

# LabVIEW™ アプリケーションビルダ

## バージョン 5.1.1

LabVIEW アプリケーションビルダは、LabVIEW で実行可能なプログラムを作成できるアドオンパッケージです。さらに、作成した実行可能プログラムは、LabVIEW 開発ソフトウェアがなくても配布することができます。実行可能プログラムを配布する際のライセンス条件については、「LabVIEW Software License Agreement」をご覧ください。

このリリースノートでは、インストール手順と、このバージョンのアプリケーションビルダで作成したアプリケーションのシステムの必要条件について説明します。アプリケーションビルダ 5.1.1 は LabVIEW 5.1.1 開発システムでお使いください。

## 目次

---

システム構成の必要条件 .....	2
UNIX .....	2
Concurrent PowerMAX のオペレーティングシステムパッチ .....	3
LabVIEW アプリケーションビルダライブラリのインストール .....	3
Windows .....	3
Macintosh .....	4
UNIX .....	4
Linux および HP-UX (英語版) .....	4
Sun (英語版) .....	5
Concurrent PowerMAX (英語版) .....	5
アプリケーションビルダライブラリのインストールを確認する .....	5
アプリケーションビルダライブラリの変更点 .....	6
バージョン 5.1 から 5.1.1 への変更点 .....	6
バージョン 5.0 から 5.1 への変更点 .....	7
LabVIEW アプリケーションの機能 .....	7
標準機能 .....	7
アプリケーションの作成 .....	8
アプリケーション機能をカスタマイズする .....	10
ビルドに含まれる VI を修正する .....	11
インストーラを作成する (Windows のみ) .....	12

アプリケーションビルド例 .....	13
アプリケーションを配布する .....	14
アプリケーションに必要なその他のファイル .....	14
配布権 .....	16
環境の設定 .....	16
VI 設定オプションを使用して VI オプションを限定する .....	17
ヘルプ情報を提供する .....	18
一般的なエラー .....	18

## システム構成の必要条件

アプリケーションビルダライブラリで作成したアプリケーションに必要な条件は、開発システムとほぼ同じです。必要なメモリ容量はアプリケーションのサイズにより異なりますが、通常は開発システム内の VI の実行に必要なメモリと同じ容量がアプリケーションにも必要です。

LabVIEW アプリケーションは、テンポラリファイルのディレクトリを利用してファイルを保存します。テンポラリファイルには容量が大きいものもありますので、テンポラリディレクトリに数メガバイトのディスク容量を確保しておくことをお勧めします。テンポラリディレクトリの表示や変更は、**編集** » **環境設定** を選択して行います。

**(Sun)** このディレクトリの TMPFS ファイルシステムを使用して、性能を向上させることができます。Solaris 2 は、デフォルトで TMPFS を使用します。

アプリケーションが中断した場合、ファイルがテンポラリディレクトリに残っていることがあります。ディスク容量がいっぱいにならないように、ときおり古いファイルを削除することをお勧めします。

## UNIX

LabVIEW アプリケーションには、OpenWindows 3、HP-VUE、または X11R6 などの X Windows System サーバが必要です。LabVIEW は Xlib を使用して独自の GUI を作成しますので、それらのアプリケーションに Motif や OpenLook などのグラフィカルユーザインタフェース (GUI) は必要ありません。

Sun 対応の LabVIEW アプリケーションビルダは、Solaris 2.4 以上で動作します。HP-UX 対応 LabVIEW は、HP-UX 10.20 以上を搭載する Hewlett-Packard Model 9000 Series 700 コンピュータ上で動作します。また、Linux 対応 LabVIEW は、RedHat Linux 5.0 以上を搭載する Linux/x86 コンピュータ上で動作します。Concurrent PowerMAX 対応 LabVIEW は、PowerMAX 4.2 以上で動作します。

ワークステーションには、32 MB の RAM と 32 MB 以上のスワップスペース保存領域があることが望まれます。24 MB より少ない RAM でもアプリケーションビルダは動作しますが、パフォーマンスは低下します。

# Concurrent PowerMAX のオペレーティングシステムパッチ

Concurrent PowerMAX 4.2 上で LabVIEW の基本的なネットワークが機能するためには、Concurrent より入手いただける下記のパッチをインストールする必要があります。

- inet-005
- システムにより、以下のうちの 1 つ。
  - base-007 Power Hawk 610、Power Hawk 620、およびすべてのシングルプロセッサ PowerStack システム
  - base-008 Power Hawk 640 およびマルチプロセッサ PowerStack II システム
  - base-009 Night Hawk システム

## LabVIEW アプリケーションビルダライブラリのインストール

旧バージョンのアプリケーションビルダからアップグレードする場合は、旧バージョンの上に LabVIEW 5.1.1 のライブラリをインストールします。

### Windows

LabVIEW アプリケーションビルダ CD-ROM の Windows ディレクトリには、32 ビットの Windows (Windows 95/98 および NT) システム用のライブラリが入っています。LabVIEW 5.1.1 は、Windows 3.x には対応していません。LabVIEW 5.0.1 は、Windows 3.x がサポートされている最終版です。



