

LabVIEW™-Benutzerhandbuch zum Application Builder

Version 7.1

Der LabVIEW Application Builder ist ein LabVIEW-Zusatzpaket zur Erstellung lauffähiger Programme und Shared Librarys (DLLs), die ohne die LabVIEW-Entwicklungsumgebung an Dritte weitergegeben werden können. Die dafür geltenden Lizenzbestimmungen werden in der *LabVIEW-Softwarelizenzvereinbarung* auf der CD mit dem LabVIEW Professional Development System und der Application-Builder-CD erläutert.

In der vorliegenden Broschüre finden Sie eine Anleitung zur Installation des Application Builders. Außerdem werden die Systemvoraussetzungen für die Verwendung der erstellten Anwendungen und Unterschiede zu vorherigen Versionen des Programms beschrieben. Ebenfalls enthalten sind Warnungen und Hinweise, die bei der Umwandlung von VIs in lauffähige Anwendungen und Shared Librarys zu beachten sind.

Um mit dem Application Builder 7.1 arbeiten zu können, muss LabVIEW 7.1 installiert sein.

Weitere Informationen ...

Die *LabVIEW-Hilfe* enthält schrittweise Anleitungen zur Erstellung von ausführbaren Dateien und Shared Librarys sowie Hinweise dazu, was bei der Konvertierung zu beachten ist.

Inhaltsverzeichnis

Systemvoraussetzungen	2
Mac OS	2
UNIX	2
Installation des Application Builders	3
Windows	3

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

Juni 2004
323592B-01

Mac OS	3
UNIX	4
Überprüfen der Installation.....	4
Warnungen und Hinweise zum Erstellen von Programmen und Shared Librarys.....	5
Änderungen zu früheren Versionen.....	7
Die Versionen 7.0 und 7.1 betreffende Änderungen	7
Die Versionen 6.1 und 7.0 betreffende Änderungen	8
Die Versionen 6.0 und 6.1 betreffende Änderungen	8

Systemvoraussetzungen

Für die ausführbaren Dateien und Shared Librarys, die mit dem Application Builder erstellt werden, gelten in etwa dieselben Systemvoraussetzungen wie für die LabVIEW-Entwicklungsumgebung. Lesen Sie dazu bitte in den *Versionshinweisen zu LabVIEW 7.1* nach. Wie viel Speicher benötigt wird, richtet sich nach der Größe der erzeugten Datei. In der Regel benötigen ausführbare Dateien und Shared Librarys etwa soviel Speicherplatz wie VIs in LabVIEW.

Für in LabVIEW erstellte ausführbare Dateien und Shared Librarys wird ein Verzeichnis für temporäre Dateien verwendet. Da einige der temporären Dateien sehr groß sein können, sollte dieses Verzeichnis nach Möglichkeit mehrere MB Speicherplatz umfassen. Der Verzeichnispfad kann nach einem Klick auf **Werkzeuge»Optionen** unter **Pfade** eingesehen werden. Bei Bedarf können Sie auch einen anderen Pfad angeben.



Hinweis Wenn eine Applikation oder Shared Library/DLL unerwartet beendet wird, können Dateien im Verzeichnis für die temporären Dateien zurückbleiben. Löschen Sie daher gelegentlich die temporären Dateien, um wieder Speicherplatz freizugeben.

Mac OS

Um eine Shared Library zu erstellen, muss mindestens die August-2002-Version der Entwicklungswerkzeuge einschließlich des Project Builders installiert sein. In der Apple Developer Connection unter developer.apple.com kann die geeignete Version heruntergeladen werden.

UNIX

Für lauffähige in LabVIEW erstellte Anwendungen und Shared Librarys ist ein X-Window-System-Server erforderlich, wie zum Beispiel Open Windows, CDE oder X11R6. Es wird keine spezielle grafische Benutzeroberfläche wie Motif oder OpenLook benötigt, da LabVIEW mit `xlib` eine eigene Oberfläche erzeugt.

(Sun) Der Application Builder für Solaris läuft nur unter Solaris 7 oder einer neueren Version.

(Linux) Für den Application Builder für Linux wird ein Intel x86 mit Kernel-Version 2.0.x, 2.2.x oder 2.4.x und einer GNU-C-Laufzeitbibliothek ab 2.1.92 (glibc) benötigt.

Installation des Application Builders

Die Standardinstallation des LabVIEW Professional Development Systems umfasst auch den Application Builder. Folgende Schritte sind zur Installation des Application Builders durchzuführen, wenn dieser einzeln erworben wurde.



