

LabVIEW™ アプリケーションビルダ ユーザガイド

バージョン 7.0

LabVIEW アプリケーションビルダは、LabVIEW を使用してスタンドアロンアプリケーション (EXE) や共有ライブラリ (DLL) を作成するためのアドオンパッケージです。これらのアプリケーションや共有ライブラリは、LabVIEW 開発システムなしで配布することができます。アプリケーションおよび共有ライブラリの配布の使用許諾条件については、LabVIEW プロフェッショナル開発システムおよびアプリケーションビルダ配布 CD の「ナショナルインスツルメンツソフトウェア使用許諾契約書」を参照してください。

このユーザガイドでは、アプリケーションビルダのインストール手順、このバージョンで作成するアプリケーションのシステム要件、および前のバージョンから変更された点について説明します。また、アプリケーションまたは共有ライブラリ用に VI を組み込む際の注意とアドバイスについても説明します。

アプリケーションビルダ 7.0 は LabVIEW 7.0 とともに使用する必要があります。

詳細については

アプリケーションまたは共有ライブラリの作成方法の詳細な手順や、アプリケーションまたは共有ライブラリ作成前の注意事項については、『LabVIEW ヘルプ』を参照してください。

目次

| | |
|-------------------------|---|
| システム構成要件..... | 2 |
| Mac OS X..... | 2 |
| Mac OS 9.x 以前..... | 2 |
| UNIX..... | 3 |
| アプリケーションビルダのインストール..... | 3 |

LabVIEW™、National Instruments™、NI™、ni.com™ は、National Instruments Corporation の商標です。本書に掲載されている製品および会社名は該当各社の商標または商号です。National Instruments 製品を保護する特許については、ソフトウェアに含まれている特許情報（ヘルプ→特許情報）、CD に含まれている patents.txt ファイル、または ni.com/patents のうち、該当するリソースから参照してください。

2003 年 6 月版
323593A-01

| | |
|--|---|
| Windows..... | 3 |
| Mac OS..... | 4 |
| UNIX..... | 4 |
| アプリケーションビルダのインストールの確認..... | 5 |
| アプリケーションまたは共有ライブラリを構築する際の 注意とアドバイス..... | 5 |
| アプリケーションビルダの変更内容..... | 6 |
| バージョン 6.1 とバージョン 7.0 の間の変更内容..... | 7 |
| バージョン 6.0 とバージョン 6.1 の間の変更内容..... | 7 |

システム構成要件

アプリケーションビルダを使用して作成するアプリケーションや共有ライブラリは、LabVIEW 開発システムとほぼ同じ要件で動作します。必要なメモリ容量はアプリケーションや共有ライブラリのサイズによって異なります。通常、アプリケーションや共有ライブラリは、開発システムで VI を実行するのに必要なメモリ容量と同量のメモリを必要とします。

LabVIEW アプリケーションおよび共有ライブラリは、テンポラリファイルの格納にディレクトリを使用します。テンポラリファイルには大きなものもあるため、ナショナルインストルメントではこのテンポラリディレクトリ用に数メガバイトのディスク領域を確保することをお勧めします。テンポラリディレクトリを表示したり変更したりするには、**ツール→オプション**を選択し、上部にあるプルダウンメニューから**パス**を選択します。



メモ アプリケーションまたは共有ライブラリが予期せずに中断した場合、ファイルはテンポラリディレクトリ内に残ったままになることがあります。テンポラリディレクトリから古いファイルを削除して、ディスク領域を解放するようにしてください。

Mac OS X

共有ライブラリを作成するには、Project Builder など、2002 年 8 月以降のデベロッパツールがインストールされている必要があります。Apple Developer Connection (developer.apple.com) で、正しいバージョンをダウンロードしてください。

Mac OS 9.x 以前

共有ライブラリを作成するには、ToolServer など、Macintosh Programmer's Workshop (MPW) がインストールされている必要があります。また、MPW のツールディレクトリにある MrC コンパイラが必要です。Codewarrior によってインストールされた MPW には、このコンパイラは入っていません。Apple Developer Connection (developer.apple.com) で、正しいバージョンをダウンロードしてください。

CodeWarrior アプリケーションから LabVIEW で作成した共有ライブラリを呼び出すには、そのアプリケーションに十分なメモリが割り当てられている必要があります。アプリケーションで使用する LabVIEW ランタイムエンジンはそれ自体が約 4 MB であるため、作成される共有ライブラリには少なくとも 5 MB を割り当てる必要があります。デフォルトのメモリ量は 384 KB で、割り当てたメモリが不十分な際に発生するエラーには、問題の原因は表示されません。

