

LABVIEW™

Version 6.0

このアップグレードノートでは、Windows、Macintosh、UNIX 対応 LabVIEW をバージョン 6.0 にアップグレードするプロセスについて説明します。

インストール手順、およびその他の重要な情報については、「LabVIEW リリースノート」をお読みください。

アップグレードノートについて

本書では、LabVIEW 6.0 へのアップグレード時に発生する可能性のある問題に関する情報や、新機能の情報について説明します。

詳細については、

LabVIEW 6.0 の新機能の詳細について、「LabVIEW ユーザマニュアル」および「LabVIEW ヘルプ」を参照してください。「LabVIEW ヘルプ」にアクセスするには、**ヘルプ** » **オンラインリファレンス** を選択します。

目次

アップグレード時の問題点	3
VI を変換する	3
Applib とツールセットをアップグレードする	3
旧バージョンの LabVIEW をアップグレードする	4
LabVIEW 5.x からアップグレードする	4
LabVIEW 4.x からアップグレードする	5
LabVIEW 3.x またはそれ以前のバージョンからアップ グレードする	6
LabVIEW 6.0 の機能	7
波形データタイプ	7
波形データタイプの成分	7
フロントパネル上の波形データ	8
波形の VI と関数	8

I/O 名制御器	8
DataSocket の改良	9
ユーザインタフェースの機能	9
メニューの再編成	9
クラスタと配列のショートカットメニューオプション	13
制御器パレットと関数パレットの操作性の改良	13
自動配線機能	14
新規作成ダイアログボックス	14
印刷ダイアログボックス	14
LabVIEW でのグラフ印刷の改良	14
VI プロパティダイアログボックス	15
カラーパレット	15
フロントパネルの制御器と表示器	16
グラフおよびチャート上の複数の x スケールと y スケール	17
グラフおよびチャートのエイリアス除去ラインのプロット	17
波形デジタルグラフ	18
ダイアル、ノブ、メータ、およびゲージのカラーバー	18
フロントパネルとブロックダイアグラムの一般的な改良点	18
フロントパネルオブジェクトのグループ化とロック	18
フロントパネルオブジェクトおよびブロックダイアグラム オブジェクトの文書化	18
While ループの改良点	18
フォーミュラノードの変更点	19
Expression Node	19
テキストラベルの切り替え	20
Windows レジストリアクセス VI	20
制御器リファレンス	20
自動的にリンクされるプロパティノード	21
制御器 Refnum	21
エラー処理の改良点	22
While ループでのエラー処理	22
Case ストラクチャでのエラー処理	22
追加エラー処理関数	22
実行ハイライトを使用してエラーを処理する	22
共有ライブラリ (DLL) を作成する	23
配列関数の強化点	23
文字列関数の強化点	23
ファイル I/O 関数	23
多形性 VI	24
バリエーションデータ	24
ActiveX の強化点	25
ActiveX のプロパティとメソッドにアクセスする	25
レポートにおける複数のプラットフォームおよび HTML の機能	25
ローカリゼーションの強化	25
LabVIEW マニュアルセットの変更点	25

アップグレード時の問題点

LabVIEW 5.x からアップグレードする場合は、「VI を変換する」、「Applib とツールセットをアップグレードする」、および「[LabVIEW 5.x からアップグレードする](#)」をお読みください。

LabVIEW 4.x またはそれ以前のバージョンからアップグレードする場合は、「VI を変換する」、「Applib とツールセットをアップグレードする」、および「[LabVIEW 4.x からアップグレードする](#)」をお読みください。

VI を変換する

LabVIEW アプリケーションのアップグレードプロセスは自動化されています。LabVIEW 4.0 またはそれ以降のバージョンで保存された VI を開くと、LabVIEW 6.0 によってその VI の変換とコンパイルが自動的に行われます。

VI の変換に必要なメモリ量は、VI やそのすべてのサブ VI のディスク上の領域を合計することによって予測できます。これらの VI が VI ライブラリにまとめてある場合は、VI が圧縮されているため、VI ライブラリのサイズのおよそ 30 パーセント分を加算します。変換プロセスでは、少なくとも上記のメモリに 3 MB を加算した量のメモリが LabVIEW の動作に必要です。

VI を一度に変換するだけのメモリがコンピュータにない場合は、VI を段階的にコンポーネントごとに変換します。変換する VI の階層を調べ、下位階層にあるサブ VI をロードして保存することから開始します。その後、徐々にその上位階層に進むことができます。**ツール » 上級 » 一括コンパイル**を選択して VI のディレクトリを変換できますが、このオプションではディレクトリまたは VI ライブラリ内の VI はアルファベット順に変換されます。変換プロセスで上位の VI が先に処理される場合、**一括コンパイル**では、上位の VI を先に開いた場合とおよそ同じ量のメモリが必要になります。

ヘルプ » LabVIEW について を選択してメモリの使用状況を監視し、それまでに使用したメモリ量の合計を表示できます。

Applib とツールセットをアップグレードする

既存のツールセットの多くは問題なく LabVIEW 6.0 で動作します。ただし、LabVIEW 6.0 で使用するには、VI を一括コンパイルする必要があります。VI の一括コンパイルの詳細については、このマニュアルの前出の「VI を変換する」セクションを参照してください。LabVIEW 6.0 は、以下の例外を除いて LabVIEW 4.0 およびそれ以降のバージョン用に設計されたツールと互換性があります。

- **(開発システム) LabVIEW アプリケーションビルダ** – LabVIEW Application Builder 6.0 にアップグレードする必要があります。プロフェッショナル開発版システムバージョン 6.0 には、アップデート済みのアプリケーションビルダライブラリが含まれています。

- **(開発システム) LabVIEW プロフェッショナル G 開発ツールキット**
プロフェッショナル G 開発ツールキット 5.0 またはそれ以降のバージョンをお持ちの場合は、LabVIEW プロフェッショナル開発システムバージョン 6.0 にアップグレードする必要があります。このアップグレードはプロフェッショナル G 開発ツールキット 5.1 の既存のユーザについては無料です。プロフェッショナル開発システムバージョン 6.0 には新しいバージョンのプロフェッショナル G 開発ツールキットが含まれています。
- **LabVIEW テストエグゼクティブ**—LabVIEW テストエグゼクティブ 5.1 またはそれ以前のバージョンをご使用の場合は、LabVIEW 6.0 で使用できるようにこれらの VI を一括コンパイルする必要があります。VI の一括コンパイルの詳細については、このマニュアルの前出の「[VI を変換する](#)」セクションを参照してください。

