LabVIEW™アップグレードノート

バージョン 8.0

アップグレードノートでは、Windows、Mac OS、および Linux 対応 LabVIEW をバージョン 8.0 にアップグレードするための手順、その際に起こり得る問題、新しい機能について説明します。

LabVIEW 8.0 の機能の詳細と同様に、LabVIEW プログラミング概念、LabVIEW の使用方法の手順、および LabVIEW の VI、関数、パレット、メニュー、ツールの詳細については、『LabVIEW ヘルプ』を参照してください。『LabVIEW ヘルプ』には、ナショナルインスツルメンツからの使用可能な LabVIEW ドキュメントのリソースもリストされています。『LabVIEW ヘルプ』にアクセスするには、**ヘルプ→ LabVIEW ヘルプを検索**を選択します。

目次

LabVIEW 8.0 へアップグレードする 2
- VI の変換 3
- ツールキット、計測器ドライバ、およびアドオンのアップグレード 3
- ナショナルインスツルメンツの追加のソフトウェアをアップグレードする 4
- 旧バージョンの LabVIEW からのアップデート 4

アップグレードおよび互換性の問題 6
- LabVIEW 7.*x* からのアップグレード 6
- LabVIEW 6.*x* からのアップグレード 24
- LabVIEW 5.*x* からのアップグレード 29
- LabVIEW 4.*x* からのアップグレード 30
- LabVIEW 3.*x* 以前のバージョンからのアップグレード 32

LabVIEW 8.0 の機能および変更点 32
- LabVIEW ライセンスをアクティブ化する 33
- LabVIEW を起動する 33
- LabVIEW をログインおよびログアウトする 33
- ドメインアカウントマネージャ 34
- 新規ダイアログボックスの機能強化 34
- RT、FPGA、ならびに PDA ターゲット 34
- LabVIEW プロジェクト 35

NATIONAL INSTRUMENTS™

LabVIEW プロジェクトライブラリ 42
シェア変数を使用してライブデータを共有する 43
ソース配布を作成する 46
旧バージョンのアプリケーションビルダからの変更点 46
新しいパレットの構成 48
メニューの再構成 52
ソース管理を使用する 53
VI および関数の機能強化 55
VI 階層ウィンドウの機能強化 70
フロントパネルの機能強化 70
LabVIEW MathScript（Windows） 82
行列データタイプ 82
オプションダイアログボックスの機能強化 85
別名で保存ダイアログボックスおよび旧バージョン用に保存ダイアログボックス 85
ファイルサイズの向上 86
複数のバージョンの LabVIEW で共有ライブラリを使用する 86
ウィンドウランタイム位置を設定する 86
ランタイム言語の環境設定を行う 86
VI サーバの機能強化 87
クラスブラウザウィンドウ 88
.NET アセンブリを使用する 89
計測器ドライバファインダ 90
計測器ドライバ VI ウィザード 90
計測器ドライバプロジェクトウィザード 90
再入可能 VI を使用およびデバッグする 91
VI 実行後にプローブを作成する 92
複数ターゲットで VI をプロファイルする 92
64 ビット整数データタイプ 93
NI Spy 93
検索 / 置換機能 94
カスタマイズ可能なキーボードショートカット 94
印刷ダイアログボックスの機能強化 94
VI アイコンを編集する 95
ドキュメントの機能強化と変更点 95
NI サンプルファインダの機能強化 98
LabVIEW 8.0 でのその他の機能および変更点 99

LabVIEW 8.0 へアップグレードする

旧バージョンの LabVIEW からアップグレードする場合は、このドキュメントの「VI の変換」、「ツールキット、計測器ドライバ、およびアドオンのアップグレード」、および「アップグレードおよび互換性の問題」セクションをはじめに参照してください。

VI の変換

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 8.0 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 を LabVIEW 8.0 で保存しないと、その VI にアクセスするたびに変換プロセスが開始され、その変換プロセスに余分なメモリリソースが使用されます。

