

LabVIEW™ Benutzerhandbuch Application-Builder

Version 7.0

Der LabVIEW Application-Builder ist ein Zusatzpaket, das dafür gedacht ist, mit LabVIEW selbständig (ohne Entwicklungsumgebung) ausführbare Programme und Shared Libraries (DLLs) zu erstellen. Diese Anwendungen und Shared Libraries können ohne die LabVIEW-Entwicklungsumgebung vertrieben werden. In den *LabVIEW-Softwarelizenzvereinbarung* auf den CDs “LabVIEW Professional Development System” und “Application-Builder”, befinden sich Informationen über Lizenzbedingungen für den Vertrieb von Applikationen bzw. “Shared Libraries”.

Dieses Handbuch enthält eine Anleitung zur Installation des Application-Builders, beschreibt Systemvoraussetzungen für die mit dieser Version erstellten Applikationen und erläutert Unterschiede zu vorherigen Versionen. Ebenfalls enthalten sind Warnungen und Hinweise, die bei der Umwandlung von VIs in ausführbare Anwendungen oder eine Shared Library zu beachten sind.

Um mit dem Application-Builder 7.0 arbeiten zu können, muss LabVIEW 7.0 auf dem System installiert sein.

Weitere Informationen ...

Die *LabVIEW-Hilfe* enthält Schritt-für-Schritt-Anleitungen zur Erstellung einer Applikation bzw. “Shared Library” sowie Warnhinweise, die vor einer solchen Erstellung in Betracht gezogen werden sollten.

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.

Juni 2003
323592A-01

ni.com

© 2003 National Instruments Corp. Alle Rechte vorbehalten. Printed in Ireland.



Inhaltsverzeichnis

Erforderliche Systemkonfiguration	2
Mac OS X	2
Mac OS 9.x oder frühere	3
UNIX	3
Installation des Application-Builder	3
Windows	4
Mac OS	4
UNIX	5
Überprüfen der Application-Builder-Installation	5
Warnungen und Hinweise beim Erstellen von Anwendungen und “Shared Libraries”	6
Änderungen zu früheren Versionen	7
Die Versionen 6.1 und 7.0 betreffende Änderungen	8
Die Versionen 6.0 und 6.1 betreffende Änderungen	9

Erforderliche Systemkonfiguration

Für die Applikationen und DLLs (Shared Libraries), die mit dem Application-Builder erstellt werden, gelten etwa dieselben Anforderungen wie für die LabVIEW-Entwicklungsumgebung. Die Speicheranforderungen sind abhängig von der Größe der Applikation bzw. der DLL. Gewöhnlich benötigen Applikationen und DLLs etwa soviel Speicherplatz, wie für die Ausführung von VIs innerhalb der LabVIEW-Entwicklungsumgebung nötig wird.

Für LabVIEW-Applikationen und -DLLs wird ein Verzeichnis zum Speichern temporärer Dateien verwendet. Einige der temporären Dateien sind so groß, dass empfohlen wird, mehrere MB Festplattenspeicher für dieses temporäre Verzeichnis freizuhalten. Sie können das temporäre Verzeichnis anzeigen und verändern, in dem Sie über **Werkzeuge»Optionen** die Option **Pfade** vom Auswahlménü auswählen.



Hinweis Wenn eine Applikation oder Shared Library/DLL unerwartet abgebrochen wird, können Dateien im Verzeichnis für die temporären Dateien zurückbleiben. Durch gelegentliches Löschen temporärer Dateien wird Festplattenspeicher freigegeben.

Mac OS X

Um eine “Shared Library” zu erstellen, müssen die August 2002-Version (oder frühere) der Entwicklungswerkzeuge inklusive Project-Builder installiert sein. In der Apple-Developer-Connection unter developer.apple.com kann die geeignete Version heruntergeladen werden.

Mac OS 9.x oder frühere

Um eine “Shared Library” zu erstellen, ist es erforderlich, dass ein Macintosh-Programmer’s-Workshop (MPW) inklusive Toolserver installiert ist. MPW muss außerdem in seinem Tools-Verzeichnis einen MrC Compiler enthalten. Der von CodeWarrior installierte MPW verfügt nicht über diesen Compiler. In der Apple-Developer-Connection unter `developer.apple.com` kann die geeignete Version heruntergeladen werden.