Hinweis Einige Virenschutzprogramme können den Installationsvorgang verlangsamen. Deaktivieren Sie daher die automatische Virenerkennung. Untersuchen Sie die Festplatte nach Abschluss der Installation auf Viren und aktivieren Sie die automatische Virenerkennung wieder.

Windows

1. Legen Sie die Installations-CD ein.
2. Starten Sie die `setup.exe`.
3. Ändern Sie gegebenenfalls den Pfad zum LabVIEW-Verzeichnis und klicken Sie anschließend auf die Schaltfläche **Installieren**.
4. Überprüfen Sie die Installation gemäß Abschnitt *Überprüfen der Installation* der vorliegenden Broschüre.

Mac OS

1. Legen Sie die Installations-CD ein.
2. Klicken Sie das Symbol **LabVIEW AppLibs Installer** doppelt an.
3. Klicken Sie auf die Schaltfläche **Install** und wählen Sie ein Zielverzeichnis aus. Wählen Sie das LabVIEW-Verzeichnis aus.
4. Überprüfen Sie die Installation gemäß Abschnitt *Überprüfen der Installation* der vorliegenden Broschüre.

UNIX

Folgende Schritte sind notwendig, um den Application Builder unter Sun oder Linux zu installieren. Um diese Bibliotheken zu installieren, werden keine Root-Benutzerrechte verlangt. Für das Schreiben in das LabVIEW-Verzeichnis wird allerdings eine Schreibberechtigung benötigt.

1. Mounten Sie die Installations-CD.
2. Geben Sie folgenden UNIX-Befehl ein:
 - **(Solaris)**

```
volcheck  
cd /cdrom /cdrom0/solaris
```

wobei *cdrom* für das Verzeichnis steht, in dem das CD-Laufwerk gemountet wurde.
 - **(Linux)**

```
cd /mnt/cdrom /linux
```

wobei *cdrom* für das Verzeichnis steht, in dem das CD-Laufwerk gemountet wurde.
3. Danach wird das Installationsprogramm durch Eingabe des folgenden Befehls ausgeführt:

```
./INSTALL
```
4. Folgen Sie den Anweisungen auf dem Bildschirm.
5. Überprüfen Sie die Installation wie im nächsten Abschnitt beschrieben.

Überprüfen der Installation

Um zu überprüfen, ob der Application Builder ordnungsgemäß installiert wurde, öffnen Sie LabVIEW und klicken Sie auf **Werkzeuge» Applikation oder "Shared Library" (DLL) erzeugen**. Im LabVIEW-Verzeichnis sollte sich nun ein Verzeichnis mit der Bezeichnung `AppLibs` befinden. Ist das nicht der Fall, muss die Installation des Application Builders wiederholt werden. Folgen Sie dazu der Anleitung im Abschnitt [Installation des Application Builders](#).

Wenn der Application Builder ordnungsgemäß installiert wurde, enthält das Verzeichnis `examples` eine Datei namens `appbuild.llb`.

Warnungen und Hinweise zum Erstellen von Programmen und Shared Libraries

Nachfolgend sind einige Hinweise aufgeführt, die beachtet werden sollten, wenn eine Anwendung oder Shared Library von einem VI erzeugt und anderen Anwendern zur Verfügung gestellt wird:

- Einige der Eigenschaften und Methoden des VI-Servers stehen in der LabVIEW-Runtime-Engine nicht zur Verfügung. Diese Eigenschaften und Methoden sollten daher in den VIs, aus denen eine ausführbare Datei oder Shared Library erstellt werden soll, nicht enthalten sein.
- Überprüfen Sie die Einstellungen im Dialogfeld **VI-Eigenschaften**. So kann das umzuwandelnde VI beispielsweise so konfiguriert werden, dass es nach dem Öffnen automatisch startet, oder dass die Schaltflächen in der Symbolleiste ausgeblendet werden.
- Es wird empfohlen, eine Beschreibung anzufertigen, so dass der Anwender ggf. auf Informationen zur Verwendung des Programms oder der Shared Library zurückgreifen kann. Wie VIs oder Anwendungen dokumentiert werden, wird im Kapitel 5, *Creating Documentation*, der *LabVIEW Development Guidelines* beschrieben.



Hinweis Beachten Sie bitte, dass es aus urheberrechtlichen Gründen nicht gestattet ist, die LabVIEW-Dokumentation zusammen mit einer selbst erstellten Anwendung oder Bibliothek an Dritte weiterzugeben.