旧バージョンの LabVIEW をアップグレードする

以下のセクションでは、LabVIEW のバージョンによって異なるアップグレード時の問題点について説明します。

LabVIEW 5.x からアップグレードする

このセクションでは、LabVIEW 5.x から LabVIEW 6.0 にアップグレードする場合に発生する可能性のある問題点について説明します。

データログファイルを変換する

旧バージョンの LabVIEW で作成したデータログファイルを開くと、LabVIEW 6.0 ではそのファイルを LabVIEW 6.0 フォーマットに変換するように促すプロンプトが表示されます。変換するように選択すると、LabVIEW によってデータログファイルが新しいフォーマットに変換されたデータで置き換えられます。ファイルを変換しないように選択するとエラーが返され、ファイルは開きません。

データログファイルを開いたときに自動的に変換するには、LabVIEW.ini ファイルなどの LabVIEW の環境設定ファイルに以下の行を追加します。

```
silentDatalogConvert=True
```

データログファイルを開いたときに自動的に変換されないようにするには、この行を False に設定します。

LabVIEW 5.x サーバと LabVIEW 6.0 クライアント間の互換性の問題

LabVIEW 6.0 クライアントを LabVIEW 5.x アプリケーションの VI サーバに接続するとエラーが生じます。これは、LabVIEW 5.x では LabVIEW 6.0 VI サーバプロトコルの新しい機能を認識できないためです。

LabVIEW 5.x クライアントから LabVIEW 6.0 アプリケーションの VI サーバへの接続は可能です。

LabVIEW における UDP の変更点

UDP VI (現在は UDP 関数)は関数 » 通信 » UDP パレットから使用できます。

LabVIEW 4.x からアップグレードする

このセクションでは、LabVIEW 4.x 以降のブールデータ形式および VI コントロール VI における変更について説明します。

ブールデータを LabVIEW 4.x に、および LabVIEW 4.x からブールデータに変換する

データの格納形式が、LabVIEW 4.x と LabVIEW 5.x の間で変更されました。LabVIEW 4.x では、データが配列でない場合はブールデータが 2 バイトに格納されます。データが配列の場合は、LabVIEW 4.x では各ブール要素が 1 ビットに格納されます。LabVIEW 6.0 ではそれが配列であるかどうかにかかわらず、1 つのブール値が 1 バイトに格納されます。この変更により、ブール値の配列をサポートできるブロックダイアグラムの関数が増え、これらの配列の動作がさらに数値の配列と一貫性のあるものになります。新しいブール値のデータ形式によって、コードインタフェース ノード (CIN) でのデータ操作に影響がありますが、LabVIEW 6.0 には既存の CIN との互換性があります。

1 つまたは複数のブール値を含んでいるバイナリデータを LabVIEW 4.x のファイルに書き込む場合、その形式は LabVIEW 5.x および 6.0 で同じデータを書き込む場合とは異なります。LabVIEW 6.0 には、LabVIEW 4.x で書き込まれたバイナリデータを読み込み、LabVIEW 4.x が読み込めるバイナリデータを書き込むメカニズムが備わっています。Write File、Read File、Type Cast、Flatten To String、および Unflatten From String の 5 つの関数には、**4.x データの変換** ショートカットメニュー項目があります。このメニュー項目を選択すると、これらの関数では、バイナリデータを LabVIEW 4.x で書き込まれたデータのように扱うことができます。LabVIEW 4.x にフォーマットされたデータを作成するには、Write File、Flatten to String、または Type Cast 関数を使用します。LabVIEW 4.x 用にフォーマットされたデータを読み込むには、Read File、Unflatten From String、または Type Cast 関数を使用します。**4.x データの変換** ショートカットメニュー項目を選択すると、LabVIEW 6.0 ではその関数上に赤い 4.x を描画して LabVIEW 4.x 形式に、あるいは LabVIEW 4.x 形式からデータを変換中であることを示します。データの変換を中止するには、**4.x データの変換** ショートカットメニュー項目をもう一度選択して無効にします。

ブール値が含まれているデータファイルが複数ある場合、これらのファイルを開き、LabVIEW 6.0 で認識できる新しいデータファイルにデータを書き込む VI を作成できます。

LabVIEW 4.x またはその前のバージョンで保存した VI を LabVIEW 6.0 でロードすると、LabVIEW 6.0 では、Write File、Read File、Type Cast、Flatten To String、および Unflatten From String 関数上に **4.x データの変換** 属性が自動的に設定されます。これらの関数は 4.x のときと同じように機能します。VI で新しい LabVIEW 6.0 ブールデータ形式を使用する必要があると判断した場合は、上記にリストした各関数上でもう一度 **4.x データの変換** メニュー項目を選択してこれを無効にします。通常、VI に、旧バージョンの LabVIEW で書き込まれたブールデータを含んでいるファイルを操作する必要がない場合、または旧バージョンの LabVIEW で実行されている VI とブールデータをやり取りする必要がない場合は、新しい LabVIEW 6.0 ブールデータ形式を使用してください。LabVIEW の今後のバージョンでは、前回のブールデータ形式のサポートは継続されない可能性があります。

データログファイルを変換する

アップグレード時のデータログファイルの変換方法の詳細については、前出のセクション「[LabVIEW 5.x からアップグレードする](#)」の「[データログファイルを変換する](#)」を参照してください。

VI 制御 VI

LabVIEW 6.0 では VI 制御 VI (`vi.lib\utility\vict1.llb`) はデフォルトのパレットセットから削除され、互換性 VI として存在しています。その機能は **関数** **アプリケーション制御** パレット上の VI サーバ関数である Open VI Reference、Call By Reference、プロパティノード、およびインボークノードに含まれています。

VI 制御 VI から渡されるエラーコードには、LabVIEW 6.0 で変更になったものがあります。旧バージョンの LabVIEW では、VI 制御 VI はエラーコード 7 および 1000 を渡していました。LabVIEW 6.0 の VI 制御 VI はコード 1004 および 1003 を渡します。LabVIEW 4.x で作成された VI によってエラーコード 7 および 1000 をチェックする場合、LabVIEW 6.0 でその VI を機能させるには変更が必要です。

DDE VI

DDE VI (`vilib\platform\dde.llb`) はデフォルトのパレットセットから削除され、互換性 VI として存在しています。

LabVIEW 3.x またはそれ以前のバージョンからアップグレードする

LabVIEW 3.x またはそれ以前のバージョンからのアップグレードについては、ナショナルインスツルメンツのウェブサイト ni.com を参照してください。

LabVIEW 6.0 の機能

このセクションでは LabVIEW 6.0 の機能と改良点について説明します。



メモ 旧バージョンの LabVIEW の機能に関する情報については、「LabVIEW 5.0 アップグレードノート」(321780A-01、ナショナルインスツルメンツのウェブサイト)で入手可能)を参照してください。

波形データタイプ

測定値を集録、解析、表示する VI を作成するために使用する VI、関数、およびフロントパネルオブジェクトは、デフォルトでその波形データタイプを使用します。波形データタイプの配列は複数の波形を表現します。