Wenn Sie in LabVIEW eine “Shared Library” von einer CodeWarrior-Anwendung aus aufrufen möchten, muss für diese Anwendung genügend Speicherplatz vorhanden sein. Da die Anwendung die ca. 4 MB große LabVIEW Run-Time-Engine nutzt, sollten immer mindestens 5 MB für “Shared Libraries” zur Verfügung stehen. Per Voreinstellung hat der zur Verfügung stehende Speicher eine Größe von 384 KB. Der Fehler, der bei zu wenig Speicherplatz auftritt, gibt keinen Hinweis auf die Ursache des Problems.

UNIX

Für LabVIEW-Applikationen und “Shared Libraries”, mit denen Frontpanels angezeigt werden, ist ein X-Windows System-Server wie OpenWindows, CDE oder X11R6 erforderlich. Für diese Applikationen und Shared Libraries ist keine spezielle grafische Benutzeroberfläche (GUI) wie Motif oder OpenLook erforderlich, weil LabVIEW mit `xlib` eine eigene GUI erzeugt.

(Sun) LabVIEW Application-Builder für Sun läuft unter Solaris 2.6 (oder höher).

(Linux) Der Application-Builder für Linux läuft unter Linux bei Intel x86er-Prozessoren mit Kernel-Version 2.0.x, 2.2.x, oder 2.4.x und GNU C-Laufzeit-Bibliothek 2.1.92 (oder höher) (`glibc`).

Installation des Application-Builder

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

Windows

Folgende Schritte sind zur Installation des Application-Builders durchzuführen.



Hinweis Einige Virenschutzprogramme können den Installationsvorgang behindern. In diesem Fall ist die automatische Virenerkennung zu deaktivieren und der Installer erneut zu starten. Nach Abschluss der Installation sollten sie die Festplatte auf Viren untersuchen und die automatische Virenerkennung wieder aktivieren.

1. Installations-CD einlegen.
2. `setup.exe` ausführen.
3. Gegebenenfalls Pfad zum entsprechenden LabVIEW-Verzeichnis ändern. Anschließend Schaltfläche **Installieren** anklicken.
4. Nachfolgend ist eine Überprüfung der Installation gemäß Abschnitt [Überprüfen der Application-Builder-Installation](#) dieses Dokumentes vorzunehmen.

Mac OS

Folgende Schritte sind zur Installation des Application-Builders durchzuführen.



Hinweis Einige Virenschutzprogramme können den Installationsvorgang behindern. In diesem Fall ist die automatische Virenerkennung zu deaktivieren und das Installationsprogramm erneut zu starten. Nach Abschluss der Installation ist die Festplatte auf Viren zu untersuchen und die automatische Virenerkennung wieder zu aktivieren.

1. Installations-CD einlegen.
2. Symbol **LabVIEW AppLibs Installer** doppelklicken.
3. Nachdem die Schaltfläche **Install** angeklickt wurde, ist ein Zielordner auszuwählen. LabVIEW-Verzeichnisordner auswählen.
4. Nachfolgend ist eine Überprüfung der Installation gemäß Abschnitt [Überprüfen der Application-Builder-Installation](#) dieses Dokumentes vorzunehmen.

UNIX

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

1. Installations-CD mounten.
2. Folgenden UNIX-Befehl (dem Betriebssystem entsprechend) eingeben:
 - **(Linux)**

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

dabei bezeichnet *cdrom* das CD-Laufwerk.
 - **(Solaris 2)**

```
volcheck
```

```
cd /cdrom/cdrom0/solaris2
```

dabei bezeichnet *cdrom* das CD-Laufwerk.
3. Danach wird das Installationsprogramm durch Eingabe des folgenden Befehls ausgeführt:

```
./INSTALL
```
4. Den Anweisungen auf dem Bildschirm ist Folge zu leisten.
5. Nachfolgend ist eine Überprüfung der Installation gemäß Abschnitt *Überprüfen der Application-Builder-Installation* dieses Dokumentes vorzunehmen.