**メモ** プロフェッショナル開発版の LabVIEW を購入されて、付属 CD が含まれていなかった場合は、プロフェッショナル開発版の CD から LabVIEW アプリケーションビルダをインストールしてください。

1. CD-ROM を挿入します。
2. Windows ディレクトリを開きます。
3. setup.exe を実行します。
4. 必要に応じてパスを LabVIEW ディレクトリに変更し、**Install** を選択します。



**メモ** ウイルス検出プログラムによっては、インストーラプログラムに割り込むものがあります。インストールを開始する前に、配布メディアでウイルスのチェックを行なってください。そして自動ウイルスチェッカーをオフにし、インストーラを起動してください。インストール終了後に、もう 1 度ハードディスクのウイルスチェックを行ない、それからウイルスチェッカーをオンにします。

# Macintosh

LabVIEW アプリケーションビルダ CD-ROM の Macintosh ディレクトリには、Power Macintosh 用のライブラリが入っています。LabVIEW 5.1.1 は、680x0 ベースの Macintosh システムには対応していません。Motorola 680x0 対応 Macintosh でサポートされている最終版のアップグレードについては、LabVIEW 5.0.x をご利用ください。



## メモ

ウイルス検出プログラムによっては、インストーラプログラムに割り込むものがあります。インストールを開始する前に、配布メディアでウイルスのチェックを行なってください。そして自動ウイルスチェッカーをオフにし、インストーラを起動してください。インストール終了後、もう 1 度ハードディスクのウイルスチェックを行ない、それからウイルスチェッカーをオンにします。

1. CD-ROM を挿入します。
2. Mac フォルダを開き、LabVIEW AppLibs Installer アイコンをダブルクリックします。
3. **Install** ボタンを選択すると、保存先のフォルダを選択するプロンプトが表示されますので、ここで LabVIEW フォルダを選択します。

# UNIX

次の項では、UNIX 対応 LabVIEW アプリケーションビルダをインストールする手順を説明します。これらのライブラリをインストールするのに Root 特権は必要ありませんが、これらのライブラリをインストールする予定の LabVIEW ディレクトリに書き込みができなくてはなりません。

```
tar xv
```

## Linux および HP-UX (英語版)

1. CD-ROM をマウントします。
2. HP-UX には、以下の UNIX コマンドをタイプ入力します。

```
cd /cdrom/HP-UX
```

ここで、`/cdrom` は CD-ROM をマウントしたディレクトリを示します。  
Linux には、以下の UNIX コマンドをタイプ入力します。

```
cd /cdrom/linux
```

ここで、`/cdrom` は CD-ROM をマウントしたディレクトリを示します。
3. 以下のコマンドをタイプ入力して、インストーラプログラムを起動します。

```
./INSTALL
```
4. 画面上の指示に従ってください。

## Sun (英語版)

1. CD-ROM をマウントします。
2. Solaris 2 では、以下の UNIX コマンドをタイプ入力します。

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

ここで、`/cdrom`は CD-ROM をマウントしたディレクトリを示します。
3. 以下のコマンドをタイプ入力して、インストールプログラムを起動します。

```
./INSTALL
```
4. 画面上の指示に従ってください。

## Concurrent PowerMAX (英語版)

1. テープドライブにテープを挿入します。
2. 以下のコマンドでディレクトリを既存の LabVIEW ディレクトリに変更します。下記のディレクトリへの書き込みアクセスがなければなりません。

```
cd /opt/labview
```
3. 以下のコマンドをタイプ入力します。

```
tar x
```

# アプリケーションビルダライブラリのインストールを確認する

---

アプリケーションビルダライブラリをインストールした後、LabVIEW を起動し、**プロジェクト » アプリケーションの作成**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かを確認します。上記の手順に従ってアプリケーションビルダライブラリをインストールしましたら、デフォルトとして AppLibs ディレクトリは LabVIEW のディレクトリの中に挿入されます。このディレクトリがない場合、上記の手順で再度アプリケーションビルダライブラリを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る場合、`example` ディレクトリには `appbuild.llb` の例が含まれます。`example` ディレクトリにこの例が含まれていなかった場合はインストールプログラムはエラーメッセージを表示したはずです (**Windows および Unix**)。この例は、この文書で後ほど説明するアプリケーションの作成方法のチュートリアルで使用します。詳しくは、本書の「[アプリケーションビルド例](#)」の項を参照してください。