メモ 旧バージョンの LabVIEW では、波形を数値配列およびクラスタとして表現しました。LabVIEW 5.x またはそれ以前のバージョンで波形を作成しアクセスする VI を作成すると、配列やクラスタとして波形の操作を継続したり、その VI を変換して波形データタイプを使用することができます。

波形データタイプには、1 つの波形に関連するデータ値とタイミング情報のデータが含まれています。

波形データタイプによって、波形成分が測定アプリケーション作成に使用する VI および関数に渡ります。波形の成分を抽出して編集するには、波形の VI および関数を使用します。

波形データタイプの成分

波形データタイプは、波形情報のみを含んでいる成分の特別なクラスタです。波形の各成分はクラスタ内の要素で、データタイプは各成分を表します。個々の成分にアクセスして操作するには、波形関数を使用します。

開始時刻 (T_0)

タイムスタンプは波形の始点の開始時刻です。波形グラフではデータプロット時にタイムスタンプを使用します。デフォルトは 0 で、これは LabVIEW システム時刻の 1904 年 1 月 1 日午前 12 時 (グリニッジ標準時) を表します。

Δt (dt)

Δt は波形の連続したデータ点間の時間です。

波形データ (Y)

波形データは波形を表す数値の 1 次元配列です。配列の各要素は波形の振幅を表すデータ値です。一般的に、配列内のデータ値の数は、データ集録デバイスから取るスキャンの回数に一致します。

波形属性

波形属性は、データ発生源のチャンネルやデバイスなどの波形データに関する情報を提供します。

フロントパネル上の波形データ

フロントパネル上で、波形データは**制御器** » I/O パレット上の**波形制御器**によって表示できます。

波形データタイプの t_0 、 dt 、 Y 、および**属性**のコンポーネントを操作したり、これらのコンポーネントを表示器として表示するために、**波形制御器**を使用します。属性にアクセスするには、波形制御器を右クリックしてショートカットメニューから**項目を表示** » **Attributes** を選択します。

制御器 » **波形**パレットの制御器を使用しても波形を表示できます。

波形の VI と関数

関数 » **波形**パレットには、波形の作成や操作、波形の個々の成分の操作を行う VI や関数が含まれています。**関数** » **データ集録**パレットおよび**関数** » **解析**パレット上にある VI の多くは、波形データタイプを使用します。

I/O 名制御器

DAQ チャンネル名、VISA リソース名、およびIVI 論理名を選択するには、**制御器** » I/O パレットまたは**制御器** » **旧バージョンの制御器** » I/O パレット上の I/O 名制御器を使用します。これらの名前は Measurement & Automation エクスプローラ (MAX) で構成します。計測器またはデータ集録デバイスと通信するには、これらの I/O 名制御器とともに I/O VI を使用します。DAQ チャンネル名、VISA リソース名、およびIVI 論理名の、MAX を使用した構成方法の詳細については、**ツール** » **Measurement & Automation エクスプローラ**を選択してください。

I/O 名定数は、**関数** » **計測器**パレットおよび**関数** » **データ集録**パレットから使用できます。

MAX は Windows でのみ使用できます。**(Macintosh)** DAQ チャンネル名を構成するには、DAQ チャンネルウィザードを使用します。

(Macintosh および UNIX) VISA リソース名を構成するには VISA 構成ユーティリティを使用します。

DataSocket の改良

他の VI や、ナショナルインスツルメンツの ComponentWorks など他のアプリケーションと、ウェブ上またはローカルコンピュータ上でデータを共有するには、ナショナルインスツルメンツの DataSocket テクノロジーを使用します。確立されている通信プロトコルが DataSocket によってまとめられ、測定や自動化が可能になります。これは、ウェブブラウザによって、異なるインターネットテクノロジーがまとめられることによく似ています。

DataSocket テクノロジーによって、複数の入出力メカニズムにアクセスできるようになります。**DataSocket 接続**ダイアログボックスを通じてフロントパネルから、または DataSocket Read および Write 関数によってブロックダイアグラムからアクセスが可能です。URL の指定によって、パブリッシュ (書き込み)、またはデータのサブスクライブ (読み取り) を行います。これはウェブブラウザ内で URL を指定する方法とよく似ています。

たとえば、ウェブ上の別のコンピュータとフロントパネル上の温度表示器のデータを共有する場合、**DataSocket 接続**ダイアログボックスで URL を指定することによって温度計のデータを公表します。別のコンピュータ上のユーザは自分のフロントパネル上に温度計を配置し、**DataSocket 接続**ダイアログボックスで URL を選択することによって、そのデータをサブスクライブします。

このパレットにはバリエーション関数も含まれています。LabVIEW におけるバリエーションデータの詳細については、後続のセクション「**バリエーションデータ**」を参照してください。

ユーザインタフェースの機能

新しいユーザインタフェースとさらに便利になった機能について、以下に説明します。

メニューの再編成

関連する機能をまとめて配置し、メニューをさらに直感的に操作しやすくするため、LabVIEW 6.0 ではメニューが再編成されました。

ファイルメニューの変更

- **印刷、プロジェクト » Documentation Tool**、および**文書の印刷**は新しい**印刷**ダイアログボックスにまとめられました。このダイアログボックスには、**ファイル » 印刷**を選択してアクセスします。
- **ファイル » 新規**を選択すると、**新規作成**ダイアログボックスが開きます。このダイアログボックスでは、**新規 VI**、**グローバル変数**、**制御器**、**ランタイムメニュー**、**多形性 VI**、**作成スクリプト**、**VI テンプ**

レート、グローバル変数テンプレート、および制御器テンプレートを作成できます。また、**新規作成**ダイアログボックスを使用して既存のテンプレートに基づく新規ファイルも作成できます。

- **テンプレートの編集**は**ファイル**メニューから削除されました。
- **ファイル**»**開く**を選択すると、**開く**ダイアログボックスが開き、すべての LabVIEW ファイルを開くことができます。テンプレート (.vit または .ctt ファイル) を開くと、LabVIEW ではファイルの保存時にテンプレートファイルの拡張子が保存されます。**ファイル**»**新規作成**ダイアログボックスの**テンプレートから開始**オプションを使用して、既存のテンプレートに基づく新規 VI、グローバル変数、または制御器を作成します。
- **コピーの保存**は削除されました。**別名で保存**ダイアログボックスには、発呼者を更新せずにコピーを保存できるチェックボックスがあります。
- **VI 設定**および**VI 情報を表示**は**ウィンドウ**メニューから移動して、新しい**VI プロパティ**ダイアログボックスにまとめられました。このボックスには**ファイル**»**VI プロパティ**を選択してアクセスします。
- **VI ライブラリを編集**、**一括コンパイル**、**CVI 計測器ドライバをインポート**、および**VXI プラグアンドプレイドライバを更新**は**ツール**メニューに移動しました。