- Wenn das VI mit Hilfe von VI Server ein anderes VI dynamisch aufruft, klicken Sie im Dialogfeld **Anwendung oder "Shared Library" (DLL) erzeugen** unter **Quelldateien** auf die Schaltfläche **Dynamisches VI hinzufügen**.
- Überprüfen Sie in diesem Fall auch, ob der Pfad zu den VIs richtig erzeugt wird, da sich die Pfade bei der Umwandlung ändern. So wird beispielsweise aus `muster.vi` in einer Anwendung der Pfad `C:\..\Anwendung.exe\muster.vi`, wobei `C:\..\Anwendung.exe` für den Pfad zur Anwendung und deren Dateinamen steht.
- Wenn das VI mit der Funktion "Aktueller Pfad des VIs" arbeitet, vergewissern Sie sich, dass diese in der ausführbaren Datei bzw. Shared Library einwandfrei funktioniert. In einer Anwendung oder Bibliothek gibt die Funktion "Aktueller Pfad des VIs" den Pfad des VIs in der Anwendungsdatei an und behandelt die Anwendungsdatei wie eine VI-Bibliothek. So würde die Funktion zum Beispiel, wenn ein VI mit dem Namen `muster.vi` in der erstellten Anwendung enthalten sein

soll, den Pfad `C:\...\Anwendung.exe\muster.vi` angeben, wobei `C:\...\Anwendung.exe` für den Pfad der Anwendung und deren Dateinamen steht.

- Wenn alle Frontpanel einer ausführbaren Anwendung geschlossen werden, wird diese automatisch beendet. Das heißt, wenn das ursprüngliche VI so programmiert ist, dass nach dem Schließen des letzten Frontpanels ein bestimmter Programmcode ausgeführt, entfällt dieser Programmteil nach der Umwandlung in eine ausführbare Datei. Verzichten Sie daher bei VIs, die in lauffähige Anwendungen umgewandelt sollen, auf solche Programmteile.
- Wenn es in der lauffähigen Anwendung einen Knoten zum Aufruf über Referenz gibt, ein VI mit einem Eigenschaftsknoten für bestimmte Frontpanel-Einstellungen oder ein Frontpanel für den Benutzer sichtbar ist, darf im Dialogfeld **Build-Einstellungen bearbeiten** unter **Panel entfernen** nicht die Option **Ja** ausgewählt sein. Wenn der Application Builder bei Auswahl der Option **Ja** das Frontpanel entfernt, geben die Knoten zum Aufruf über Referenz und die Eigenschaftsknoten einen Fehler aus. Diese können sich unter Umständen auf das Verhalten der Anwendung auswirken.
- Vergewissern Sie sich, wenn das VI benutzerdefinierte Laufzeitmenüs enthält, dass alle Menüeinträge in der LabVIEW-Runtime-Engine verfügbar sind.
- Erstellen Sie zusätzlich zur Anwendung ggf. eine Konfigurationsdatei (wie zum Beispiel `labview.ini` unter Windows) mit Ihren Einstellungen für die LabVIEW-Arbeitsumgebung.
- Legen Sie nach Möglichkeit ein Informationsdialogfeld (**Über**) mit Angaben zur erstellten Anwendung an.
- Wenn sich auf dem Rechner, auf dem die erstellte Anwendung laufen soll, noch keine LabVIEW-Runtime-Engine installiert ist, muss diese zusammen mit der Anwendung oder Shared Library bereitgestellt werden. (**Windows**) Zum Starten eines Buildvorgangs mit dem Application Builder müssen Sie entweder als Administrator angemeldet sein oder über Administratorrechte verfügen. (**Linux**) Wenn Sie die LabVIEW-Runtime-Engine auf einem System installieren, auf dem eine Anwendung mit Analyse-VIs ausgeführt werden soll, muss das LabVIEW Run-Time Engine Analysis Support Package (`labview71-rte-aal-7.1-1.i386.rpm`) installiert werden. Geben Sie dazu bei entsprechender Aufforderung **Ja** ein.

In der *LabVIEW-Hilfe* sind die hier genannten Hinweise noch einmal ausführlich beschrieben, und Sie finden eine vollständige Liste der VI-Server-Eigenschaften und Methoden, die nicht mit der LabVIEW-Runtime-Engine ausführbar sind.

Änderungen zu früheren Versionen

In den folgenden Abschnitten werden die Änderungen beschrieben, die in die neuesten Application-Builder-Versionen eingeflossen sind.

