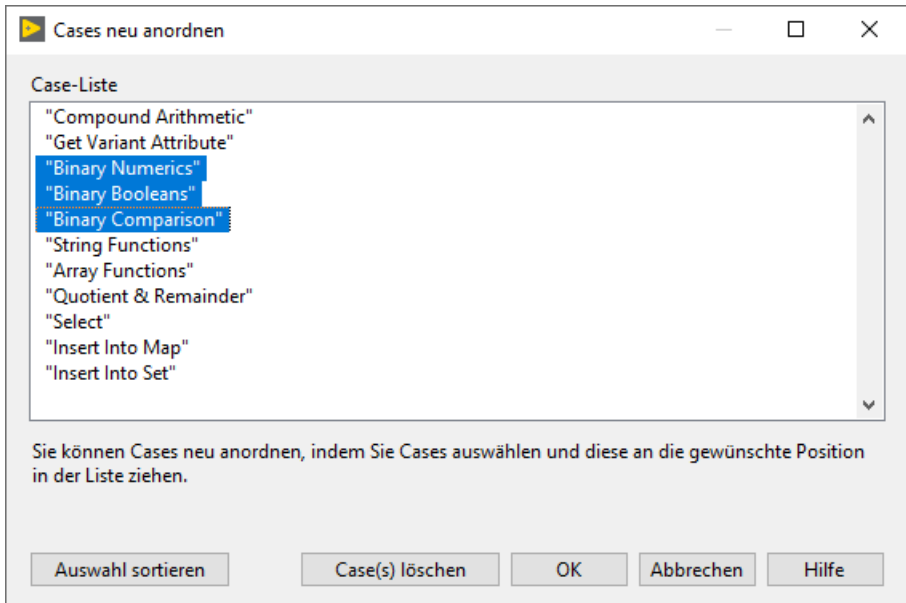
Klicken Sie mit der rechten Maustaste auf den Strukturrahmen und wählen Sie **Case anzeigen** aus. Das Dialogfeld **Case anzeigen** ersetzt den bisherigen gleichnamigen Kontextmenüpunkt.



Neuanordnen von Cases bzw. Unterdiagrammen

Das Dialogfeld **"Cases neu anordnen"** wurde überarbeitet und enthält die folgenden neuen Funktionen:

- Neuanordnen mehrerer Cases
- Löschen eines oder mehrerer Cases
- Sortieren ausgewählter Cases in der Case-Liste
- Ändern der Größe des Dialogfelds



[Idee des NI-Diskussionsforen-Mitglieds "Intaris"]

Löschen von Daten für Variant-Datentypen

Um Daten für einen Variant-Datentyp zu löschen, klicken Sie mit der rechten Maustaste auf das Variant-Bedienelement oder die Variant-Konstante und wählen Sie **Datenoperationen** »**Daten löschen** aus.

[Idee vom NI-Diskussionsforen-Mitglied "altenbach"]

Verbesserungen beim Anzeigen von Fehlern

In LabVIEW 2020 hat sich die Art und Weise geändert, wie ein funktionstüchtiges VI oder eine funktionstüchtige Bibliothek auf fehlerhafte abhängige Komponenten hinweist. In bisherigen LabVIEW-Versionen wird in der **Fehlerliste** ein Fehler bezüglich der unmittelbar abhängigen Komponente des VIs oder der Bibliothek angezeigt. Bei einem Doppelklick auf die Fehlermeldung öffnet sich die betreffende Komponente. In LabVIEW 2020 erscheint der Fehler in der **Fehlerliste** wie folgt: **Ursache des Problems: Die abhängige Komponente ist fehlerhaft**. Bei einem Doppelklick auf diese Meldung wird die abhängige Komponente geöffnet, die für den Fehler verantwortlich ist.

Einblenden des Dialogfelds "Cluster-Größe"

Neben einem Klick mit der rechten Maustaste kann jetzt auch ein Doppelklick auf die Funktion "Array nach Cluster" ausgeführt werden, um zum Dialogfeld "**Cluster-Größe**" zu gelangen.