編集メニューの変更

- **制御器の編集**は**制御器を編集する**に名前が変更されました。
- **編集メニュー**は**ランタイムメニュー**に名前が変更されました。
- **検索**および**検索結果を表示**が**プロジェクト**メニューから追加されました。
- **環境設定**は**オプション**に名前が変更され、**ツール**メニューに移動しました。
- **ユーザ名**は**ツール**メニューに移動しました。
- **パスワードキャッシュ消去**は削除されました。**オプション**ダイアログボックスの**その他**ページに、**パスワードキャッシュ消去**ボタンがあります。

プロジェクトメニューからツールメニューに名前変更

プロジェクトメニューは**ツール**メニューに名前を変更されました。**ツール**メニューは以下のように変更されています。

- **(Windows) Measurement & Automation エクスプローラ**によって Measurement & Automation エクスプローラ (MAX) にアクセスします。コンピュータに接続されている計測器およびデータ集録ハードウェアを構成するには、MAX を使用します。
- **計測器拡張メニュー**が追加されました。ここには**計測器ドライバネットワーク**および**CVI 計測器ドライバをインポート**の項目が含まれています。

- **DAQ ウィザード** 拡張メニューの名前がデータ集録に変更されました。ここでは **DAQ Channel Viewer** および **DAQ ソリューション ウィザード** 項目が含まれています。
(Macintosh) データ集録メニューにも **DAQ チャネルウィザード** が含まれています。
- **文書化ツール** は印刷ダイアログボックスに移動しました。このダイアログボックスには **ファイル** » **印刷** を選択してアクセスします。
- **ファイルマネージャ** は **VI ライブラリマネージャ** に名前が変更されました。
- **VI レビジョン履歴** は **ウィンドウメニュー** から移動しました。
- **VI 階層を表示**、**この VI の発呼者**、**この VI のサブ VI**、**開かれていないサブ VI**、および **開かれていない Type Defs** は、**参照メニュー** に移動しました。
- **検索** および **検索結果を表示** が **編集メニュー** に移動しました。
- **プロフィールウィンドウを表示** は **VI のプロファイル** に名前が変更され、**上級拡張メニュー** に移動しました。
- **VI ライブラリを編集** は **ファイルメニュー** から移動しました。
- **上級拡張メニュー** が追加されました。ここでは、**一括コンパイル**、**文字列をエクスポート**、**文字列をインポート**、**ActiveX コントロールをインポート**、**プロファイル VI** が含まれています。
- **比較拡張メニュー** が追加されました。これはすべての比較関数をグループ化したものです。この項目はプロフェッショナル開発システム構成の LabVIEW ユーザにのみ有効です。
- **アプリケーションビルダ** をお持ちの場合は、**アプリケーションを作成** 項目の名前が **アプリケーションまたは共有ライブラリ (DLL) を作成** に変わりました。

参照メニューの追加

これは新しいメニューです。参照メニューには、現在の VI やその階層の様子を表示可能にする項目が含まれています。

- **VI 階層を表示** は **プロジェクトメニュー** から移動しました。
- **この VI の発呼者** は **プロジェクトメニュー** から移動しました。
- **この VI のサブ VI** は **プロジェクトメニュー** から移動しました。
- **開かれていないサブ VI** は **プロジェクトメニュー** から移動しました。
- **開かれていない Type Defs** は **プロジェクトメニュー** から移動しました。

ウィンドウメニューの変更

- **VI 情報を表示**は**ファイル**メニューに移動され、**VI プロパティ**に名前が変更されました。
- **履歴を表示**は**VI レビジョン履歴**に名前が変更され、**ツール**メニューに移動しました。

ヘルプメニューの変更

- **シンプルヘルプ**は**ヘルプ**メニューから削除されました。**ヘルプ**ウィンドウを**シンプルヘルプ**と**詳細なヘルプ**で切り替えるには、**ヘルプ**ウィンドウの左下隅にある**シンプルヘルプ**ボタンをクリックします。
- **(現在の) VI のオンラインヘルプ**は**ヘルプ**に名前が変更されました。
- **サンプルの検索**は**サンプル**に名前が変更されました。
- **LabVIEW マニュアル**は**印刷版マニュアルを表示**に名前が変更されました。
- **オンラインチュートリアル**は**ヘルプ**メニューから削除されました。「**LabVIEW チュートリアル**」には、**LabVIEW ダイアログボックス**の**LabVIEW チュートリアル**ボタンをクリックするとアクセスできます。

ショートカットメニューの再編成

LabVIEW 6.0 ではショートカットメニューが再編成されています。通常使用される機能はメニューの上位に設置されており、上級機能は**上級**拡張メニューに置かれています。

- **表示**は**項目を表示**に名前が変更されました。
- **制御器に変更**および**表示器に変更**が定数ショートカットメニューに追加されました。
- **定数に変更**がフロントパネル端子ショートカットメニューに追加されました。
- **データ操作**拡張メニューから複数の項目が削除され、新しい**上級**拡張メニューに追加されました。
 - **キー操作**は**上級**拡張メニューに移動しました。
 - **同期表示**は**上級**拡張メニューに移動しました。
 - **オンラインヘルプ**は**ヘルプ**に名前が変更され、**上級**拡張メニューのあるすべての端子の**上級**拡張メニューに移動しました。**上級**拡張メニューのない端子については、**ヘルプ**は**上級**ショートカットメニューの最上位にあります。
 - **カスタマイズする**は**上級**拡張メニューに追加されました。
- **データ操作**の下位にあった**説明**項目は**説明とヒント**に名前が変更され、そのショートカットメニューの最上位に移動しました。
- **中断点設置**および**中断点解除**が、このショートカットメニューの最上位に追加されました。

クラスタと配列のショートカットメニューオプション

クラスタおよび配列のショートカットメニューにいくつかの項目が追加されました。

- **上級** » **制御器を隠す** を選択すると、クラスタまたは配列の制御器は表示されません。
- **上級** » **有効なステータス** を選択すると、制御器を淡色表示させたり、無効にしたり、さらに有効にしたりできます。
- **上級** » **非表示要素を表示** を選択すると、クラスタ内の非表示になっていた項目が表示されます。
- **自動サイズ調整** によって、クラスタは**合うようにサイズ変更**、**水平に揃える**または**垂直に揃える**ようにサイズ変更されます。クラスタを水平方向または垂直方向に自動的に揃える場合には、クラスタ内の項目がオーバーラップすることはありません。
- **上級** » **前に要素を挿入**および**要素を削除** を選択すると、1次配列内の配列要素を挿入したり削除できます。
- **上級** » **前に行を挿入**、**前に列を挿入**、**行を削除**、**列を削除** を選択すると、2次元配列内の配列要素の行や列を挿入したり削除できます。