UNIX

フロントパネルを表示する LabVIEW アプリケーションおよび共有ライブラリには、OpenWindows、CDE、X11R6 などの X Window System サーバが必要です。LabVIEW では xlib を使用して独自のグラフィカルユーザインタフェース (GUI) が作成されるため、アプリケーションおよび共有ライブラリでは Motif や OpenLook などの GUI は不要です。

(Sun) Sun 対応 LabVIEW アプリケーションビルダは Solaris 2.6 以降で動作します。

(Linux) Linux 対応アプリケーションビルダは、カーネルバージョンが 2.0.x、2.2.x、または 2.4.x の Intel x86 プロセッサで、GNU C ランタイムライブラリ 2.1.92 以降 (glibc) の Linux で動作します。

アプリケーションビルダのインストール

LabVIEW プロフェッショナル開発システムをインストールすると、デフォルトでアプリケーションビルダがインストールされます。アプリケーションビルダを別途購入した場合は、以下の手順でインストールしてください。

Windows

アプリケーションビルダをインストールするには以下の手順を実行してください。



メモ

一部のウイルス検出プログラムは、インストーラの実行を妨げる場合があります。インストールの際には、自動ウイルス検出プログラムを無効にしてください。インストールが終了したら、ハードディスクのウイルスをチェックして無効にしたウイルス検出プログラムを有効にします。

1. インストール CD を挿入します。
2. setup.exe を実行します。
3. 必要に応じて、パスを LabVIEW ディレクトリに変更し、**インストール** ボタンをクリックします。
4. このユーザガイドの「[アプリケーションビルダのインストールの確認](#)」のセクションで説明する方法に従って、インストールを確認します。

Mac OS

アプリケーションビルダをインストールするには以下の手順を実行してください。



メモ

一部のウイルス検出プログラムは、インストーラの実行を妨げる場合があります。インストールの際には、自動ウイルス検出プログラムを無効にしてください。インストールが終了したら、ハードディスクのウイルスをチェックして無効にしたウイルス検出プログラムを有効にします。

1. インストール CD を挿入します。
2. **LabVIEW Applibs Installer** アイコンをダブルクリックします。
3. **Install** ボタンをクリックすると、インストーラは保存先フォルダを選択するようプロンプトします。LabVIEW フォルダを選択します。
4. このユーザガイドの「[アプリケーションビルダのインストールの確認](#)」のセクションで説明する方法に従って、インストールを確認します。

UNIX

Linux または Sun にアプリケーションビルダをインストールするには、以下の手順を実行します。これらのライブラリのインストールにルート権限は必要ありませんが、インストールする LabVIEW への書き込みは可能になっている必要があります。

1. インストール CD をマウントします。
2. 各オペレーティングシステムに対応する以下の UNIX コマンドを入力します。
 - **(Linux)**

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

ここで、*cdrom* は CD をマウントしたディレクトリです。
 - **(Solaris 2)**

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

ここで、*cdrom* は CD をマウントしたディレクトリです。
3. 次のコマンドを入力してインストーラを実行します。

```
./INSTALL
```
4. 画面上の指示に従います。
5. このユーザガイドの「[アプリケーションビルダのインストールの確認](#)」のセクションで説明する方法に従って、インストールを確認します。

アプリケーションビルダのインストールの確認

アプリケーションビルダをインストール後、LabVIEW を起動し、**ツール→アプリケーションまたは共有ライブラリ (DLL) を作成**を選択します。LabVIEW ディレクトリに `applibs` ディレクトリが含まれていることを確認します。このディレクトリが存在しない場合は、このユーザガイドの「[アプリケーションビルダのインストール](#)」のセクションを参照して、アプリケーションビルダを再インストールしてください。

ライブラリが正しくインストールされていれば、`examples` ディレクトリには `appbuild.llb` が含まれています。

アプリケーションまたは共有ライブラリを構築する際の注意とアドバイス

以下のリストは、ユーザに配布するアプリケーションまたは共有ライブラリに VI を組み込む際の注意とアドバイスを示します。

- 一部の VI サーバプロパティおよびメソッドは、LabVIEW ランタイムエンジンではサポートされていません。アプリケーションや共有ライブラリに組み込む VI に、そのようなプロパティやメソッドを使用しないでください。
- **VI プロパティ** ダイアログボックスの設定が正しいことを確認します。たとえば、VI を開いたときに実行するようにしたり、ツールバー上のボタンを非表示にするように構成したい場合があります。
- アプリケーションまたは共有ライブラリとともにドキュメントを配布することをお勧めします。そうすることで、ユーザはアプリケーションまたは共有ライブラリの使用に際して必要な情報を手に入れることができます。ユーザドキュメントの作成の詳細については、『[LabVIEW Development Guidelines](#)』マニュアルの Chapter 5 「[Creating Documentation](#)」を参照してください。



