

LabVIEW™ アプリケーションビルダ ユーザガイド

バージョン 7.1

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、および旧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 7.1 は LabVIEW 7.1 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詳細な作成手順および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	2
UNIX.....	2
アプリケーションビルダのインストール.....	3
Windows.....	3
Mac OS	3
UNIX.....	4

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。USI (Xerces C++、ICU、および HDF5) で使用されるコンポーネントについては、以下の著作権が適用されます。条件および免責条項については、USICopyrights.chm をご参照ください。

2004 年 5 月版
323593B-01

アプリケーションビルダのインストールの確認.....	4
アプリケーションおよび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.....	7
バージョン 7.0 とバージョン 7.1 の間の変更内容.....	7
バージョン 6.1 とバージョン 7.0 の間の変更内容.....	8
バージョン 6.0 とバージョン 6.1 の間の変更内容.....	9

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。LabVIEW 開発システムの要件に関する詳細は、『LabVIEW 7.1 リリースノート』を参照してください。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、LabVIEW 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、ディレクトリにテンポラリファイルを格納します。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ツ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

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/ja) で、正しいバージョンをダウンロードしてください。

UNIX

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 などの特定の GUI は不要です。

(Sun) Solaris 対応 LabVIEW アプリケーションビルダは Solaris 7 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のデフォルトインストールでは、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、以下の手順に従ってください。



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。このため、インストールを始める前に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すべて有効にします。

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 LabVIEW ディレクトリへのパスを変更し、**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、以下の手順に従ってください。



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。このため、インストールを始める前に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すべて有効にします。

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、インストーラによって保存先フォルダを選択するように促すプロンプトが表示されます。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。

Sun または Linux にアプリケーションビルダをインストールするには、以下の手順に従ってください。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 ディレクトリ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。
 - **(Solaris)**

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

ここで、*cdrom* は CD をマウントしたディレクトリです。
 - **(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。
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. 画面上の指示に従ってください。
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。

アプリケーションビルダ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.lib* が含まれています。

アプリケーションおよび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。そのようなプロパティやメソッドは、アプリケーションや共有ライブラリに組み込む VI には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、スクロールバー、ツールバー上のボタンを非表示にするように VI を構成することができます。
- ユーザが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入手できるように、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。ユーザドキュメントの作成についての詳細は、『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 製品のドキュメントを配布しないでください。LabVIEW 製品のドキュメントは、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 を動的にロードする場合は、**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 を動的にロードする場合は、アプリケーションまたは共有ライブラリが VI のパ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。
- VI が「現在の VI のパス (Current VI's Path)」関数を使用する場合は、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されません。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。
- 「リファレンス呼び出しノード (Call By Reference Node)」を使用してアプリケーションで VI を呼び出す場合、VI が「プロパティノード (Property Node)」を使用してフロントパネルのプロパティを設定する場合、またはユーザにフロントパネルを表示する場合は、**ビルド設定を編集**ダイアログボックスの**パネルを削除**プルダウンメニューでは**はい**を選択しないでください。**はい**を選択すると、LabVIEW でフロントパネルが削除され、フロントパネルを参照する「リファレンス呼び出しノード」と「プロパティノード」は、アプリケーションまたは共有ライブラリの動作に影響を与える可能性があるエラーを返します。
- 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。
- Windows の labview.ini ファイルなど、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検討してください。
- 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を作成することも考慮してください。
- ターゲットコンピュータに LabVIEW ランタイムエンジンがインストールされていない場合は、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。**(Windows)** アプリケーションビルダを使用して作成したインストーラを実行するには、ネットワーク管理者またはネットワーク管理者の権限を持つユーザとしてログオンする必要があります。**(Linux)** 解析 VI を使用してアプリケーションを実行するターゲットコンピュータに LabVIEW ランタイムエンジンをインストールする場合、LabVIEW Run-Time Engine Analysis Support Package (labview71-rte-aal-7.1-1.i386.rpm) をインストールするように促すプロンプトが表示されたら、yes と入力してください。

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。

アプリケーションビルダのバージョン 5.0 とバージョン 6.0 の間の変更内容に関する詳細については、ナショナルインスツルメンツのウェブサイト ni.com/info (英語) または ni.com/jp/info で入力コード rdabrrj を入力して、『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。日本語版のマニュアルは ni.com/jp/manuals にある『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 7.0 とバージョン 7.1 の間の変更内容