制御器パレットと関数パレットの操作性の改良

制御器パレットおよび関数パレット上の操作バーによって、ウェブブラウザで操作ボタンを使用する要領でこれらのパレットを操作できます。サブパレットボタンをクリックすると、パレット全体が選択したサブパレットに変わります。操作バーにも検索機能が含まれており、パレット項目をより早く見つけるのに役立ちます。

制御器および関数パレットには以下の操作ボタンがあります。

- **上** – パレット階層で1つ上位に移動します。
- **検索** – パレットを検索モードに変更します。検索モードでは、テキストベースの検索を実行してパレット内で制御器、VI、または関数を探すことができます。
- **オプション** – 関数ブラウザオプションダイアログボックスを開きます。このダイアログボックスからパレットの外観を構成できます。

パレット上で制御器、VI、および関数を検索する

制御器および関数パレット上の関数、VI、または制御器についてテキストベースの検索を実行するには、**検索**ボタンを使用します。名前の頭文字(**先頭**)またはキーワード(**含む**)によって検索できます。制御器、VI、または関数をダブルクリックすると、パレット上のその位置に移動できます。

自動配線機能

自動配線機能を使用すると、オブジェクトをブロックダイアグラム上に配置するだけで簡単にそのオブジェクトを配線できます。また、ブロックダイアグラムに既にあるオブジェクトを自動的に配線することもできます。自動配線機能を使用すると、LabVIEW では最適な組み合わせとなる端子が接続され、適合しない端子は接続されません。

自動配線機能では、選択したオブジェクトをブロックダイアグラム上の他のオブジェクトの近くに移動すると、有効な配線を示すため一時的な配線が表示されます。提示された配線が適正な場合、そのオブジェクトをドロップすると自動的に配線されます。

オブジェクトの位置を決めている間にスペースバーを押すと、自動配線のオンとオフを切り替えることができます。関数パレットからオブジェクトを選択する場合や <ctrl> キーを押したままドラッグして既にブロックダイアグラム上にあるオブジェクトのコピーを作成する場合は、デフォルトで自動配線は有効になります。位置決めツールを使用してブロックダイアグラム上に既にあるオブジェクトを移動する場合は、デフォルトで自動配線は無効になります。

自動配線が有効になっていると、選択されたオブジェクトはドラッグ時にその外観が保持されます。自動配線が無効になっていると、選択されたオブジェクトはドラッグ時に点線の枠になります。

新規作成ダイアログボックス

新規作成ダイアログボックスを使用して LabVIEW のファイルを作成します。新規 VI、多形性 VI、制御器、グローバル変数、ランタイムメニュー、VI テンプレート、制御器テンプレート、およびグローバル変数テンプレートを作成できます。また、**新規作成**ダイアログボックスを使用して既存のテンプレートに基づく新しいファイルも作成できます。

印刷ダイアログボックス

VI、制御器、グローバル変数、またはテンプレートのマニュアルを印刷したり、マニュアルを HTML、RTF、または標準テキストファイルに保存するには、**印刷**ダイアログボックスを使用します。印刷するマニュアルをカスタマイズしたり、ページ設定のプロパティを構成することもできます。

LabVIEW でのグラフ印刷の改良

LabVIEW 6.0 ではプリンタ分解能を使用してグラフやチャートを印刷します。明確な線や画像でグラフを鮮明に印刷することによって、主に標準印刷や PostScript 印刷に効果が現れます。

VI プロパティダイアログボックス

以下のような VI の多くの特徴を構成するには、**VI プロパティダイアログ** ボックスを使用します。

- VI アイコンの外観
- VI の呼び出し時にフロントパネルをどのように表示するか、およびどこに表示するか
- VI の呼び出し時の VI の実行方法
- ヘルプウィンドウに表示されるテキストの内容、および VI 履歴オプションを使用するかどうか
- VI のパスワード保護を有効にするかどうか

VI プロパティダイアログ ボックスにアクセスするには、ブロックダイアグラムまたはフロントパネル上の VI アイコンを右クリックしてショートカットメニューから **VI プロパティ** を選択するか、**ファイル** » **VI プロパティ** を選択します。

VI プロパティダイアログ ボックスは、以前は **VI 設定** ダイアログボックス および **VI 情報** ダイアログボックスにあった特徴の多くを組み合わせています。

カラーパレット

カラーパレットは、カラーツールを選択し、オブジェクトや背景を右クリックすると使用可能になります。このカラーパレットは広範なスペクトルのカラーパレットに置き換えられました。カラーパレットの一段目にはグレースケールがあります。二段目には背景色やフロントパネルオブジェクトに適した柔らかい色のスペクトルが入っています。三段目には、ハイライトに適した色のスペクトルが入っています。一番下の段には、透過色と最近使用した 10 色を保存するウィンドウがあります。

またこの一番下の段には現在選択されている色を表示するウィンドウもあります。前景色と背景色のあるオブジェクトでは、カラー表示器の左半分に前景色、右半分に背景色が表示されます。オブジェクトの前景色のみ、または背景色のみにも色を付ける場合は、<f> または キーをそれぞれ押したまま色を選択します。

(Windows および Macintosh) オペレーティングシステムの**カラーパレット**にアクセスするには、**その他** ボタンをクリックします。カラーパレットから、色相、彩度、明度、および、赤、緑、青の値を使用して特定の色を選択できます。

(UNIX) more ボタンをクリックすると、LabVIEW ダイアログボックスにアクセスします。このダイアログボックスで赤、緑、および青の値を定義することによって色を選択できます。

フロントパネルの制御器と表示器

制御器パレットは再編成され、新しい制御器および表示器に更新されました。

3次元制御器と表示器

多くのフロントパネルオブジェクトは、さらに洗練された3次元の外観になるように再設計されました。色を付けたりオブジェクトのサイズを変更するなどの機能を含め、オブジェクトの機能と動作は変更されていません。制御器を最適な外観にするには、モニタは少なくとも16ビット色に設定する必要があります。

制御器パレットの再編成

LabVIEW 6.0では新しい機能を盛り込むため、制御器パレットが再編成されました。

- 新しいI/Oパレットには新しいDAQ、VISA、およびIVI論理名制御器や、IMAQセッション制御器が含まれています。
- **文字列 & パス**パレットには、文字列とファイルパスの制御器が含まれています。
- **リング & Enum**パレットには、メニューリングなどのリングおよびenum制御器が含まれています。
- 新しい**リスト & 表**パレットには表とリストボックスの制御器が含まれています。
- 元のフロントパネルオブジェクトは**制御器 » 旧バージョンの制御器**パレットから使用できます。これらのオブジェクトは256色および16色のモニタに最適です。