Überprüfen der Application-Builder-Installation

Nachdem der Application-Builder installiert worden ist, muss LabVIEW gestartet werden. Klicken Sie dann auf **Werkzeuge»Applikation oder “Shared Library” (DLL) erzeugen**. Überzeugen Sie sich davon, dass das LabVIEW-Verzeichnis ein `AppLibs`-Verzeichnis enthält. Wenn dieses Verzeichnis nicht vorhanden sein sollte, ist der Application-Builder erneut zu installieren, wie in Abschnitt *Installation des Application-Builder* dieses Dokuments beschrieben.

Wenn die Bibliotheken ordnungsgemäß installiert sind, enthält das Verzeichnis `examples` die Datei `appbuild.llb`.

Warnungen und Hinweise beim Erstellen von Anwendungen und “Shared Libraries”

Die folgende Liste enthält einige Warnungen und Hinweise, die beachtet werden sollten, wenn VIs in Anwendungen oder “Shared Libraries” umgewandelt und danach an Anwender weitergegeben werden sollen.

- Einige der Eigenschaften und Methoden des VI-Servers stehen in der LabVIEW Run-Time-Engine nicht zur Verfügung. Es ist zu vermeiden, diese Eigenschaften und Methoden in den VIs zu verwenden, aus denen eine Anwendung oder “Shared Library” erstellt werden soll.
- Es ist sicherzustellen, dass die richtigen Einstellungen im Dialogfeld **VI-Eigenschaften** vorgenommen wurden. Beispielsweise kann das VI so konfiguriert werden, dass es beim Öffnen ausgeführt wird, oder dass die Schaltflächen der Symbolleiste verborgen werden.
- Es wird empfohlen, zusammen mit der Anwendung oder “Shared Library” eine Beschreibung anzufertigen, mit deren Hilfe der Anwender ggf. auf Informationen zu deren Verwendung zurückgreifen kann. Weitere Informationen zur Erstellung einer Anwender-Dokumentation befinden sich in Kapitel 5, *Creating Documentation* des Handbuchs *LabVIEW Development Guidelines*.



Hinweis Es ist *nicht* zulässig, die Produktdokumentation von LabVIEW zusammen mit der eigenen erstellten Anwendung oder “Shared Library” weiterzugeben. Die Produktdokumentation von LabVIEW steht unter Urheberrecht.

- Wenn das VI andere VIs über “VI Server” oder “Aufruf über Referenz” dynamisch aufruft, müssen diese VIs durch Anklicken der Schaltfläche **Dynamisches VI hinzufügen** im Register **Quelldateien** des Dialogs **Applikation oder “Shared Library” (DLL) erzeugen** hinzugefügt werden.
- Wenn das VI andere VIs mit VI-Server oder dem Knoten “Aufruf über Referenz” dynamisch laden sollte, ist sicherzustellen, dass der Pfad zu den VIs richtig erzeugt wird. Die Pfade zu den VIs verändern sich, wenn die dynamisch geladenen VIs von Anwendungen oder “Shared Libraries” aus aufgerufen werden. Zum Beispiel wird aus `xyz.vi` in einer Anwendung der Pfad `C:\...\Anwendung.exe\xyz.vi`, dabei steht `C:\...\Anwendung.exe` für den Pfad zur Anwendung und deren Dateinamen.
- Wenn das VI die Funktion “Aktueller Pfad des VI’s” benutzen sollte, ist sicherzustellen, dass dann die Funktion in der Anwendung oder “Shared Library” auch einwandfrei arbeitet. In einer Anwendung oder Bibliothek gibt die Funktion “Aktueller Pfad des VI’s” den Pfad des VIs in der Anwendungsdatei an und behandelt die Anwendungsdatei

wie eine VI-Bibliothek. Zum Beispiel gibt die Funktion für `xyz.vi` in einer Anwendung den Pfad `C:\...\Anwendung.exe\xyz.vi` an; dabei steht `C:\...\Anwendung.exe` für den Pfad der Anwendung und deren Dateinamen.