Anzeigen des letzten Eintrags in einem Ring-Bedienelement

Der letzte Eintrag in einem Ring-Bedienelement lässt sich ausblenden, damit Benutzer diesen während der VI-Ausführung nicht auswählen können. Klicken Sie dazu mit der rechten Maustaste auf das Ring-Bedienelement und wählen Sie **Letzten Menüpunkt beim Ausführen ein-/ausblenden** aus. Die gleiche Funktionsweise lässt sich programmgesteuert mit der Eigenschaft "**Letzter Wert ausgeblendet**" umsetzen.

Verbesserungen an Dialogfeldern

In LabVIEW 2020 wurden die folgenden Konfigurationsdialogfelder für LabVIEW-Webdienste hinzugefügt oder überarbeitet.

- Am Dialogfeld "**Eigenschaften für Webdienst**" gab es folgende Änderungen:
 - **Berechtigungen des NI-Webservers**—Auf dieser neuen Seite können Sie selbst definierte Berechtigungen für den Webdienst hinzufügen.
 - **Einstellungen zum HTTP-Methoden-VI** —Auf der neuen Registerkarte **NI-Webserver** lassen sich die zum Aufrufen des HTTP-Methoden-VIs erforderlichen Berechtigungen festlegen. Die Registerkarte **Sicherheit** wurde in **Webserver von Anwendungen** umbenannt und ermöglicht das Konfigurieren der Berechtigungen für den Webserver von Anwendungen.
- Im Dialogfeld "**Eigenschaften für Paket**" ist eine Seite namens **Webdienste** hinzugekommen. Hier können Sie die für das Einbinden in ein Paket verfügbaren Webdienste konfigurieren.
- Auf der Seite **Webdienste** des Dialogfelds "**Eigenschaften für Anwendung**" ist die Option **NI-Webserver** hinzugekommen. Mit dieser Option können Sie festlegen, dass Webdienste in einer eigenständigen Anwendung auf dem NI-Webserver gehostet werden sollen.
- Auf der Seite **Webserver** des Dialogfelds "**Optionen**" wird in LabVIEW 2020 zwischen dem NI-Webserver und dem Webserver von Anwendungen für das Hosten des Webdienstes unterschieden.

Auf der Seite "**Vererbung**" des Dialogfelds "**Eigenschaften für Klasse**" wurden folgende Änderungen vorgenommen:

- Es wurde ein Abschnitt mit dem Namen **Übergeordnete Schnittstellen** hinzugefügt, in dem die Schnittstellen, von denen die Klasse abgeleitet ist, definiert werden können.
- Der Abschnitt **Klassenhierarchie** wurde in **Hierarchie der übergeordneten Klassen** umbenannt.
- Die Option **Vererbung ändern** wurde in **Übergeordnete Klasse ändern** umbenannt.

Verbesserungen am DLL-Importassistenten


Auf der Seite "**VIs und Elemente konfigurieren**" des DLL-Importassistenten stehen die folgenden neuen Optionen für mehrere Parameter und Funktionen bereit:

- **Auf alle übereinstimmenden Parameter anwenden**—Wendet die Typeinstellungen des aktuellen Parameters auf alle Funktionsparameter an, die der Deklaration in der Header-Datei entsprechen.
- **Auf alle anwenden**—Wendet die Aufrufkonvention der aktuellen Funktion auf alle Funktionen in der DLL an.

Verbesserungen am Blockdiagramm

In LabVIEW 2020 wurden die nachfolgend aufgeführten Verbesserungen am Blockdiagramm vorgenommen.


Löschen ungültiger Verbindungswege

 In bisherigen LabVIEW-Versionen führte ein unterbrochener Verbindungsweig dazu, dass die gesamte Verbindung als ungültig angezeigt wurde, und beim Entfernen des Verbindungsweigs wurde die gesamte Verbindung entfernt. In LabVIEW 2020 wird nur noch der betreffende Verbindungsweig als ungültig angezeigt. Bei Auswahl von **Bearbeiten » Ungültige Verbindungen entfernen** und beim Betätigen von <Strg + B> wird demnach nur der unterbrochene Verbindungsweig entfernt. Die ganze Verbindung wird nur dann gelöscht, wenn alle ihre Verbindungswege ungültig sind.

wenn nur ein Eingang verbunden ist. Klicken Sie dazu bei gedrückter <Strg>-Taste auf einen der beiden Eingänge.