# アプリケーションビルダライブラリの変更点

---

## バージョン 5.1 から 5.1.1 への変更点

この項では、バージョン 5.1 から 5.1.1 で変更された機能について説明します。

- 以前は、作成した時点から移動されたスクリプトファイルを開こうとした場合、アプリケーションビルダがエラーを発生しました。今は、作成してからビルドスクリプトファイルを移動できるようになりました。
- アプリケーションビルダは名前参照された DLL に関して問題がありました。
- アプリケーションにサポートファイルを追加している際、**アプリケーションビルダ**ダイアログボックスのソースファイルタブにある**ディレクトリ内容を追加**のボタンでディレクトリの内容すべてを追加できるようになりました。
- VI ライブラリ (.11b) のすべてをサポートファイルとしてアプリケーションに追加できるようになりました。以前は、VI ライブラリからは個々のファイルしか追加できませんでした。
- **ソースファイルタブ**および**インストールのカスタマイズ (インストール先の設定)**ダイアログボックスにあるリストボックスから一つ以上の項目を選択できるようになりました。容易にファイルの削除とインストール先の設定が行えるようになりました。
- インストーラの設定でインストール先のディレクトリからの**インストール先**および**相対パス**の設定が行えるようになりました。アプリケーションの EXE ファイルはインストール先のディレクトリへのみインストールできることにご注意ください。
- **VI 設定**タブの VI リストはアルファベット順に表示されるようになりました。
- **(Windows)** インストーラを作成するディレクトリを指定できるようになりました。
- **(Windows)** アプリケーションに ActiveX サーバを使用しているときは、自動的にタイプライブラリが追加されるようになりました。タイプライブラリのインストール先およびビルドに関する変更はできません。ActiveX サーバを使用しているときは EXE ファイルとタイプライブラリのバージョンが一致しなければならないので、EXE ファイルの**既存ファイルの入れ換え**のインストール設定を変更することはできません。

## バージョン 5.0 から 5.1 への変更点

この項では、バージョン 5.0 から 5.1 で変更された機能について説明します。

LabVIEW 5.1 では、アプリケーションを構築するプロセスが簡略化されました。旧バージョンでアプリケーションを構築するには、まずライブラリに VI を保存し、それから**アプリケーションの作成**ダイアログボックスを使用してアプリケーションを構築しなければなりません。さらに、Windows でインストーラを構築するには、**配布キットの作成**ダイアログボックスを使用する必要があります。

LabVIEW 5.1 では、**アプリケーションの作成**ダイアログボックスを使用して、これらの操作をすべて行ないます。**アプリケーションの作成**ダイアログボックスのタブ内で、アプリケーションをさまざまな設定に構成します。それらの設定を確定してからスクリプトに保存し、アプリケーションの再構築を簡単に行います。詳細については、本書の「[アプリケーションの作成](#)」の項を参照してください。

## LabVIEW アプリケーションの機能

---

LabVIEW の機能の詳細については、「G プログラミングリファレンスマニュアル」の第 27 章「アプリケーションを管理する」を参照してください。この章では、ソースコードを複数の開発者が管理するためのヒントや、VI 履歴項目の使用方法を説明しています。

### 標準機能

LabVIEW アプリケーションは、VI 上の操作だけでできる簡略化されたユーザインタフェースを特徴としています。メニューには編集オプションが含まれていません。その例として、**保存**コマンドや、**関数**パレット、**制御器**パレットがありません。

メニューには VI 操作に関連する項目のみが表示されます。VI の編集はできませんので、ポップアップメニューは短く、開発システムで VI 実行中に表示する項目と同じものしか表示されません。

**(Windows および UNIX)** ポップアップメニューにアクセスするには、制御器や表示器をマウスで右クリックします。

**(Macintosh)** ポップアップメニューにアクセスするには、コマンドキーを押したまま制御器や表示器をクリックします。

ユーザが使用できる機能は以下の通りです。

- 制御器の操作とその値の変更
- ストリップチャートやグラフ表示器の対話式操作
- スケール範囲の変更

- 制御器、表示器、および配列要素のデフォルト値への設定
- 制御器や表示器のポップアップメニューを使用して、制御器や表示器からデータを切り取り、他の制御器へコピーまたは貼り付け
- 制御器や表示器のポップアップメニューを使用して、項目の説明の表示、およびグラフの制御器パレットの表示など他のランタイム操作の実行
- 開発者が無効にしていない実行パレットボタンの使用
- フロントパネルのログおよび印刷
- **VI 情報を表示**の使用
- ヘルプウィンドウの使用による、制御器および表示器の説明の表示