また、再コンパイルを含む、変更が保存されていない VI のパフォーマンスで、実行時間の効率が極端に低下することがあります。パフォーマンスやメモリに関する問題の詳細については、『LabVIEW ヘルプ』を参照してください。

メモ LabVIEW 8.0 で保存した VI は、旧バージョンの LabVIEW ではロードできません。VI を保存して LabVIEW 7.1 で実行するには、**ファイル→旧バージョン用に保存**を選択します。LabVIEW 7.1 以前のバージョンで使用する予定のある VI は、LabVIEW8.0 で保存する前にバックアップコピーをとっておくようにしてください。

すべての 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んでください。トップレベル VI は最後に開いて変換してください。**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することもできます。ただし、一括コンパイルではディレクトリまたは LLB 内の VI はアルファベット順に変換されます。上位の VI から先に変換処理すると一括コンパイルとなり、トップレベルの VI を最初に開くときと同程度のメモリ容量が必要になります。

メモリの使用状況を監視し、現在使用しているメモリ量の概要を表示するには、**ヘルプ→バージョン情報**を選択します。

ツールキット、計測器ドライバ、およびアドオンのアップグレード

LabVIEW 8.0 をインストールしたら、ツールキットおよびアドオンがすべて互換バージョンであることを確認し、そのツールキットおよびアドオンを LabVIEW 8.0 のディレクトリに再インストールします。最初に、旧バージョンの LabVIEW からツールキットをアンインストールする必要がある場合があります。インストールの詳細については、LabVIEW ツールキットまたはアドオンツールのドキュメントを参照してください。

また、LabVIEW 8.0 で使用するため、既存のツールキット、計測器ドライバ、アドオン VI を一括コンパイルしなければなりません。VI の一括コンパイルの詳細については、このドキュメントの「VI の変換」セクションを参照してください。

以下のツールキット、計測器ドライバ、およびアドオンは、LabVIEW 8.0 で使用するためにアップグレードまたはダウンロードする必要があります。

- LabVIEW アプリケーションビルダを使用する場合は、LabVIEW アプリケーションビルダ 8.0 にアップグレードする必要があります。LabVIEW8.0 プロフェッショナル開発システムにはアプリケーションビルダ 8.0 が含まれています。LabVIEW アプリケーションビルダのインストール方法の詳細については、`labview¥readme` ディレクトリにある『LabVIEW アプリケーションビルダ Readme』を参照してください。
- LabVIEW 8.0 では、VI Analyzer 1.1 を使用しなければなりません。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）にアクセスし、info code として `exd8yy` と入力すると、VI Analyzer 1.1 の Upgrade Advisor にアクセスすることができます。
- LabVIEW 8.0 で Internet Toolkit 6.0 を使用するには、追加の VI をダウンロードする必要があります。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）にアクセスし、info code として `exkrkb` と入力すると、必要な VI をダウンロードできます。
- HP/Agilent 34401A デジタルマルチメータ（DMM）の計測器ドライバは、ナショナルインスツルメンツの DMM テンプレートドライバとより類似した仕様になりました。このドライバは、LabVIEW 7.*x* 以前で使用した HP34401A ドライバと互換性がありません。LabVIEW 7.*x* HP34401A ドライバとの互換性が必要な場合は、National Instruments Instrument Driver Network（`ni.com/idnet`）からドライバをダウンロードしてください。

ナショナルインスツルメンツの追加のソフトウェアをアップグレードする

LabVIEW 8.0 では NI TestStand 3.5 以降を使用する必要があります。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）にアクセスし、info code として `exd8yy` と入力すると、Upgrade Advisor にアクセスして、NI TestStand 3.5 以降をダウンロードできます。

LabVIEW 8.0 では Logos 4.6 以降を使用する必要があります。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）にアクセスし、info code として `exymjt` と入力すると、最新の Logos をダウンロードできます。

旧バージョンの LabVIEW からのアップデート

LabVIEW を新しいバージョンにアップグレードしても、新バージョンは別のディレクトリにインストールされるため、コンピュータ上にある旧バージョンの LabVIEW には影響がありません。バージョン 5.*x* 以前の LabVIEW は、`labview` ディレクトリにインストールされます。バージョン 6.0 以降の LabVIEW は、`labview` *x.x* ディレクトリにインストールさ