Idee der NI-Diskussionsforen-Mitglieder "altenbach" und "tst"

Festlegen von Symboltext


 In LabVIEW 2020 können Sie mit der **Schnelleinfüge**-Tastenkombination <Strg + K> ein Symbol erstellen und dieses einfach mit Text versehen. Wenn Sie <Strg + Leertaste> und dann <Strg + K> drücken, fügt LabVIEW den Dateinamen des VIs als Text in das Symbol ein. Zu langer Text wird abgeschnitten.

Sie können auch mit der rechten Maustaste auf das VI-Symbol klicken und **Symbol auf VI-Namen festlegen** auswählen.

Sie können den Text anpassen, indem Sie auf <Strg + Leertaste> drücken, den gewünschten Text eingeben und dann <Strg + K> betätigen.


[Idee des NI-Diskussionsforen-Mitglieds "tst"]

Kontexthilfe für Schnelleinfügeobjekte

 In der **Kontexthilfe** werden nun Angaben zum Objekt angezeigt, das aktuell im Dialogfeld "**Schnelleinfügeliste**" ausgewählt ist.

[Idee des NI-Diskussionsforen-Mitglieds "elset191"]

Sinnvolle automatische Verbindungen für mittels Tastenkombination eingefügte Objekte

 Der Algorithmus zum Erkennen und Herstellen der benötigten Verbindungen für Objekte, die über die **Schnelleinfügeliste** auf der ausgewählten Verbindung eingefügt werden, hat sich verbessert. Wenn Sie die **Schnelleinfüge**-Tastenkombination <Strg + I> betätigen, verbindet LabVIEW das Objekt mit den Anschlüssen, die sich dafür am sinnvollsten eignen. Die Verbesserung betrifft folgende Funktionen:

- Binäre numerische Funktionen wie "Addieren", "Subtrahieren", "Multiplizieren" und "Dividieren"
- Binäre boolesche Funktionen wie "UND", "ODER", "Exklusiv-ODER", "Nicht-UND", "Nicht-ODER", "Exklusiv-NICHT-ODER" und "Implikation"
- Binäre Vergleichsfunktionen wie "Gleich?", "Ungleich?", "Größer?", "Kleiner?", "Größer oder gleich?" und "Kleiner oder gleich?"
- Mehrfacharithmetik
- Variantattribut ermitteln
- String-Funktionen wie "Muster suchen", "String suchen und ersetzen" und "String durchsuchen und teilen"
- Array-Funktionen wie "Aus Array löschen" und "In Array einfügen"
- Quotient und Rest
- Auswählen

- In Map einfügen
- In Set einfügen

[Idee des NI-Diskussionsforen-Mitglieds "D"]*

Neue und geänderte VIs und Funktionen

LabVIEW 2020 umfasst folgende neue VIs und Funktionen: Informationen zu VIs, Funktionen und Knoten finden Sie im Buch **VI- und Funktionenübersicht** der *LabVIEW-Hilfe*.

Neue VIs und Funktionen

Webdienst-VIs

Die Palette "Webdienste" wurde überarbeitet und enthält folgende neue Unterpaleetten:

- Webserver von Anwendungen—Enthält VIs für Webdienste, die auf dem Webserver von Anwendungen gehostet werden. Dazu zählen VIs zum Konfigurieren von Skripten für ESP (Embedded Server Pages), zum Ver- und Entschlüsseln übertragener Daten sowie zum Verwalten von HTTP-Sessions auf dem Webserver.
- NI-Webserver—Enthält das neue VI "Authentifizierungsdetails für NI-Webserver abrufen" zum Abfragen der Authentifizierungsangaben für den NI-Webserver.