メモ 作成したアプリケーションまたは共有ライブラリとともに、LabVIEW の製品ドキュメントを 配布しないでください。LabVIEW 製品ドキュメントは、著作権で保護されています。

- VI サーバまたは「リファレンス呼び出しノード (Call By Reference Node)」を使用して VI が他の VI を動的にロードした場合は、**アプリケーションまたは共有ライブラリ (DLL) を作成** ダイアログボックスの **ソースファイルタブ**にある **ダイナミック VI を追加** ボタンをクリックして、動的にロードされた VI を追加する必要があります。
- VI サーバまたは「リファレンス呼び出しノード」を使用して VI が他の VI を動的にロードした場合は、アプリケーションまたは共有ライブラリが VI のバスを正しく作成していることを確認してください。アプリケーションまたは共有ライブラリに動的にロードされた VI を

追加すると、VI へのパスが変更されます。たとえば、foo.vi をアプリケーションに作成すると、パスは C:\%. .\Application.exe\foo.vi となり、C:\%. .\Application.exe がアプリケーションおよびファイル名のパスとなります。

- VI が「現在の VI のパス (Current VI's Path)」関数を使用した場合は、関数がアプリケーションまたは共有ライブラリで正常に動作することを確認します。アプリケーションまたは共有ライブラリでは、「現在の VI のパス」関数はアプリケーションファイル内の VI のパスを返し、そのアプリケーションファイルを VI ライブラリのように扱います。たとえば、foo.vi をアプリケーションに作成すると、関数は C:\%. .\Application.exe\foo.vi というパスを返します。ここで、C:\%. .\Application.exe はアプリケーションおよびファイル名のパスです。
- アプリケーション内のすべてのフロントパネルを閉じると、アプリケーションが停止します。アプリケーションに組み込んだ VI に、最後のフロントパネルが閉じた後に実行するコードが含まれている場合、このコードはアプリケーションでは実行しません。最後のフロントパネルが閉じた後に実行するブロックダイアグラムコードは書かないようにしてください。
- VI がカスタムランタイムメニューを使用する場合は、VI が使用するすべてのアプリケーションメニュー項目が LabVIEW ランタイムエンジンで使用可能であることを確認します。
- 使用している LabVIEW 作業環境設定を含む環境設定ファイルを、アプリケーションとともに配布することをお勧めします。
- また、アプリケーションに関する情報を表示するための**バージョン情報**ダイアログボックスも作成することをお勧めします。
- アプリケーションまたは共有ライブラリとともに、LabVIEW ランタイムエンジンを配布する必要があります。

LabVIEW ランタイムエンジンでサポートされていない VI サーバのプロパティおよびメソッドの一覧など、アプリケーションまたは共有ライブラリを作成する際の注意とアドバイスの詳細については、『LabVIEW ヘルプ』を参照してください。