れます。ここで、*x.x* はバージョン番号です。旧バージョンの LabVIEW をアンインストールせずに、LabVIEW 8.0 をインストールできます。

既存バージョンの LabVIEW を入れ替えるには、既存バージョンの LabVIEW をアンインストールして、LabVIEW 8.0 のインストーラを実行し、デフォルトのインストールディレクトリを、旧バージョンの LabVIEW をインストールしたのと同じ `labview` ディレクトリに設定します。

(Windows) コントロールパネルの「アプリケーションの追加と削除」を使用して既存の LabVIEW のバージョンを削除することによっても、既存の LabVIEW のバージョンを LabVIEW 8.0 に置換できます。アンインストーラを実行しても、`labview` ディレクトリにあるユーザが作成したファイルは削除されません。

メモ LabVIEW をアンインストールまたは再インストールする場合、`.llb` ファイルに保存したすべての VI および制御器を含む、`vi.lib` ディレクトリにある `.llb` ファイルがアンインストールされます。VI や制御器を**制御器**パレットや**関数**パレットに追加するには、その VI または制御器を `user.lib` ディレクトリに保存します。

旧バージョンの LabVIEW の環境設定を使用する場合は、旧バージョンがインストールされている `labview` ディレクトリから LabVIEW 環境設定ファイルをコピーします。

注意 LabVIEW 8.0 環境設定ファイルを旧バージョンのファイルと置換する場合は、LabVIEW に追加された環境設定を旧バージョンに上書きします。

LabVIEW 8.0 をインストールした後、LabVIEW 環境設定ファイルを LabVIEW 8.0 ディレクトリにコピーします。

(Windows) 環境設定は `labview.ini` ファイルに保存されます。

(Mac OS) 環境設定は、ホームディレクトリの `Library:Preferences` フォルダ内の `LabVIEW Preferences` ファイルに保存されます。

(Linux) 環境設定は、ユーザのホームディレクトリ内の `.labviewrc` ファイルに保存されます。

旧バージョンの LabVIEW の `user.lib` ディレクトリからのファイルを使用するには、旧バージョンがインストールされている `labview` ディレクトリのファイルをコピーします。LabVIEW 8.0 をインストールしたら、ファイルを LabVIEW 8.0 ディレクトリの `user.lib` ディレクトリにコピーします。

アップグレードおよび互換性の問題

LabVIEW のバージョンにより異なるアップグレードおよび互換性の問題点については、以下のセクションを参照してください。

LabVIEW 7.*x* からのアップグレード

LabVIEW 7.*x* から LabVIEW 8.0 にアップグレードする際、以下のような互換性に関する問題が発生する可能性があります。

サポートされているプラットフォーム

LabVIEW 8.0 では、サポートされているプラットフォームについて以下のように変更されました。

- LabVIEW 7.1 以降では、Windows Me/98/95 はサポートされていません。LabVIEW 8.0 では Windows NT もサポートされていません。
- LabVIEW 8.0 では Windows XP x64 をサポートしますが、現行の NI ドライバの XP x64 に関するサポート限られています。
- LabVIEW 8.0 は、Mac OS X 10.2 以前のバージョンをサポートしていません。
- LabVIEW 8.0 は Sun Solaris をサポートしていません。

システム要件

LabVIEW 7.*x* は、128 MB 以上の RAM を必要としますが、256 MB 以上であることを推奨します。LabVIEW 8.0 は、256 MB 以上の RAM を必要としますが、512 MB 以上であることを推奨します。

LabVIEW 7.*x* は 800 × 600 ピクセルの画面の解像度を必要としますが、1024 × 768 ピクセルの画面の解像度を推奨します。LabVIEW 8.0 には、1024 × 768 ピクセルの画面の解像度が必要です。

(Windows) LabVIEW 7.*x* には、Pentium III 以降または Celeron 600 MHz 以降のプロセッサが必要ですが、Pentium 4 以降のプロセッサを推奨します。LabVIEW 8.0 には、Pentium III または Celeron 866 MHz 以上のプロセッサが必要ですが、Pentium 4 以上のプロセッサを推奨します。