- Wenn alle Frontpanel einer Anwendung geschlossen werden, wird diese nicht weiter ausgeführt. Code des VIs in Ihrer Anwendung, der, nachdem alle Frontpanel geschlossen wurden, ausgeführt würde, wird nicht mehr in dieser Anwendung ausgeführt. Es ist zu vermeiden, Blockdiagrammcode zu schreiben, der erst ausgeführt wird, nachdem das letzte Frontpanel geschlossen wurde.
- Wenn das VI benutzerdefinierte Laufzeit-Menüs enthalten sollte, ist sicherzustellen, dass alle Menüeinträge der Anwendung, die das VI verwendet, in der LabVIEW Run-Time Engine verfügbar sind.
- Es wird empfohlen, zusammen mit der Anwendung eine Konfigurationsdatei zur Verfügung zu stellen, welche die von Ihnen gemachten Einstellung der LabVIEW-Arbeitsumgebung enthält.
- Es sollte außerdem ein **Informations**-Dialogfeld vorgesehen werden, das Informationen über die Anwendung liefert.
- Die LabVIEW Run-Time Engine ist zusammen mit der Anwendung oder “Shared Library” weiterzugeben.

Weitere Informationen zu diesen Warnhinweisen und Empfehlungen zur Erstellung einer Applikation oder “Shared Library”, inklusive einer vollständigen Liste der VI-Server-Eigenschaften und Methoden, die nicht in der LabVIEW-Run-Time-Engine ausführbar sind, befinden sich in der *LabVIEW-Hilfe*.

Änderungen zu früheren Versionen

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

Weitere Informationen über Unterschiede zwischen den Application-Builder-Versionen 5.0 und 6.0 befinden sich in den *LabVIEW Application-Builder-Versionshinweisen*.

Die Versionen 6.1 und 7.0 betreffende Änderungen

In der folgenden Liste werden die Änderungen zwischen Version 6.1 und 7.0 beschrieben:

- Wenn für eine Applikation oder “Shared Library” Unterstützung durch einen seriellen oder parallelen Port oder Hardware-Konfiguration notwendig sein sollte, können diese Einstellungen durch Aktivieren entsprechender Funktionen im Dialogfeld **Installationsprogramm-Einstellungen (Fortgeschritten)** vorgenommen werden.
- Wenn das LabVIEW-Echt-Zeit-Modul installiert sein sollte, kann die Funktion **Unterstützung für LabVIEW RT** im Dialogfeld **Installationsprogramm-Einstellungen (Fortgeschritten)** aktiviert werden, so dass auch dieses Modul ausgeführt werden kann.
- Wenn ein Programm nach Abschluss der Installation, aber noch vor dem Beenden des Installationsprogramms ausgeführt werden soll, ist die Funktion **Auf Ende des Vorgangs warten** zu aktivieren.
- Wenn die Eigenschaft “Applikation:Kommandozeilen-Argumente” zum Lesen benutzerdefinierter Kommandozeilen-Argumente, welche beim Start einer Applikation oder Programmbibliothek übergeben werden, genutzt wird, besteht die Möglichkeit, um sämtliche Argumente als benutzerdefinierte Argumente zu übergeben, die Option **Alle Kommandozeilen-Argumente an Applikation übergeben** der Registerkarte **Applikationseinstellungen** des Dialogfeldes **Applikation oder “Shared Library” (DLL) erzeugen** zu aktivieren. Dadurch entfällt die Eingabe von zwei Bindestrichen vor den benutzerdefinierten Kommandozeilenargumenten.
- Wenn eine Applikation oder “Shared Library” aus einem VI erstellt wird, das benutzerdefinierte Bedienelemente oder Typendefinitionen enthält, werden durch den Application-Builder die Verbindungen zu diesen Bedienelementen und Typendefinitionen getrennt, sodass die Dateien weder in Applikation noch in “Shared Library” aufgenommen werden müssen. Sollte das VI ein polymorphes VI enthalten, entfernt der Application-Builder alle nicht verwendeten Instanzen des polymorphen VIs, wodurch die Applikation oder “Shared Library” verkleinert wird.
- Programmbibliotheken, die in LabVIEW erstellt worden sind, verwenden ANSI-Datentypen anstelle von LabVIEW-Datentypen im “Header”.