## アプリケーションの作成

---

LabVIEW でアプリケーションを作成するには、以下の手順に従ってください。

1. **プロジェクト » アプリケーションの作成**を選択します。**アプリケーションの作成**ダイアログボックスが表示されます。**アプリケーションの作成**ダイアログボックスには、**ターゲット**、**ソースファイル**、および**VI 設定**のタブが含まれています。Windows および Macintosh では、**アプリケーション設定**タブがあります。Windows のみに**インストーラ**タブが含まれています。新しいビルドファイルを作成したり、以前に作成したビルドファイルをロードしたりすることができます。
  - 新しいアプリケーションを作成する場合は、ステップ 2 に進みます。
  - ビルドファイルが既にある場合は、**ロード**をクリックしてロードする .bld ファイルを選択し、ステップ 6 に進みます。
2. **ターゲット**タブで、次の情報を指定します。
  - **アプリケーション名** — 作成するアプリケーションの名前
  - **保存ディレクトリ** — 新しいアプリケーションを作成して保存するディレクトリのパス名
  - **サポートファイルディレクトリ** — すべてのサポートファイルを保存するディレクトリのパス名
3. **ターゲット**タブの**ビルド**セクションで、以下のオプションから 1 つを選択します。
  - **すべての VI を含む単一アプリケーション** — ユーザの VI をすべて含むアプリケーションを 1 つ作成するには、このオプションをチェックします。
  - **サブ VI に外部ファイルを使用する小型アプリケーション** — メインのアプリケーションを小さくしておきたい場合は、このオプションをチェックします。



4. **ソースファイル**タブで、アプリケーションの構成要素となる VI を構成できます。最上位 VI や動的にロードされた VI を指定したり、VI 以外のファイル (readme ファイルなど) を追加して指定することができます。追加するソースファイルの種類により、次の手順に従ってください。
  - a. 最上位 VI を追加する場合は、**最上位 VI を追加**をクリックします。**ファイルを開く**ダイアログボックスが表示され、ファイル名を入力するか、追加したい VI を探すことができます。最上位 VI を選択すると、そのサブ VI と関連ファイル (メニューファイルや DLL など) もすべて自動的に含まれます。最上位 VI は、アプリケーションを開始すると自動的に開いて実行されます。
  - b. ダイナミック VI を追加したい場合は、**ダイナミック VI を追加**をクリックします。VI が VI サーバを使用してサブ VI を動的に呼び出している場合、LabVIEW はそれを自動的に検知できません。このオプションを使って追加する必要があります。
  - c. サポートファイルを追加したい場合、**サポートファイルを追加**をクリックします。このオプションを使用すると、データファイルをユーザアプリケーションディレクトリに自動的にコピーします。サポートファイルのディレクトリ全体を追加する場合は、ディレクトリの内容を追加をクリックしてください。
  - d. リストからファイルを削除する場合は、ファイルをクリックしてハイライト表示にし、**ファイルを削除**をクリックします。1 つ以上のファイルを削除するには <Shift> キーを押しながらすべてのファイルをクリックして、ハイライト (選択) します。ファイルを削除をクリックします。
5. **保存**をクリックして入力した情報を保存します。**別名で保存**ダイアログボックスが表示されます。\*.bld という拡張子のあるファイル名を入力します。
6. **ビルド**をクリックします。**ビルドステータス**ダイアログボックスが表示されます。
7. 作成操作が終了したら、**完了**をクリックして**アプリケーションの作成**ダイアログボックスを閉じます。

# アプリケーション機能をカスタマイズする

1. 保存先の設定をカスタマイズしたい場合は、次の手順に従います。  
ソースファイルタブから**インストールのカスタマイズ**を選択します。  
**インストール先の設定**ダイアログボックスが表示されますので、そこで以下の設定を構成します。
  - **ビルドの保存先**フィールドにソースパスを指定したり、その右にあるボタンをクリックしてディレクトリを参照して、保存先ディレクトリを修正できます。このオプションを使用して、特定のサブディレクトリにサポートファイルを入れることができます。
  - **(Windows) スタートメニュー**にインストーラの一部としてプログラム項目を追加したい場合は、**プログラム項目を作成**チェックボックスを選択して名前を入力します。
  - **(Windows) メニューリング**から**インストールディレクトリ**、**Windows ディレクトリ**、または Windows システムディレクトリを選択してインストール先を指定できます。インストール先からの相対パスにもインストールを指定できます。
  - **(Windows) インストーラ**を作成する際に、**既存ファイルの入れ替え**の方法を指定することができます。プロンプトを表示させるか、またその方法について、**常に入れ替える**、**尋ねる**、**新しい場合入れ替える**、または**入れ替えない**のいずれかを選択してください。



## メモ

ご使用のアプリケーションに ActiveX サーバを使用しているときは、自動的にタイプライブラリが追加されるようになりました。EXE ファイルとタイプライブラリのバージョンが一致しなければならないので、タイプライブラリのインストール先の変更、ビルドに関する変更、および EXE ファイルの**既存ファイルの入れ替え**のインストール設定の変更はできません。