タブ制御器

フロントパネル制御器および表示器を狭い領域で重ねる場合は、タブ制御器を使用します。タブ制御器はページとタブで構成されています。フロントパネルオブジェクトをタブ制御器の各ページに配置し、異なるページを表示するセレクトアとしてタブを使用します。タブ制御器上に配置できるフロントパネルオブジェクトの数に制限はありません。タブ制御器は、**制御器 » 配列 & クラス**、**制御器 » ダイアログ**パレット、および対応する**制御器 » 旧バージョンの制御器**パレットから使用できます。

まとめて使用するフロントパネルオブジェクトが複数ある場合や操作の特定の段階では、タブ制御器は便利です。たとえば、計測を開始する前にまず複数の設定値を構成し、次に進行に合わせてパラメータの側面の変更を可能にし、さらに特定のデータを表示し格納するVIがあるとします。タブ制御器を使用すると、そのVIの各段階の制御器と表示器のみを表示できます。

ブロックダイアグラム上では、デフォルトでタブ制御器はenum制御器になります。タブ制御器に配置された制御器および表示器の端子は別のブロックダイアグラムの端子として現れます。

多項目リストボックス

オプションのリストおよび各オプションの情報を提示させるには、**制御器** » **リスト & 表**パレット上にある多項目リストボックスおよび**制御器** » **ダイアログ制御器**パレット上にあるダイアログ多項目リストボックスを使用します。多項目リストボックスは1つまたは複数を選択するリストボックスに似ていますが、多項目リストボックスにはオプションの大きさや作成された日付などの情報を各オプションに追加できる列が含まれています。列数の設定や各列のヘッダの作成が可能です。複数選択や単一選択の多項目リストボックスも作成できます。

グラフおよびチャート上の複数の x スケールと y スケール

グラフおよびチャートで複数の x スケールと y スケールがサポートされるようになりました。複数のスケールを使用すると、共通の x または y スケールを共有しない複数のプロットを表示できます。

グラフとチャートのスケール凡例

複数のスケールを扱えるように、波形グラフおよびチャートにスケール凡例が追加されました。スケールプロパティを構成するだけでなく、スケール凡例を使用してスケールにラベルを付けることができます。

操作ツールを使用時に**スケール形式**ボタンを押すと、形式、精度、マッピングモード、プロットとスケールラベル表示の切り替え、グリッド、グリッドライン、およびグリッドカラーのフォーマットを構成できます。

グラフおよびチャートのエイリアス除去ラインのプロット

エイリアス除去ラインによってチャートおよびグラフのラインプロットの外観を向上できます。エイリアス除去ライン描画を有効にすると、色の影によってラインプロットはなめらかに表示されます。エイリアス除去ライン描画では、ライン幅、ラインスタイル、ポイントスタイルなどは変更できません。



メモ エイリアス除去は波形デジタルグラフでは使用できません。

プロット凡例が表示されていない場合、チャートおよびグラフを右クリックし、**項目を表示** » **プロット凡例**を選択して凡例上で右クリックし、**エイリアス除去**を選択します。



メモ エイリアス除去ライン描画は計算主導です。エイリアス除去ラインプロットによってVIの速度に影響があることがあります。

波形デジタルグラフ

デジタルデータを表示するには、**制御器** » **グラフパレット**から使用できるデジタル波形グラフを使用します。これはタイミングダイアグラムやロジックアナライザで作業する場合に効果的です。

ダイヤル、ノブ、メータ、およびゲージのカラーバー

フロントパネル上のダイヤル、ノブ、メータ、またはゲージのいずれにもカラーバーを追加できます。カラーバーは、ゲージが危険な値に達していることを示す警告範囲などのデータ範囲を視覚的に示すのに便利です。

フロントパネルとブロックダイアグラムの一般的な改良点

フロントパネルおよびブロックダイアグラムを改善するため、いくつかの変更が行われました。以下のセクションで新機能について説明します。

フロントパネルオブジェクトのグループ化とロック

フロントパネルオブジェクトをグループ化し、そのフロントパネル上の位置をロックできます。グループ化されたオブジェクトはその相対位置が維持されますが、ロックされたオブジェクトはフロントパネル上の位置が維持され、削除できません。

フロントパネルオブジェクトおよびブロックダイアグラムオブジェクトの文書化

個々のフロントパネルオブジェクトおよびブロックダイアグラムオブジェクトに個別の説明を追加できます。**説明とヒント**ダイアログボックスに入力する情報は、そのオブジェクトの**ヘルプ**ウィンドウおよび作成するVIの文書に、ヒントラベルとして表示されます。

ヒントラベルは簡単な説明で、オブジェクトの上にカーソルを配置するとそのオブジェクトの上または近くに表示されます。ヒントが入力されていないと、ヒントラベルは表示されません。



メモ

ヒントは制御器、表示器、または定数にのみ使用できます。ブロックダイアグラムの関数やVIにはヒント情報を入力できません。

While ループの改良点

LabVIEW 6.0 では、While ループの条件設定が簡単になりました。デフォルトで、While ループはその条件端子に FALSE の値が渡されると停止します。条件端子の動作を変更するには、端子や While ループの枠を右クリックして **True の場合停止**または **True の場合、継続**を選択します。デフォルトの動作は **True の場合、継続**です。

フォーミュラノードの変更点

フォーミュラノードで以下の新しい操作が実行可能になりました。

- 整数ビットごとの演算 (&, |, ~, <<, >>, および ^)



メモ

旧バージョンの LabVIEW の指数演算 ^ は ** によって置き換えられ、VI を LabVIEW 6.0 に再コンパイルするとき自動的に更新されます。現在、^ 演算はビットごとの排他的論理和になっています。

- ショートカット割り当て (+=, -=, *=, /=, >>=, <<=, &=, \=, ^=, |=, %=, および **=)
- 増分 (++) 前後および減分 (--) 前後の演算

フォーミュラノードにも、配列の指標が数値データであることやビット演算の演算対象が整数データであることを確かめるタイプチェックが実装されています。

配列の複数の指標が範囲内にあることを確認します。配列の場合、範囲外の値はデフォルトでゼロに、範囲外の割り当てはデフォルトで nop になり、演算が行われないことを示します。

フォーミュラノードによって自動タイプ変換も実行されます。

フォーミュラノードでの C 言語の機能

テキストベースの方程式の実行だけでなく、フォーミュラノードを使用して多くの C 言語関数を実行できます。C 言語の機能を利用すると、C 言語プログラミングのスコープルールで、if ステートメント、While ループ、For ループ、Do ループ、switch ステートメント、break および continue ステートメント、配列操作、および複合ステートメントを実行できます。

Expression Node

1 つの変数を持つ表現や方程式を計算するには、**関数 » 数値パレット**で使用できる Expression Node を使用します。Expression Node は、方程式が変数は 1 つだけでも他の点は複雑な場合に効果的です。