アプリケーションビルダの変更内容

以下のセクションでは、アプリケーションビルダでの各バージョンにおける変更点について説明します。

バージョン 5.0 およびバージョン 6.0 間の変更内容の詳細については、『LabVIEW アプリケーションビルダリリースノート』を参照してください。

バージョン 6.1 とバージョン 7.0 の間の変更内容

バージョン 6.1 と 7.0 の間での変更内容は以下のとおりです。

- アプリケーションまたは共有ライブラリで、シリアルまたはパラレルポートのサポートやハードウェア構成情報が必要な場合、**上級インストーラ設定**ダイアログボックスで適切なチェックボックスをオンにすることで、そのサポートおよび情報をインストーラに含めることができます。
- LabVIEW Real-Time モジュールがインストールされている場合は、**上級インストーラ設定**ダイアログボックスの **LabVIEW RT サポート** チェックボックスをオンにすることで、LabVIEW Real-Time モジュールのサポートをインストーラに含めることができます。
- インストール完了後、インストーラの終了前にプログラムを実行するようにするには、**上級インストーラ設定**ダイアログボックスにある**終了まで待機**チェックボックスをオンにします。
- 「アプリケーション: コマンドライン引数」プロパティを使用して、アプリケーションまたは共有ライブラリが起動したときに渡されたユーザ定義コマンドライン引数を読み取る場合、**アプリケーションまたは共有ライブラリ (DLL) を作成のアプリケーション設定**タブにある**すべてのコマンドライン引数をアプリケーションに渡す**チェックボックスをオンにして、すべての引数をユーザ定義引数として渡すことができます。それにより、コマンドラインのユーザ定義引数の前にハイフンを 2 つ入力する必要がなくなります。
- カスタム制御器やタイプ定義を含む VI からアプリケーションまたは共有ライブラリを作成すると、アプリケーションビルダは制御器とタイプ定義の接続を解除するので、アプリケーションまたは共有ライブラリにファイルを含める必要がなくなります。VI に多形性 VI が含まれる場合、アプリケーションビルダは使用していないすべてのインスタンスを多形性 VI から削除して、アプリケーションまたは共有ライブラリのサイズを縮小します。
- LabVIEW で作成した共有ライブラリは、生成するヘッダに LabVIEW データタイプではなく ANSI タイプを使用します。

バージョン 6.0 とバージョン 6.1 の間の変更内容

バージョン 6.0 と 6.1 の間での変更内容は以下のとおりです。

- **保存先**ダイアログボックスには、2 つのダイアログボックスがあります。1 つめのダイアログボックス、**ビルド先設定**は、**アプリケーションまたは共有ライブラリ (DLL) を作成**ダイアログボックスの**ソースファイル**タブから表示することができ、ファイルごとのビルド先設定を表示します。2 つめのダイアログボックス、**インストール先設定**

は、**アプリケーションまたは共有ライブラリ (DLL) を作成**ダイアログボックスの**インストール**タブから表示することができ、ファイルごとのインストール設定を表示します。

- アプリケーションまたは共有ライブラリを作成してビルドスクリプトファイルを保存したら、**ロード**ボタンの隣にある矢印をクリックして、最近使用したビルドスクリプトファイルを表示することができます。
- **(Windows)** LabVIEW は Microsoft Installer (MSI) インストーラを生成します。デフォルトインストールディレクトリは、10 種類の中から選ぶことができるようになりました。
- **(Windows)** Microsoft Installer (MSI) はディスク分割機能をサポートしないため、**アプリケーションまたは共有ライブラリ (DLL) を作成**ダイアログボックスの**インストール設定**タブから**メディアサイズ**ボタンおよび**ディスク 1 に残す空き容量 (kB)** ボタンが削除されています。
- **(Windows)** アプリケーションビルダでは常にアンインストールが作成されるため、**上級インストール設定**ダイアログボックスから**アンインストールを作成**オプションは削除されています。さらに、LabVIEW ランタイムエンジンの必要な部分のみをインストールするようにインストールを構成することもできます。
- アプリケーションをリモートで制御するには、**アプリケーションまたは共有ライブラリ (DLL)** ダイアログボックスの**インストール**タブにある**上級**ボタンをクリックして、**リモートパネルライセンスサポート**チェックボックスをオンにし、インストールに NI License Manager ユーティリティを含める必要があります。NI License Manager に関する詳細については、**スタート→プログラム→National Instruments → NI License Manager** を選択します。
- **(Windows)** インストーラを作成できる言語の数は、12 言語から 4 言語 (英語、フランス語、ドイツ語、日本語) に縮小されました。
- アプリケーションまたは共有ライブラリを開発している場合は、そのアプリケーションまたは共有ライブラリとともに、ユーザ定義またはサードパーティの関連するエラーコードテキストファイルを配布する必要があります。アプリケーションビルダを使用してアプリケーションまたは共有ライブラリ用のインストールを作成する場合、アプリケーションビルダは配布するエラーコードテキストファイルを選択するようプロンプトします。アプリケーションビルダでは、ファイルのインストールサブディレクトリも構成されます。

サードパーティユーティリティを使用してインストールを作成する場合は、配布するエラーコードテキストファイルを見つけて、そのファイル用の正しいインストールサブディレクトリを構成します。ユーザ定義のエラーコードテキストファイルは `labview\user.lib\errors`、サードパーティのファイルは `labview\projects\errors` にあります。アプリケーション用のインストールを作成する場合は、エラーコードテキストファイルをアプリケーションディレクトリの `user.lib\errors` サブディレクトリにインストールします。共有ライブラリ用のインストールを作成する場合は、エラーコードテキスト

ファイルを National Instruments¥shared¥errors ディレクトリにインストールします。それらのディレクトリには、エラーコードを翻訳した言語サブディレクトリを入れることもできます。