2. **(Windows および Macintosh) アプリケーション設定**タブで、アプリケーションの機能をカスタマイズすることができます。  
Macintosh でのメモリサイズを指定したり、Windows でアイコンや ActiveX サーバの機能をカスタマイズできます。
  - a. **(Windows) 独自のアイコン**を指定したい場合は、**カスタムアイコン**チェックボックスをクリックしてアイコン (.ico ファイル) へのパスを指定します。
  - b. **(Windows) ActiveX サーバ**を有効にしたい場合は、**ActiveX サーバを有効にする**チェックボックスをクリックします。  
そうすることにより、アプリケーションは ActiveX クライアントからの要求に応答することができます。ユーザーアプリケーションの ActiveX サーバの機能性は、LabVIEW ActiveX サーバのサブセットです。アプリケーション myapp.exe を作成すると、LabVIEW は実行可能ファイルとともに ActiveX タイプライブラリ myapp.tlb も作成します。タイプライブラリは作成可能なクラスである *Application* と、ディスパッチクラスの

*Virtual Instrument* を定義し、それらのクラスのプロパティとメソッドをエクスポートします。これらクラス、プロパティ、およびメソッドについては、LabVIEW Help ディレクトリの `lvcomm.hlp` を参照してください。アプリケーションを配布する際は、タイプライブラリとヘルプファイルが実行可能ファイルとともに入っていることを確認してください。

サーバ名 `ActiveX` サーバにアプリケーション名を割り当てると、アプリケーションはシステムレジストリ内で固有に判別されません。アプリケーションが完成したら、システムのレジストリに登録するため、少なくとも1回は実行する必要があります。アプリケーションが登録されると、ActiveX クライアントはサーバ名を使ってサーバオブジェクトにアクセスします。例えば、サーバ名を `myapp` と指定した場合、クライアントは `myapp.application` を使用して `application` オブジェクトにアクセスします。

- c. **(Macintosh)** メモリサイズ制御器を使用して、アプリケーションに割り当てられるメモリを指定します。

## ビルドに含まれる VI を修正する

---

**VI 設定** タブを使用して、アプリケーションビルドの一部に入れたいユーザ VI の修正を指定します。ユーザは、様々なウィンドウオプションや実行オプションの VI 設定を有効にしたり無効にすることができます。これらの設定は構築プロセスのみに適用され、ユーザのソース VI には影響しません。

アプリケーションを最小限にとどめるため、LabVIEW はデバッグコードやブロックダイアグラム、および不要なフロントパネルを削除します。LabVIEW はほとんどすべてのケースで、どのパネルが必要かを判断することができます。ただし、VI サーバを使用してフロントパネルを動的に開いた場合は、**VI 設定** タブを使用してパネルが必要なことを指定しなければなりません。編集する VI の行をクリックし、**ビルド設定の編集** ボタンを選択します。さらに、**サブ VI ノード設定** ダイアログボックスを通してサブ VI のフロントパネルを表示している場合は、その VI にフロントパネルが必要であることを指定しなければなりません。

編集は一度に一つの VI のみ可能です。デフォルトでは、不要なフロントパネルはすべて削除されるようになっています。**パネルを削除** オプションを **いいえ** に設定すれば、デフォルトを無視してフロントパネルを含めるようにすることができます。

設定を変更するには、VI を選択してリスト内でハイライト表示にします。**ビルド設定を編集** ボタンをクリックすると、**ビルド設定の編集** ダイアログボックスが表示されます。各設定については、**はい**、**いいえ**、または**変更しない** を選択することができます。すべての設定を終えたら、**適用** をクリックします。ビルドの各 VI がすべて所定の設定になっていることを確認します。

# インストーラを作成する (Windows のみ)

---

1. インストーラを作成するには、**インストーラ**タブで次の手順に従います。**インストーラを作成**チェックボックスをクリックします。このタブのページの以下のセクションを確認します。

- インストール名
- インストーラディレクトリ
- スタートメニューのプログラムグループ
- デフォルトインストールディレクトリ
- インストール言語
- メディアサイズ
- ディスク 1 に残す空き容量 (KB)

**インストーラ**タブで指定したインストーラディレクトリにインストーラファイルが作成されます。このディレクトリにはセットアッププログラムと、data.001、data.002、のように名前のついたファイルが含まれます。ディスクイメージをフロッピーに入れる場合は、setup と data.001 ファイルを第 1 のフロッピーに、data.002 ファイルを第 2 のフロッピー、という順にコピーすることをお勧めします。

**メディアサイズ**項目は、720 KB、1.2 MB、または 1.4 MB のフロッピーに対し、ファイルを断片化する方法を指定します。ファイルを CD で配布する場合でも、ファイルの断片化が必要です。ただし、インストーラを CD またはドライブから実行する場合は、すべてのファイルを同じディレクトリに入れて、そのディレクトリからセットアッププログラムを実行します。