(Linux) LabVIEW 7.*x* には、Pentium III 以降または Celeron 600 MHz 以降のプロセッサが必要ですが、Pentium 4 以降のプロセッサを推奨します。LabVIEW 8.0 には、Pentium III または Celeron 866 MHz 以上のプロセッサが必要ですが、Pentium 4 以上のプロセッサを推奨します。

(Windows) LabVIEW 7.*x* では、LabVIEW の最小インストールには最低 130 MB のディスク容量、完全インストールには 550 MB のディスク容量が必要です。LabVIEW 8.0 では、LabVIEW の最小インストールには最低

900 MB のディスク容量、完全インストールには 1.2 GB のディスク容量が必要です。

(Mac OS) LabVIEW 7.*x* では、LabVIEW の最小インストールには最低 280 MB のディスク容量、完全インストールには 350 MB のディスク容量が必要です。LabVIEW 8.0 では、LabVIEW の最小インストールには最低 500 MB のディスク容量、完全インストールには 600 MB のディスク容量が必要です。

(Linux) LabVIEW 7.*x* では、LabVIEW の最小インストールには最低 200 MB のディスク容量、完全インストールには 300 MB のディスク容量が必要です。LabVIEW 8.0 では、LabVIEW の最小インストールには最低 400 MB のディスク容量、完全インストールには 550 MB のディスク容量が必要です。

(Linux) LabVIEW 7.*x* は、GNU C Library（`glibc`）バージョン 2.1.3 以降が必要ですが、ナショナルインスツルメンツは GNU C Library バージョン 2.2.4 以降を推奨します。LabVIEW 8.0 には、GNU C Library バージョン 2.2.4 以降が必要です。

(Linux) LabVIEW 7.*x* は、Red Hat Linux 7.0 以降、Mandrake Linux 8.0 以降、SuSE Linux 7.1 以降、または Debian Linux 3.0 以降で動作します。LabVIEW 8.0 は、Red Hat Enterprise Linux WS 以降、MandrakeLinux/Mandriva 10.0 以降、または SuSE Linux 9.1 以降で動作します。

カスタムパレットセット

LabVIEW 8.0 はカスタムパレットセットをサポートしていません。カスタムパレットセットを使用せずにパレットセットを編集できます。LabVIEW 8.0 でのパレット変更の詳細については、このドキュメントの「パレット編集の機能強化」のセクションを参照してください。

VI と関数の動作変更

以下の VI および関数の動作は LabVIEW 8.0 で変更されました。

.NET VI と .NET アプリケーション

LabVIEW 8.0 で .NET の関数およびアプリケーションを使用するには、.NET Framework 1.1 Service Pack 1 またはそれ以降が必要です。.NET Framework 1.1 Service Pack 1 をインストールする前に Microsoft .NET Framework 1.1 Hotfix KB886904 を削除する必要があります。

最後に LabVIEW 7.*x* に保存した .NET VI をロードすると、アセンブリファイルがこの VI と同じディレクトリに存在したり、LabVIEW 7.*x* で **ツール→上級→ .NET アセンブリリファレンス**を選択してアセンブリファイルを登録していても、LabVIEW 8.0 は VI が参照するアセンブリを検索することを促すプロンプトを表示することがあります。

解析 VI のアルゴリズム

LabVIEW 7.1 以降では、解析 VI で BLAS/LAPACK アルゴリズムが使用されています。これにより、生成される結果がより正確になりました。LabVIEW 8.0 では、これらの VI は**数学**および**信号処理**パレットにあります。

信号を付加 Express VI

LabVIEW 7.*x* では、「信号を付加（Append Signals）」Express VI の**入力信号 A** が空の場合、または配線されていない場合、単一信号または組み合わせた信号を**入力信号 B** に配線すると、「**付加信号**」出力が空になります。LabVIEW 8.0 では、**入力信号 A** が空または配線されていない場合に、単一信号を**入力信号 B** に配線すると、Express VI は**入力信号 B** を返します。組み合わせた信号のみを**入力信号 B** に配線する場合には、組み合わせた信号で各信号は次の信号に加えられ、結果として 1 つの信号が作成されます。