WebSocket-VIs

Die Palette "Datenaustausch" enthält einen Link zur Installation von WebSocket-VIs zum Datenaustausch mit Webbenutzeroberflächen mittels Streaming. Klicken Sie in der Palette "Datenaustausch" auf **Zusatzpaket für WebSockets installieren**, um das WebSockets Toolkit mit Hilfe des VI Package Managers (VIPM) von JKI zu installieren. Die WebSocket-VIs werden auf der Palette **Datenaustausch »WebSockets** angezeigt.

[Besonderen Dank an Sam Sharp, den Entwickler des WebSockets-Zusatzpakets.]

Mehrere Fehler (VIs)

Die Palette "Dialog & Benutzeroberfläche" enthält die neue Unterpalette "Mehrere Fehler". Mit den VIs der Palette "Mehrere Fehler" können Sie einen Fehler-Cluster in unterschiedliche Formate umwandeln oder die Attribute eines Fehler-Clusters bearbeiten.

TLS-Funktionen

Zur Palette "TCP" ist eine neue Unterpalette mit dem Namen "Transport Layer Security (TLS)" hinzugekommen. Das TLS-Protokoll stellt eine Schnittstelle ähnlich dem Transmission Control Protocol (TCP) dar, wird jedoch über TCP hinausgehend verschlüsselt und authentifiziert. Die TLS-Unterstützung in LabVIEW beruht auf TCP-Funktionen. LabVIEW arbeitet mit der Version 1.2 von TLS. Die TLS-Funktionen können nicht für FPGA-Zielsysteme verwendet werden.

Verschiedene neue VIs und Funktionen

LabVIEW 2020 umfasst folgende sonstige neue VIs und Funktionen:

- Die folgenden VIs dienen zum Berechnen von Hash-Werten:
 - Mit dem neuen VI "Dateiprüfsumme" auf der Palette "Fortgeschrittene Dateifunktionen" wird der Hash-Wert für den Inhalt einer Datei berechnet.
 - Das neue VI "Byte-Array-Prüfsumme" auf der Palette "Datenbearbeitung" berechnet den Hash-Wert eines Byte-Arrays.

LabVIEW arbeitet mit folgenden Algorithmen zur Berechnung von Hash-Werten:

- SHA-256
 - SHA-224
 - SHA-512
 - SHA-384
 - SHA-512/256
 - SHA-512/224
 - SHA3-224
 - SHA3-256
 - SHA3-384
 - SHA3-512
- Das neue VI "Speicherstatus ermitteln" auf der Palette "Speicherverwendung" gibt die Menge an physischem Speicher aus, die von einem LabVIEW-Prozess verwendet wird. Darüber hinaus ermittelt das VI, wie viel Speicher das gesamte System aktuell benötigt.
 - Das neue VI "NI-GUID erzeugen" auf der Palette "Weitere String-Funktionen" generiert eine GUID-Zeichenfolge (Global Unique Identifier).
 - Das neue VI "Bereich für Typ" auf der Palette "Numerisch" gibt den Mindest- und den Höchstwert des Eingangsdatentyps aus.
 - Das neue VI "Enum nach Enum-Array" auf der Palette "Konvertierung" fasst die Einträge des Enum-Elements in einem Array zusammen.
 - Das neue VI "Übergeordnete Klasse und Member-VI-Info der LV-Klasse ermitteln" auf der Palette "Datentyp-Parsing" ruft Angaben zu übergeordnete Klassen und Member-VIs der in einem Variant gespeicherten LabVIEW-Klasse oder -Schnittstelle ab.

Änderungen an VIs und Funktionen

LabVIEW 2020 umfasst folgende geänderte VIs und Funktionen:

- Der Standarddatentyp des Eingangs **Name oder relativer Pfad** der Funktion "Pfad erstellen" hat sich von "String" in "Pfad" geändert. Beim Erstellen eines Bedienelements oder einer Konstante an diesem Eingang wird ein Pfad-Bedienelement oder eine Pfadkonstante erzeugt. Der Eingang arbeitet jedoch auch noch mit Strings.
- Die Funktionen "Nach spezifischerer Klasse" und "Nach allgemeinerer Klasse" arbeiten in dieser Version auch mit Schnittstellen. Mit Hilfe dieser Funktionen lässt sich eine

Verbindung eines Klassen- oder Schnittstellentyps in einen anderen Klassen- oder Schnittstellentyp umwandeln.