**ディスク 1 に残す空き容量 (KB)** 項目では、第 1 のディスク上のスペースを予約できます。第 1 のフロッピーに readme ファイルを入りたい場合に、ディスク容量の予約が必要です。

2. **上級**ボタンをクリックします。**上級インストール設定**ダイアログが表示されます。
3. アンインストールを作成したい場合は、**アンインストールを作成**チェックボックスを選択します。
4. インストール終了後プログラムを実行したい場合は、**実行可能ファイルをインストール後に実行**チェックボックスを選択します。この項目を使用してインストールを完了させるプログラムを実行することもできます。例えば、.ini ファイルやレジストリファイルを修正する DOS バッチプログラムや C プログラムを書く場合があります。必要な修正を行なうには、まずファイルをインストールの一部としてインストールして、その後で実行します。実行するファイルは、インストールしたファイルのどれか 1 つである必要があります。

- インストール完了後に実行可能ファイルを実行したい場合は、**コマンドライン引数**を使用してプログラムに渡された引数を指定することができます。標準の引数を指定する以外にも、以下の項目のいずれかをコマンドライン引数文字列に埋め込むことができます。

%dest ユーザが選択したアプリケーションインストールディレクトリ

%src setup.exe を含むディレクトリ

%group インストールプログラムグループ名

%name インストール名

これらのオプションのいずれかがインストール時に存在すると、インストーラは引数が実行可能ファイルに渡される前に適切な値に置き換えます。

## アプリケーションビルド例

---

LabVIEW 開発システムでアプリケーションを作成する例については、以下の手順に従ってください。

- examples¥appbuild.llbにある Sample VI を開きます。この VI は Histogram VI を含む解析 VI をいくつか呼び出し、さらにその解析 VI が外部サブルーチンを呼び出します。**VI 設定**ダイアログボックス内の項目は現在、ウィンドウのいくつかの属性が非表示になるように構成されています。
- Sample VI を実行して動作を確認します。終了したら**停止**ボタンをクリックします。LabVIEW を終了する場合以外は、**終了**ボタンはクリックしないでください。
- Sample VI を閉じます。
- アプリケーションの作成**ダイアログを開くために**プロジェクト»アプリケーションの作成**を選択します。
- アプリケーションに sample.exe (**Windows**) または sample (**Macintosh**) という名前を付けます。
- EXE ファイルの保存する場所を保存先のディレクトリで設定します。
- ソースファイル**ページから最上位 VI を追加を選択します。ファイル名フィールドに examples¥appbuild.llb¥sample を入力します。
- ダイナミック VI を追加を選択して、examples¥appbuild.llb¥About Sample をファイル名フィールドに入力します。アプリケーションは、名前が About で始まる VI を自動的に**アプリケーションについて**ダイアログボックスとして使用します。
- ビルド**をクリックしてアプリケーションをビルドします。

10. アプリケーションを実行する際、最上位 VI が自動的に開いて実行します。アプリケーションビルダは VI が実行されるまえに、ツールバーの**停止**ボタンを自動的に使用不可に設定します。サンプル VI のサブ VI はすべて自動的にアプリケーションに追加されます。
11. この時点で、**アプリケーションの作成**ダイアログボックス内のその他のオプションを使用して、アプリケーションのビルドを簡略化したいと考えているかもしれません。例えば、アプリケーションではツールバーを非表示にし、ソース VI ではデバッグしやすいよう表示させたい場合などです。そのような変更を行なうには、**VI 設定**タブを使用します。また Windows では、**アプリケーション設定**タブを使用してカスタムアイコンを指定したり、**インストーラ**タブを使用してインストーラを作成することができます。
12. アプリケーションは、デフォルトで簡易メニューを使用します。サンプルアプリケーションのメニューを見て、内容を確認してください。このメニューをさらにカスタマイズして、**ファイル**»**開く**のようなオプションや**操作**メニューのデータロギングオプションなどを削除したい場合があります。また、**終了**オプションを変更したい場合もあります。アプリケーションにその操作をさせるのではなく、カスタムオプションにして Quit Application 関数を呼び出す前にユーザ独自の処理（ファイルを閉じる、I/O の中断など）を行なえるようにします。

## アプリケーションを配布する

---

以下の項では、LabVIEW アプリケーションの配布について説明します。

### アプリケーションに必要なその他のファイル

作成したアプリケーションに他の必要なファイルを添付して配布する必要があることがあります。この場合、実行やその他の操作に必要なファイルは、アプリケーションと同じディレクトリに入っていないとなりません。

作成したアプリケーションが GPIB やデータ集録ボードを使用する場合、ボードに付属のハードウェアドライバをインストールする必要があります。

**(Windows および UNIX)** アプリケーションでシリアルポートを使用する場合は、アプリケーションに LabVIEW ディレクトリの serpdrv ファイルを含める必要があります。また Windows で National Instruments のデータ集録ボードを使用する場合は、LabVIEW ディレクトリにある daqdrv ファイルを含める必要があります。アプリケーションの作成ダイアログボックスディレクトリでは、**サポートファイル**を追加オプションを使用してこれらのファイルを含めることができます。