Expression Node では入力端子に渡した値が変数の値として使用されます。出力端子では計算値が返されます。

Expression Node の多形性

Expression Node の入力端子は、配線した制御器または定数と同じデータタイプになります。出力端子は入力端子と同じデータタイプになります。入力のデータタイプには、非複素スカラ、非複素スカラの配列、または非複素スカラのクラスタがあります。配列やクラスタでは、Expression Node によって入力配列やクラスタの各要素に式が適用されます。

テキストラベルの切り替え

回転または上下に整列した垂直テキストを表示するには、テキストラベルの切り替えを使用します。

テキストラベルを切り替えるには、ラベル上で右クリックして**垂直に配置**を選択し、拡張メニューから以下のいずれかのオプションを選択します。

- **なし**—ラベルは通常のテキストとして表示されます。
- **上下に整列**—ラベルは上下に整列したテキストとして表示されます。
- **時計回り**—ラベルテキストは時計回りに 90 度回転されます。この結果、下に向かって読む垂直方向のテキストになります。
- **反時計回りに**—ラベルテキストは反時計まわりに 90 度回転されます。この結果、上に向かって読む垂直方向のテキストになります。

Windows レジストリアクセス VI

Windows レジストリキーの作成、開く、クエリ、列挙、閉じる、および削除を実行するには、**関数 » 上級 » Windows レジストリアクセス VI** パレットから使用できる Windows レジストリアクセス VI を使用します。Windows レジストリキーの値の列挙、読み取り、書き込み、および削除を実行することもできます。



注意 Windows レジストリを誤って変更すると、Windows オペレーティングシステムが動作しなくなる場合があります。

制御器リファレンス

制御器リファレンスによってフロントパネル情報をサブ VI に渡すことができます。制御器リファレンスがサブ VI に渡されると、プロパティノードおよびインボークノードを使用してプロパティを読み込んで構成したり、参照されるフロントパネルオブジェクトのメソッドを呼び出すことができます。



メモ 属性ノードは LabVIEW から削除されました。属性ノードは制御器 refnum が自動的に入力されるプロパティノードによって置き換えられました。

旧バージョンの LabVIEW の属性ノードは、6.0 ではプロパティノードによって置き換えられています。そのため、データ名が変更された属性もあります。以下に新しい属性名をリストします。

- Caption および Caption Visible プロパティ
- 文字列およびパス制御器の Scroll Position、Selection、および Text Colors プロパティ
- リング制御器の Ring Text Size、Ring Text Colors、および Numeric Text Colors プロパティ
- カラーバー制御器の Numeric Text Colors、Ramp Visible、および Dig Display Visible プロパティ

- 多くのグラフまたはチャートのスケールのプロパティ
- 多くのグラフまたはチャートのプロットのプロパティ
- 多くのグラフカーソルのプロパティ
- 多くのスライドスケールのプロパティ

これらの属性をクラスタに入れてそのクラスタを Unbundle by Name に配線すると、データ名が変更されているために枠が壊れた状態が表示されます。これは手作業で修正する必要があります。この問題を修正するには、Unbundle by Name ノードを右クリックして適切なプロパティをもう一度選択します。

自動的にリンクされるプロパティノード

フロントパネル制御器からプロパティノードを作成すると、そのプロパティノードは制御器に自動的にリンクされます。参照される制御器を検出するには、プロパティノード上で右クリックして**検索** » **制御器** を選択するか、プロパティノードをダブルクリックして制御器を探します。

自動的にリンクされるプロパティノードを選択し、**編集メニューから選択範囲をサブ VI に変換** を選択してサブ VI を作成すると、制御器リファレンスが作成され、サブ VI に配線されます。

LabVIEW 5.x またはそれ以前のバージョンで作成された属性ノードは、デフォルトで自動的にリンクされるプロパティノードに変わります。

制御器 Refnum

制御器 » **Refnum** パレットには、制御器リファレンスをサブ VI に渡したり、サブ VI から受け取るために使用する制御器 refnum があります。制御器 refnum をフロントパネルにドロップすると、一般的な制御器リファレンスタイプになります。制御器 refnum を構成するには、特定のフロントパネル制御器をドラッグします。たとえば、デジタル制御器用の制御器 refnum を作成するには、制御器 refnum をドロップし、デジタル制御器をその上にドラッグします。

一般的なタイプの制御器 refnum を作成するには、制御器 refnum を右クリックし、**VI サークラスを選択拡張メニュー** から選択します。一般的なタイプの refnum を厳密に類別化されたものにするには、この refnum を右クリックし、**データタイプを含む** を選択します。

フロントパネル制御器 refnum をその参照される制御器として表示させるには、refnum 上で右クリックし、**制御器を表示** を選択します。アイコンに復元するには、refnum 上で右クリックし、**アイコンを表示** を選択します。

エラー処理の改良点

ブロックダイアグラムでのエラー処理を改善するため、いくつかの変更が行われました。プール値を受け入れる多くの関数およびストラクチャではエラークラスタのステータスプール値パラメータも識別され、ブロックダイアグラム上でエラークラスタを簡単に配線できます。

While ループでのエラー処理

While ループの繰り返し端子にエラークラスタを配線すると、端子の機能にエラー処理が加わります。ショートカットメニュー項目の **True** の場合停止および **True** の場合、**継続**は、**エラーで停止**および**エラーの場合、継続**に変わります。

Case ストラクチャでのエラー処理

Case ストラクチャの条件端子にエラークラスタを配線すると、サブダイアグラム表示ウィンドウには**エラー**および**エラーなし**の2つのケースが表示されます。Case ストラクチャの枠の色は、**エラー**のケースでは赤、**エラーなし**のケースでは緑に変わります。

追加エラー処理関数

関数 » アプリケーション制御パレットの Stop と Quit LabVIEW 関数および**関数 » 比較**パレット上の Select 関数も、入力値としてエラークラスタを受け入れます。これらの関数のいずれかにエラークラスタを配線している状態でエラーが発生すると、その関数に値 TRUE が渡されます。

関数 » 時間 & ダイアログパレットには、Merge Errors VI が含まれています。この関数を使用すると、別の関数からのエラー I/O クラスタを合わせることができます。この VI では、最初にステータスが TRUE のエラークラスタを探します。最初に検出されたエラーがレポートされます。この VI によってエラーが検出されない場合、警告を探し、最初に検出された警告が返されます。警告も検出されない場合、エラーステータス FALSE が返されます。

実行ハイライトを使用してエラーを処理する

実行ハイライトを使用して VI のデバッグを実行すると、各ノードでデバッグ実行後、そのデータタイプと同じ色の枠を付けて値が表示されます。エラークラスタからエラーがレポートされると、表示されるエラー値には赤い枠が付きます。エラーがレポートされない場合、値 **OK** の枠は緑になります。