- Das VI "Auf identischen oder abgeleiteten Typ überprüfen" bietet Unterstützung für Schnittstellen. Dieses VI bewahrt nur dann die Funktionstüchtigkeit des Aufrufer-VIs, wenn die Eingangsdaten entweder derselben oder der abgeleiteten Eingangsklasse oder Schnittstelle angehören.
- Der Knoten "Elternmethode aufrufen" heißt fortan "Methode der übergeordneten Klasse aufrufen". Mit Hilfe dieses Knotens lässt sich die am nächsten liegende Vorgängerimplementierung einer Klassenmethode aufrufen.
- Das VI "TDMS-Dateimonitor" und das Dialogfeld **TDMS-Dateimonitor** wurden umgestaltet und sind jetzt intuitiver zu bedienen. Sie können Daten einer *.tdms-Datei anzeigen und Einstellungen zur Darstellung auf einer einzigen Seite ändern, anstatt zwischen mehreren Registerkarten zu wechseln.
- In früheren LabVIEW-Versionen hat die Funktion "Fehler zusammenfassen" immer den ersten Fehler oder die erste Warnung in den Eingangs-Fehler-Clustern ausgegeben. In LabVIEW 2020 dagegen kann die Funktion so konfiguriert werden, dass alle Fehler und Warnungen in einem Fehler-Cluster ausgegeben werden. Klicken Sie dazu mit der rechten Maustaste auf die Funktion und wählen Sie **Alle Fehler speichern** aus.

Neuerungen und Änderungen an Eigenschaften, Methoden und Ereignissen

LabVIEW 2020 umfasst die nachfolgend aufgeführten neuen und geänderten Ereignisse, Eigenschaften und Methoden:

Neue Eigenschaften, Methoden und Ereignisse

LabVIEW 2020 umfasst die nachfolgend aufgeführten neuen Ereignisse, Eigenschaften und Methoden:

- Zur Klasse "LV-Klassenbibliothek" sind die folgenden Eigenschaften und Methoden hinzugekommen:
 - Übergeordnete Schnittstellen
 - Übergeordnete Bibliotheken
 - Ist Schnittstelle
 - Übergeordnete Schnittstellen schreiben
 - Übergeordnete Bibliothekspfade schreiben
 - Übergeordnete Schnittstelle hinzufügen
 - Übergeordnete Schnittstelle entfernen
- Zur Klasse "Anwendung" sind die folgenden Eigenschaften und die folgende Methode hinzugekommen:
 - Paletten:Elementennamen
 - Paletten:Funktionsnamen
 - LabVIEW-Klasse:Schnittstelle erstellen

- Zur Klasse "VI" sind die folgende Eigenschaft und die folgende Methode hinzugekommen:
 - Ausführung:Verbindungswerte speichern
 - Änderungen an Element-VI übernehmen
- Zur Klasse "GObject" sind folgende Eigenschaften hinzugekommen:
 - Gruppiert
 - Gruppenmitglied-Refs[]
 - Gesperrt
- Zu den Klassen "Array" und "Array-Konstante" ist folgende Eigenschaft hinzugekommen:
 - Index-Rechteck
- Zur Klasse "Pfad" ist folgende Eigenschaft hinzugekommen:
 - Suchschaltflächen-Rechteck
- Zur Klasse "Ring" sind folgende Eigenschaften hinzugekommen:
 - Letzter Wert ausgeblendet
 - Variante
- Zur Klasse "Baumstrukturelement" ist folgende Eigenschaft hinzugekommen:
 - Fokus-Objekt
- Zur Klasse "MehrspaltigesListenfeld" sind folgende Eigenschaften hinzugekommen:
 - Ausgewählte Zelle
 - Zellengruppen:Schriftfarben
 - Zellengruppen:Hintergrundfarben
- Zur Klasse "Projekt" ist folgende Methode hinzugekommen:
 - Objekt von Objekt-ID
- Zu den Klassen "Numerisch", "NumerischeKonstante" und "ÜberschreibbarerParameteranschluss" sind Eigenschaften zum Konfigurieren der Festkomma-Funktionsweise für numerische Elemente und Konstanten sowie zum Konfigurieren der Ausgabe von Festkommawerten bei mathematischen Grundelementen hinzugekommen.
- Zu den Klassen "Ring", "Enum" und "Kombinationsfeld" sind folgende Ereignisse hinzugekommen:
 - "Ausführen"-Menü geöffnet
 - "Ausführen"-Menü geschlossen
- Zur Klasse "Element" ist folgendes Ereignis hinzugekommen:
 - Kontextmenü verworfen