バージョン 7.0 と 7.1 の間での変更内容は以下のとおりです。

- 動的にロードした VI をアプリケーションに追加すると、LabVIEW を RT ターゲットの対象にしない限り、アプリケーションビルダにはデフォルトで動的にロードした VI のフロントパネルが含まれます。
- 作成したアプリケーションまたは共有ライブラリから、タイプ定義を接続解除して未使用の多態性 VI のインスタンスおよびそのサブ VI を削除することにより、アプリケーションや共有ライブラリのサイズを減らすことができます。**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アプリケーション設定タブにあるタイプ定義を削除し、未使用の多態性 VI のインスタンスを削除する**チェックボックスからチェックマークを外すと、アプリケーションまたは共有ライブラリにあるタイプ定義および多態性 VI のインスタンスとそのサブ VI がすべてアプリケーションビルダに含まれます。
- (Windows および Linux)** アプリケーションまたは共有ライブラリが解析 VI を使用する場合、**上級インストーラ設定で解析 VI サポート**チェックボックスにチェックマークを付けて、ターゲットコンピュータに VI のサポートをインストールします。アプリケーションまたは共有ライブラリが解析 VI を使用しない場合、**解析 VI サポート**チェックボックスからチェックマークを外してインストーラのサイズを減らします。
- (Windows)** アプリケーションまたは共有ライブラリがストレージ VI を使用する場合、**上級インストーラ設定でストレージ VI サポート**チェックボックスにチェックマークを付けて、ターゲットコンピュータに VI のサポートをインストールします。アプリケーションまたは共有ライブラリがストレージ VI を使用しない場合、**ストレージ VI サポート**チェックボックスからチェックマークを外してインストーラのサイズを減らします。
- (Windows)** アプリケーションビルダを使用して作成したインストーラを実行するには、ネットワーク管理者またはネットワーク管理者の権限を持つユーザとしてログオンする必要があります。

- ビルドスクリプトを保存したのとは異なるプラットフォームにそのビルドスクリプトをロードすると、アプリケーションビルダは、**ターゲットファイル名**テキストボックスの拡張子を、スクリプトをロードするプラットフォームのデフォルトのファイル拡張子に変更します。
- **(Linux)** 解析 VI を使用してアプリケーションを実行するターゲットコンピュータに LabVIEW ランタイムエンジンをインストールする場合、LabVIEW Run-Time Engine Analysis Support Package (labview71-rte-aal-7.1-1.i386.rpm) をインストールするように促すプロンプトが表示されたら、yes と入力してください。

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。

- **(Windows)** アプリケーションまたは共有ライブラリで、シリアルポート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にチェックマークを付けることで、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。
- **(Windows)** 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**チェックボックスにチェックマークを付けることで、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。
- **(Windows)** 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**チェックボックスにチェックマークを付けます。
- 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の起動時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**ダイアログボックスの**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**チェックボックスにチェックマークを付けて、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。これにより、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。
- 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。

- **(Windows) 保存先ダイアログボックス**は、2つのダイアログボックスに分かれています。1つめのダイアログボックスの**ビルド先設定**は、ファイルごとのビルド先設定を示し、**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ます。2つめのダイアログボックスの**インストール先設定**は、ファイルごとのインストール設定を示し、**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**タブから表示することができます。
- 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後、**ロード**ボタンの隣にある矢印をクリックすると、最近使用したビルドスクリプトファイルのリストを表示することができます。
- **(Windows) LabVIEW**では Microsoft Installer (MSI) インストーラが生成されます。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ます。
- **(Windows) Microsoft Installer (MSI)**ではディスク分割機能がサポートされていないため、**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ラが作成されるため、**上級インストール設定**ダイアログボックスから**アンインストールラを作成**オプションが削除されています。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ラ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) を作成**ダイアログボックスの**インストール設定**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**チェックボックスにチェックマークを付け、インストールラに NI License Manager ユーティリティを含める必要があります。NI License Manager に関する詳細は、**スタート→プログラム→National Instruments → NI License Manager** を選択してください。
- **(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(日本語、英語、フランス語、ドイツ語)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に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ラを作成する場合は、アプリケーションビルダが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に促すプロンプトが表示されます。また、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。

サードパーティのユーティリティを使用してインストーラを作成する場合は、配布するエラーコードテキストファイルを検索して、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`、ツールキットのエラーコードテキストファイルは `labview\projects\errors` にあります。アプリケーション用のインストーラを作成する場合は、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ラを作成する場合は、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。こ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。