共有ライブラリ (DLL) を作成する

(プロフェッショナル開発システム) LabVIEW 6.0 では、スタンドアロンアプリケーションの作成と同様に共有ライブラリ (DLL) を作成できます。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。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が表示されます。このダイアログボックスには次のタブがあります。

- **ターゲット** – このタブでは、共有ライブラリの保存場所に名前を付けたり、その場所を選択できます。
- **ソースファイル** – このタブでは、共有ライブラリにファイルを追加したり、共有ライブラリからファイルを削除できます。
- **VI 設定** – このタブで共有ライブラリのさまざまなウィンドウや VI の実行オプションを編集します。
- **(Windows) App 設定** – このタブで DLL のカスタムアイコンを指定します。
- **(Windows) インストール** – このタブで共有ライブラリのインストーラを作成します。デフォルトでは LabVIEW ではインストーラは作成されません。

配列関数の強化点

LabVIEW には配列を操作する Replace Array Subset、Insert Into Array、および Delete From Array の 3 つの新しい関数が含まれています。これらは **関数 » 配列** パレット内から使用できます。多くの既存の配列関数は LabVIEW 6.0 では再設計され、配列の操作が簡単になりました。新しい配列機能には以下の機能が含まれます。

- 入力配列の寸法に基づいて配列関数の大きさを自動調整する
- 操作中の配列のサブセットに基づいて入力をグループ化する

文字列関数の強化点

LabVIEW には、文字列内で文字やサブ文字列を検索したり置き換える関数があります。 **関数 » 文字列** パレット上にある新しい Replace Substring および Search/Replace String 関数によって、それぞれ文字列の任意の部分をサブ文字列と置き換えたり、指定したサブ文字列を別のサブ文字列で置き換えたりできます。 **関数 » 文字列 » その他の文字列関数** パレット上にある Split String 関数では、1 文字ではなく、任意の文字列を検索できます。

ファイル I/O 関数

文字列、数値、パス、およびブールデータをテキストとしてフォーマットし、そのテキストをファイルに書き込むには、Format Into File 関数を使用します。多くの場合、Format Into String 関数で文字列をフォーマットし、フォーマットされた文字列を Write Characters to File VI または Write File 関数で書き込む操作の代わりに、この関数を使用できます。

Format Into File 関数では、データがテキストファイルに現れる順番を決めることもできます。ただし、この関数を使用してファイルにデータを追加したり、ファイル内の既存データを上書きすることはできません。これらの操作を行うには、Write File 関数とともに Format Into String 関数を使用します。

文字列、数値、パス、およびブール値についてファイル内のテキストをスキャンし、そのテキストをデータタイプに変換するには、Scan from File 関数を使用します。多くの場合に、Read File 関数または Read Characters from File VI でファイルからデータを読み取り、読み取った文字列を Scan from String 関数でスキャンする代わりに、この関数を使用できます。

Scan from File 関数はファイル内のすべてのテキストを読み取ります。ただし、この関数を使用してファイル内でスキャンを開始する場所を決めることはできません。この操作を行うには、Scan from String 関数とともに Read Characters from File VI を使用します。

多形性 VI

多形性 VI では、1つの入力端子または出力端子で異なるデータタイプを受け入れることができます。多形性 VI は同じコネクタペーンパターンを持つサブ VI の集合体です。各サブ VI は多形性 VI のインスタンスです。ある VI のコネクタペーンの端子が入力端子である場合、別の VI のコネクタペーン上の対応する端子も入力端子であるか、使用されていない必要があります。

たとえば、**関数 » ファイル I/O » 構成ファイル VI** パレットから使用できる Read Key VI は多形性 VI です。その**デフォルト値**端子は、ブール値、数値倍精度、32 ビット整数、パス、文字列、または符号なし 32 ビット整数のデータタイプを受け入れます。

バリエーションデータ

バリエーションデータタイプには、LabVIEW または他のアプリケーションでの実行時に異なるデータタイプであった可能性のあるデータを格納します。バリエーションデータタイプは ActiveX 関数およびプロパティで使用されます。

バリエーションデータを作成して操作するには、**関数 » 上級 » データ操作 » バリエーション**パレット上にあるバリエーション関数を使用します。いずれの LabVIEW のデータタイプもバリエーションデータタイプに変換すると、変換したデータを他の LabVIEW の VI や関数で使用できます。たとえば、文字列をバリエーションデータに変換すると、バリエーションデータタイプはテキストを格納し、テキストが文字列であることを示します。

ActiveX の強化点

LabVIEW 6.0 には、ActiveX のプロパティとメソッドにより簡単にアクセスできる機能があります。

ActiveX のプロパティとメソッドにアクセスする

ActiveX オブジェクトを右クリックし、ショートカットメニューから **プロパティの作成** または **メソッドの作成** を選択して、フロントパネルやブロックダイアグラムから ActiveX オブジェクトの新しいプロパティノードやメソッドノードを作成します。関数パレットを介して操作する必要はなく、ActiveX オブジェクトを制御するときにプロパティノードやメソッドノードを選択してブロックダイアグラム上に配置します。

レポートにおける複数のプラットフォームおよび HTML の機能

LabVIEW 6.0 では、すべてのプラットフォームで HTML ベースのレポートを作成できます。Windows プラットフォームでは、印刷物ベースのレポートを作成して印刷できます。これらの VI は **関数** » **レポート生成** パレットにあります。



メモ HTML ベースのレポート生成機能は LabVIEW 6.0 のベースパッケージでは使用できません。

ローカリゼーションの強化

ラベルやキャプションなどのブロックダイアグラム上の文字列はエクスポート可能です。VI をローカライズする場合にこの機能は便利です。また、文字列定数、パス定数、リング定数、列挙定数 (フォント情報のみ)、配列およびクラスタ定数、およびフリーラベルのブロックダイアグラムオブジェクトに関連する文字列もエクスポート可能です。

LabVIEW マニュアルセットの変更点

ナショナルインスツルメンツでは LabVIEW 6.0 のマニュアルを前回のバージョンから変更しました。特定の LabVIEW 機能および VI 作成時にこれらの機能を使用する方法については、**ヘルプ** » **オンラインリファレンス** から使用できる「LabVIEW ヘルプ」を参照してください。LabVIEW 環境での開発における概念的な情報については、「LabVIEW ユーザマニュアル」を参照してください。データ集録および計測器制御 VI については、「LabVIEW Measurements Manual」を参照してください。上級 VI の開発については PDF 形式で使用可能な「LabVIEW アプリケーションノート」を参照してください。

LabVIEW のマニュアルの概要については、「LabVIEW ユーザマニュアル」の「第 1 章 LabVIEW の概要」のセクション「LabVIEW マニュアルのリソース」を参照してください。