Änderungen an Methoden

In LabVIEW 2020 wurden Änderungen an den folgenden Methoden vorgenommen:

- Der Parameter **Elternklassen berücksichtigen?** der Methode "LV-Klassenbibliothek:Enthält Implementierung?" wurde in **Übergeordnete Objekte**

berücksichtigen? umbenannt. Wenn mehrere gleichnamige Methoden der Klasse von übergeordneten Schnittstellen abgeleitet sind und die Klasse keine eigene Überschreibmethode bereitstellt, gibt diese Methode einen Fehler aus.

- Zur Methode "Bibliothek:Quellbereich:Ermitteln und Weiterleiten" ist der Parameter **Aufforderung überspringen?** hinzugekommen. Mit diesem Parameter wird angegeben, ob bei einem VI mit dynamischer Bindung zuerst eine Bestätigung angefordert werden soll, bevor die Einstellung für den Zugriffsbereich auf alle Überschreib-VIs angewendet wird.

Änderungen und Neuerungen in vorherigen LabVIEW-Versionen

Alle seit Ihrer bestehenden LabVIEW-Version implementierten Änderungen und Neuerungen sind in den Upgrade-Hinweisen der jeweiligen Versionen aufgeführt. Zum Öffnen dieser Dokumente besuchen Sie die Seite ni.com/info und geben Sie einen der folgenden Infocodes ein:

- Hinweise zum Upgrade auf LabVIEW 2016—[upnote16de](#)
- Hinweise zum Upgrade auf LabVIEW 2017—[upnote17de](#)
- Hinweise zum Upgrade auf LabVIEW 2018—[upnote18de](#)
- Hinweise zum Upgrade auf LabVIEW 2019—[upnote19de](#)

Änderungen vorbehalten. Weitere Informationen zu Marken von NI finden Sie in den *NI Trademarks and Logo Guidelines* auf ni.com/trademarks. Sonstige hierin erwähnte Produkt- und Firmenbezeichnungen sind Marken oder Handelsnamen der jeweiligen Unternehmen. Nähere Informationen über den Patentschutz von Produkten/Technologien von NI finden Sie unter **Hilfe»Patente** in Ihrer Software, in der Datei `patents.txt` auf Ihrem Datenträger oder unter *National Instruments Patent Notice* auf der Website ni.com/patents. Informationen zu Endbenutzer-Lizenzverträgen (EULAs) und Rechtshinweisen von Drittanbietern finden Sie in der Readme zu Ihrem NI-Produkt. Informationen zu den Global-Trade-Compliance-Richtlinien von NI sowie zu Bezugsquellen für relevante HTS-Codes, ECCNs und andere Import-/Exportangaben finden Sie auf ni.com/legal/export-compliance unter der Überschrift *Export Compliance Information*. NI übernimmt weder ausdrücklich noch stillschweigend irgendeine Gewährleistung für die Richtigkeit oder Vollständigkeit der in diesem Dokument enthaltenen Informationen und haftet nicht für Fehler. Kunden aus US-Regierungsbehörden: Die in diesem Dokument enthaltenen Informationen wurden auf private Kosten entwickelt und unterliegen den anwendbaren beschränkten Rechten und beschränkten Datenrechten, die in FAR 52.227-14, DFAR 252.227-7014 und DFAR 252.227-7015 dargelegt sind.

© 1998—2020 National Instruments. Alle Rechte vorbehalten.