Die Versionen 6.0 und 6.1 betreffende Änderungen

In der folgenden Liste werden die Änderungen zwischen Version 6.0 und 6.1 beschrieben:

- Das **Ziel**-Dialogfeld ist in zwei getrennten Dialogfelder aufgeteilt. Der erste Dialog, **Einstellung für Build-Ziel**, der über die Registerkarte **Quell dateien** des Dialogs **Applikation oder "Shared Library" (DLL) erzeugen** aufgerufen werden kann, legt die Zielverzeichnisse der einzelnen Dateien fest. Der zweite Dialog, **Einstellungen für Installer**, der über die Registerkarte **Installer-Einstellungen** des Dialogs **Applikation oder "Shared Library" (DLL) erzeugen** aufgerufen werden kann, legt die Zielverzeichnisse der Dateien der Installer-Einstellungen fest.
- Nachdem eine Applikation oder "Shared Library" erzeugt und die Skriptdatei gespeichert worden ist, kann durch einen Klick auf den Pfeil neben der Schaltfläche **Laden** eine Liste der zuletzt verwendeten Skriptdateien eingesehen werden.
- **(Windows)** LabVIEW erstellt Microsoft-Installer (MSI). Sie können zehn verschiedene Installationsverzeichnisse auswählen.
- **(Windows)** Die Schaltflächen **Datenträgergröße** und **Platz auf erster Diskette reservieren (kB)** wurden aus der Registerkarte **Installer-Einstellungen** des Dialogfeldes **Applikation oder "Shared Library" (DLL) erzeugen** entfernt, weil der Microsoft-Installer (MSI) keine Aufteilung auf Diskette unterstützt.
- **(Windows)** Die Option **Erstelle Deinstaller** wurde aus dem Menü **Installer-Einstellungen (Fortgeschritten)** entfernt, weil der Application-Builder nun immer einen Deinstaller erstellt. Zusätzlich kann der Installer so konfiguriert werden, dass nur die gewünschten Teile der LabVIEW-Run-Time-Engine installiert werden.
- Um die Anwendung über Netzwerk zu steuern, ist der NI-Lizenzmanager in den Installer zu integrieren. Klicken Sie dazu auf **Fortgeschritten** in der Registerkarte **Installer-Einstellungen** des Dialogs **Applikation oder "Shared Library" (DLL) erzeugen** und aktivieren Sie dort die Option **Unterstützung für lizenzierte netzsteuerbare Panels**. Weitere Informationen zum NI-Lizenzmanager befinden sich unter **Start»Programme»National Instruments»NI-Lizenzmanager**.
- **(Windows)** Die Anzahl der Sprachen, für die Installationsprogramme erstellt werden können, wurde von 12 auf 4 herabgesetzt (Englisch, Französisch, Deutsch und Japanisch).
- Wenn eine Anwendung oder Shared Library erstellt wird, müssen alle relevanten benutzerdefinierten oder von Dritten zur Verfügung gestellten Fehlercodedateien zusammen mit der Anwendung oder Programmbibliothek ebenfalls zur Verfügung gestellt werden. Wenn

der Application-Builder eingesetzt wird, um ein Installationsprogramm für eine Anwendung oder eine “Shared Library” zu erstellen, fordert sie der Application-Builder auf, die Fehlercodedateien, die weitergegeben werden sollen, auszuwählen. Der Application-Builder erstellt auch das Installations-Unterverzeichnis für diese Dateien.

Wenn eine Applikation eines Drittherstellers verwendet wird, um einen Installer zu erstellen, müssen Sie feststellen, wo sich die weiterzugebenden Fehlercodedateien befinden und sie müssen die entsprechenden Unterverzeichnisse für die Dateien konfigurieren. Textdateien mit benutzerdefinierten Fehlercodes befinden sich im Verzeichnis `labview\user.lib\errors\`; die Dateien der Dritthersteller werden unter `labview\projects\errors` abgelegt. Wenn ein Installer für eine Applikation erstellt wird, sind die Fehlercodedateien in einem Unterverzeichnis des Applikationsverzeichnisses `user.lib\errors` zu installieren. Wenn ein Installer für eine Programmbibliothek erstellt wird, dann sind die Fehlercodedateien im Verzeichnis `National Instruments\shared\errors` zu installieren. Jedes dieser Verzeichnisse kann auch ein Sprach-Unterverzeichnis mit übersetzten Fehlercodes enthalten.