Für alle Neuerungen, die es zwischen Version 5.0 und 6.0 gab, lesen Sie bitte die *Versionshinweise zum LabVIEW Application Builder*. Das Dokument finden Sie auch online auf unserer Website ni.com/info, wenn Sie die Zeichenkombination `rdabrg` eingeben.

Die Versionen 7.0 und 7.1 betreffende Änderungen

Zwischen Version 7.0 und 7.1 gab es folgende Änderungen:

- Wenn Sie einer Anwendung ein dynamisch zu ladendes VI hinzufügen, ist das Frontpanel standardmäßig immer enthalten, sofern keine Anwendung für ein RT-System erstellt wurde.
- In den erstellten ausführbaren Anwendungen oder Shared Librarys können alle Verbindungen zu Typdefinitionen getrennt und alle ungenutzten Instanzen von polymorphen VIs einschließlich ihrer SubVIs entfernt werden, so dass die Datei möglichst klein wird. Wenn Sie diese Einstellung nicht wünschen, deaktivieren Sie im Dialogfeld **Applikation oder Shared Library (DLL) erzeugen** unter **Applikations-Einstellungen** die Option **Typdefinitionen trennen und nicht verwendete Instanzen polymorpher VIs entfernen**.
- **(Windows oder Linux)** Wenn in der ausführbaren Datei oder Bibliothek Analyse-VIs vorkommen, aktivieren Sie im Dialogfeld **Fortgeschrittene Installer-Einstellungen** die Option **Unterstützung für Analyse-VIs**, damit die dazu erforderlichen Hilfsdateien mitinstalliert werden. Wenn keine Analyse-VIs in ihrer Anwendung enthalten sind, entfernen Sie das Häkchen aus diesem Auswahlfeld.
- **(Windows)** Wenn in der erstellten Anwendung oder Bibliothek Datenspeicher-VIs vorkommen, aktivieren Sie im Dialogfeld **Fortgeschrittene Installer-Einstellungen** die Option **Unterstützung für Datenspeicher-VIs**. Wenn keine Datenspeicher-VIs in ihrer Anwendung enthalten sind, entfernen Sie das Häkchen aus diesem Auswahlfeld, damit die Datei so klein wie möglich gehalten wird.
- **(Windows)** Zum Starten eines Buildvorgangs mit dem Application Builder müssen Sie entweder als Administrator angemeldet sein oder über Administratorrechte verfügen.
- Wenn ein Build-Skript auf einer anderen Plattform geladen wird, als es gespeichert wurde, verwendet der Application Builder im Feld **Zieldateiname** die Standardweiterung der Zielplattform.
- **(Linux)** Wenn Sie die LabVIEW-Runtime-Engine auf einem System installieren, auf dem eine Anwendung mit Analyse-VIs ausgeführt

werden soll, muss das LabVIEW Run-Time Engine Analysis Support Package (labview71-rte-aal-7.1-1.i386.rpm) installiert werden. Geben Sie dazu bei entsprechender Aufforderung **Ja** ein.

Die Versionen 6.1 und 7.0 betreffende Änderungen

Zwischen Version 6.1 und 7.0 gab es folgende Änderungen:

- **(Windows)** Wenn die erstellte *.exe- oder *.dll-Datei mit einem seriellen oder parallelen Port zusammenarbeiten oder Informationen über die Konfiguration von Geräten abrufen soll, aktivieren Sie die entsprechenden Optionen im Dialogfeld **Fortgeschrittene Installer-Einstellungen**.
- **(Windows)** Aktivieren Sie bei installiertem LabVIEW Real-Time Module im Dialogfeld **Fortgeschrittene Installer-Einstellungen** die Option **Unterstützung für LabVIEW RT**.
- **(Windows)** Wenn nach Abschluss der Installation, aber noch vor Beendigung des Installationsprogramms ein Programm gestartet werden soll, ist die Option **Nach Installation starten** zu aktivieren.
- Wenn die Eigenschaft “Applikation:Kommandozeilen-Argumente” genutzt wird, damit die erstellte Datei durch Eingabe eines Befehls in der Kommandozeile gestartet werden kann, und alle Bestandteile der Eingabe als benutzerdefiniert interpretiert werden sollen, aktivieren Sie auf der Registerkarte **Applikations-Einstellungen** die Option **Alle Kommandozeilen-Argumente an Applikation übergeben**. Dadurch entfällt die Eingabe der zwei Bindestriche vor den benutzerdefinierten Parametern.
- Bei Shared Libraries, die in LabVIEW erstellt worden sind, werden in der Header-Datei ANSI-C- anstelle von LabVIEW-Datentypen verwendet.