**(Windows)** Windows 上で LabVIEW の実行可能プログラムを開発して、他のコンピュータにインストールする際は、LabVIEW ランタイムエンジンをインストールする必要があります。プログラムを使用するコンピュータには、プログラムを実行する前に LabVIEW ランタイムエンジンインストーラを使用してランタイムエンジンをインストールしなければなりません。アプリケーションビルダを使用してプログラムを配布する場合、ランタイムエンジンは自動的にインストールされます。これにより、実行可能プログラムのサイズが大幅に縮小されます。アプリケーションビルダで作成したインストーラでアプリケーションを配布する場合、インストーラがランタイムエンジンを自動的にインストールします。ランタイムエンジンの詳細については [www.ni.com](http://www.ni.com) を参照してください。Application Builder と Run-time Engine をキーワードとして Knowledge Base を検索してください。



## メモ

アプリケーションビルダが正常にインストールされている場合、ランタイムエンジンは自動的にデフォルトの位置にインストールされています。アプリケーションビルダが正常にインストールされている場合、旧バージョンのランタイムエンジンを上書きできません。ランタイムエンジンは同バージョンの LabVIEW で作成された EXE (実行可能) ファイルのみを実行できます。ランタイムエンジンはコンピュータにインストールする最初のプログラムのみインストールする必要があります。

**(Windows)** 以下の項目は、作成するアプリケーションの機能により、必要とされる追加ファイルです。

- アプリケーションで ActiveX タイプライブラリや ActiveX コントロールを使用している場合は、タイプライブラリをインストールする必要があります。必要に応じて ActiveX コントロールやサーバのヘルプファイルをインストールできます。また、タイプライブラリを登録しなければなりません。それを行なうには、ActiveX コントロールおよびサーバに別途のインストーラを使用するのが最も良い方法です。ActiveX ライブラリやコントロールの多くに、その ActiveX コンポーネントを再配布する前に検討しなければならないライセンスの確認事項があります。アプリケーションのインストールに ActiveX コンポーネントのインストールを統合したい場合は、InstallShield や WISE など他社のインストーラを使用してください。
- LabVIEW ActiveX コンテナは、resource ディレクトリにある `ole_lv5container.dll` という DLL を使用します。ActiveX コントロールを含むアプリケーションを作成してそれを他のコンピュータに移動する場合は、このファイルをビルドしたアプリケーションと同じファイル、または System ディレクトリに入れる必要があります。
- LabVIEW には `msvcrt.dll` と `mfc42.dll` という 2 つのシステム DLL が必要ですが、それらはすべてのシステムにインストールされない場合もあります。`msvcrt.dll` は常に必要ですが、`mfc42.dll` はアプリケーションが ActiveX コンテナを使用する

場合にのみ必要です。それらのファイルが入っていない場合は、LabVIEWにより Windows の System ディレクトリにインストールされます。ユーザは、アプリケーションを実行するコンピュータ上にこれらのファイルがあることを確認する必要があります。それらのファイルは Windows の System ディレクトリ、またはアプリケーションと同じディレクトリのいずれかに入れることができます。Windows ディレクトリに入れた場合は、他のアプリケーションによってインストールされたこれらの DLL の新しいバージョンを上書きしないよう注意が必要です。

**(Macintosh)** 解析ルーチンまたはプログラム間通信 (PPC) VI を使用する Power Macintosh 用のアプリケーションを作成している場合は、Shared Libraries フォルダのコピーをアプリケーションの入っているフォルダに入れます。**サポートファイルを追加**オプションを使用して、これらを配布アプリケーションに含めることができます。

## 配布権

ご使用のプラットフォームでの配布権に関する情報につきましては、ソフトウェアパッケージに付属の「LabVIEW Software License Agreement」を参照してください。

## 環境の設定

---

作成したアプリケーションには、**環境設定**ダイアログボックスがありません。変更を行なうと、LabVIEW はアプリケーションの環境設定情報を環境設定ファイル (**Windows および Macintosh**)、または .labviewrc ファイル (**UNIX**) に書き込みます。この情報の形式は LabVIEW の形式と同じです。詳しくは、「G プログラミングリファレンスマニュアル」の第 7 章「環境をカスタマイズする」を参照してください。

**(Windows)** 異なるのはファイル名のみで、labview の代わりにアプリケーション名になっています。例えば、作成したアプリケーションに simple.exe という名前を付けた場合、環境設定は Simple.ini に保存されます。アプリケーションには必ず環境設定ファイル (.ini) を入れるようにしてください。そのためには、そのファイルを**アプリケーションの作成**ダイアログでサポートファイルとして扱います。

**(Macintosh)** 異なるのはファイル名のみで、labview の代わりにアプリケーション名になっています。例えば、作成したアプリケーションに simple.exe という名前を付けた場合、環境設定は Simple Preferences に保存されます。アプリケーションには必ず環境設定ファイルを入れるようにしてください。そのためには、そのファイルを**アプリケーションの作成**ダイアログでサポートファイルとして扱います。