Die Versionen 6.0 und 6.1 betreffende Änderungen

Zwischen Version 6.0 und 6.1 gab es folgende Änderungen:

- **(Windows)** Das **Ziel**-Dialogfeld besteht jetzt aus zwei getrennten Dialogfeldern. Im ersten Dialogfeld, **Einstellungen für Build-Ziel**, das sich öffnet, wenn Sie auf der Registerkarte **Quelldateien** des Dialogfelds **Applikation oder “Shared Library” (DLL) erzeugen auf Zielverzeichnisse auswählen** klicken, werden die Zielverzeichnisse für die einzelnen Dateien ausgewählt. Das zweite Dialogfeld, **Installer-Eigenschaften**, das sich öffnet, wenn Sie auf der Registerkarte **Installer-Einstellungen** auf **Eigenschaften** klicken, zeigt die Einstellungen des Installationsprogramms zu jeder einzelnen Datei an.
- Nachdem eine Applikation oder Shared Library erzeugt und die entsprechende Skriptdatei gespeichert wurde, kann durch einen Klick auf den Pfeil neben der Schaltfläche **Laden** eine Liste der zuletzt verwendeten Skriptdateien eingesehen werden.

- **(Windows)** LabVIEW erstellt Microsoft-Installationsprogramme (MSI). Es kann zwischen zehn verschiedenen Installationsverzeichnissen ausgewählt werden.
- **(Windows)** Die Schaltflächen **Datenträgergröße** und **Platz auf erster Diskette reservieren (KB)** wurden von der Registerkarte **Installer-Einstellungen** entfernt, da es mit dem Microsoft-Installer nicht möglich ist, eine Datei auf mehrere Datenträger einer bestimmten Größe aufzusplitten.
- **(Windows)** Die Option **Deinstallationsprogramm erstellen** wurde aus dem Dialogfeld **Fortgeschrittene Installer-Einstellungen** entfernt, da der Application Builder nun grundsätzlich ein Deinstallationsprogramm erstellt. Darüber hinaus kann das Installationsprogramm so konfiguriert werden, dass nur bestimmte Teile der LabVIEW-Runtime-Engine installiert werden.
- Für Netzwerk-Panel-Anwendungen ist der NI-Lizenzmanager in das Installationsprogramm zu integrieren. Klicken Sie dazu auf der Registerkarte **Installer-Einstellungen** auf **Erweitert** und aktivieren Sie die Option **Unterstützung für lizenzierte netzwerksteuerbare Panels**. Informationen zum NI-Lizenzmanager finden Sie unter **Start» Programme»National Instruments»NI License Manager**.
- **(Windows)** Die Anzahl der Sprachen, für die Installationsprogramme erstellt werden können, wurde von zwölf auf vier reduziert (Englisch, Französisch, Deutsch und Japanisch).
- Wenn eine ausführbare Datei oder eine Shared Library erzeugt wird, müssen alle dazugehörigen Textdateien und Fehlercodes einschließlich der von Drittherstellern erstellten mitgeliefert werden. Diese Dateien können im Application Builder bei der Erstellung des entsprechenden Installationsprogramms ausgewählt werden. Der Application Builder erstellt auch ein Unterverzeichnis, in das diese Dateien installiert werden.

Wenn ein Installationsprogramm mit einem Programm eines Drittherstellers erstellt wird, ist festzustellen, wo sich die weiterzugebenden Fehlercodes oder Textdateien befinden und die Unterverzeichnisse für die Dateien selbst anzulegen. Die Textdateien mit benutzerdefinierten Fehlercodes befinden sich im Verzeichnis `labview\user.lib\errors`; die Dateien für die Toolkits werden unter `labview\projects\errors` gespeichert. Wenn ein Installationsprogramm für eine Anwendung erstellt wird, sind die Fehlercode-Dateien in das Unterverzeichnis `errors` des Verzeichnisses `user.lib` zu installieren, das sich im Programmverzeichnis befindet. Wenn ein Installationsprogramm für eine Shared Library erstellt wird, müssen die Fehlercodes in das Verzeichnis `National Instruments\shared\errors` installiert werden. In jedem dieser Verzeichnisse kann auch ein Unterverzeichnis für in andere Sprachen übersetzte Fehlercodes enthalten sein.