(UNIX) スタンドアロンアプリケーションの環境設定ファイルを作成するには、ターゲットコンピュータに .labviewrc ファイルを作成し、ユーザのホームディレクトリ内に入れる必要があります。このファイル内で、<application file name>.<preference> の形式で環境設定を作成します。そのためには、**アプリケーションの作成**ダイアログボックスでそのファイルをサポートファイルとして扱います。例えば、LabVIEW の .labviewrc ファイルは、環境が labview.<preference> と設定されています。実行可能ファイル特定の環境を新しく作成するには、LabVIEW で環境を設定して .labviewrc ファイルのコピーをターゲットコンピュータのホームディレクトリに入れるのが簡単な方法です。ここでユーザがしなければならないのは、すべての実行可能ファイル名が labview で始まるように変更することです。ただし環境設定ファイル名は .labviewrc のままです。

## VI 設定オプションを使用して VI オプションを限定する

アプリケーションを設計する際、どのユーザオプションが必要かを考えてください。例えば、LabVIEW 開発システムには**実行停止**ボタンがありますので、VI のテストや停止が簡単にできます。このボタンは、重要なデータの入出力が進行中でもプログラムを直ちに停止してしまうので、ユーザがこの方法で VI を停止するのは避けたい方がよいでしょう。代わりに、フロントパネル制御器を使用してプログラムを同期させて停止します。実行停止ボタンはデフォルトでは構築プロセスからは除かれています。

VI の **VI 設定**項目を使用して、**実行停止**ボタンのような実行操作をユーザが使用できるようにすることができます。VI がサブ VI である場合は、**サブ VI ノード設定**コマンドを使用して、例えばサブ VI が呼び出されたときにフロントパネルが開くように構成するなど、サブ VI の操作を限定することができます。

これらのオプションの多くは、**アプリケーションの作成**ダイアログボックスの **VI 設定**タブにもあります。このページを使用して、ビルトの一部として適用する変更点を特定することができます。こうすることにより、デバッグがしやすい方法で VI を構成しながら、完成したアプリケーションで変更することもできるようになります。ユーザがコントロールのデフォルト値を設定またはグラフのオートスケールを設定しないように、ランタイムのポップアップメニューを使用不可に設定することができます。

さらに、パネルをカスタマイズすることもできます。例えば、スクロールバーを非表示にしたり、特定の VI をダイアログボックスのように機能させることができます。**VI 設定** » **ダイアログボックス**を選択すると、パネルはモーダル、つまりアクティブの時ユーザは他のパネルと対話できない状態になります。

# ヘルプ情報を提供する

VI 開発者は他のユーザ用に、VI のロードや操作に必要な情報を含め、VI を文書化することをお勧めします。通常の LabVIEW ドキュメントセットは著作権により保護されています。したがって、作成したユーザアプリケーションにこのマニュアルを添付することはできません。

オンラインヘルプを作成するには、各パネルの **VI 情報を表示** ダイアログボックスの説明フィールドに情報を入力することができます。また、制御器や表示器の **説明** ダイアログボックスに情報を書き込むこともできます。ユーザがヘルプウィンドウを開いて表示器上にカーソルを移動すると、ヘルプウィンドウは説明の情報を表示します。

## 一般的なエラー

UNIX 対応 LabVIEW の起動時に発生する可能性のある一般的なエラーを以下の表に示します。

表 1 UNIX での一般的な LabVIEW 起動エラー

エラーメッセージ	説明
"Xlib: connection to :0.0 refused by server" "Client is not authorized to connect to server"	<p><b>ケース 1: 推定される原因</b> — LabVIEW を 1 つのコンピュータで実行して、別のコンピュータで表示しようとしている。</p> <p><b>ケース 1: 解決法</b> — 表示するコンピュータにウィンドウを開く許可を与えるため、LabVIEW を実行している xhost と &lt;コンピュータ名&gt; を使用する。</p> <p><b>ケース 2: 推定される原因</b> — ディスプレイサーバでウィンドウを開く許可を得ていないユーザが LabVIEW を実行しようとした。通常このエラーは、一時的に Root (スーパーユーザ) など他のユーザになるため su コマンドを実行した後起こります。</p> <p><b>ケース 2: 解決法</b> — su コマンドを終了し、ログインユーザとして LabVIEW を起動します。</p>
"Executable version doesn't match resource file"	<p><b>推定される原因</b> — LabVIEW の実行可能ファイルのバージョンが labview.rsc のバージョンと一致していません。</p> <p><b>解決法</b> — 構成ファイルの appResFilePath パラメータで、labview.rsc ファイルへのパスが正しく設定されていることを確認してください。</p>



321315D-01

0100