比較関数

LabVIEW 7.*x* 以前では、比較関数を使用してバリアントデータを比較すると、LabVIEW は 2 つのバリアントの長さを比較して、バリアントをビットごとに比較しました。LabVIEW 8.0 では、バリアントの実際の入力情報をエンコードした入力コード付きのバリアントデータの比較を開始し、その他の入力特有の属性を比較します。

内積 VI

LabVIEW 7.0 では、以下の式を使用し、「内積（Dot Product）」VI で *X* および *Y* の入力ベクトルの内積を計算します。

$$X^{*}Y = \sum_{i=0}^{n-1} x_i y_i$$

LabVIEW 7.1 以降では、「内積（Dot Product）」VI で以下の式を使用して複素数入力の内積を計算します。

$$X^{*}Y = \sum_{i=0}^{n-1} x_i y_i^{*}$$

ここで、y_i^{*} は、y_i の複素共役です。

Easy Text Report VI（Mac OS および Linux）

「Easy Text Report」VI のコネクタペーンが変更されました。LabVIEW 8.0 で、「Easy Text Report」VI が使用されている LabVIEW 7.*x* 以前のバージョンで最後に保存された VI を開くと、サブ VI を右ク

リックして、ショートカットメニューから **Relink To SubVI** を選択する必要があります。

文字列にフォーマット関数

LabVIEW 7.*x* では、`%o` または `%x` の形式指定子の構文要素を「文字列にフォーマット（Format Into String）」関数で使用すると、その入力を文字列に変換する前に、浮動小数点入力を 32 ビット整数に四捨五入します。

LabVIEW 8.0 では、これらの形式指定子の構文要素は入力を文字列に変換する前に、この関数で浮動小数点入力を 64 ビット整数に四捨五入します。

数値結合関数

LabVIEW 7.*x* 以前では、、「数値結合（Join Numbers）」関数は 32 ビット整数入力を 16 ビット整数に強制的に変更し、1 つの 32 ビット整数を作成していました。LabVIEW 8.0 では、「数値結合」関数で 32 ビット整数入力を組み合わせて 1 つの 64 ビット整数を生成できるようになりました。

メモ LabVIEW 7.*x* の VI を LabVIEW 8.0 で開くと、32 ビット整数入力が 16 ビット整数に強制的に変更されます。

文字列 / 数値変換関数

LabVIEW 7.*x* では、数値を 16 進数文字列に変換、数値を 8 進数文字列に変換、および数値を 10 進数文字列に変換関数は、入力を文字列に変換する前に浮動小数点入力を 32 ビット整数に四捨五入します。

LabVIEW 8.0 では、これらの関数は入力を文字列に変換する前に、浮動小数点入力を 64 ビット整数に四捨五入します。ただし、LabVIEW 8.0 で LabVIEW 7.*x* の VI を開く場合は、浮動小数点入力を 32 ビット整数に四捨五入することで互換性と機能を維持します。

VI リファレンスを開く関数

LabVIEW 7.*x* で「VI リファレンスを開く（Open VI Reference）」関数の **VI パス**入力がパスで、メモリ内の VI が同じ名前で存在する場合、メモリ内の VI へのパスが指定したパスと一致しなくても、VI へリファレンスを返します。

LabVIEW 8.0 では、「VI リファレンスを開く」関数の **VI パス**入力が文字列で、**VI パス**がメモリ内の VI の限定されたファイル名に一致する場合にのみ、ターゲット上で VI を開きます。**VI パス**がパスの場合、同じターゲット上の同じパスでメモリ内の VI を検索します。一致するパスで VI を検索できない場合は、指定したパスでディスクから VI をロードしようとします。ファイルが見つからない場合、またはファイルがメモリ内の他の VI と競合する場合、エラーが発生します。限定されたファイル名についての詳細は、この章の「LabVIEW プロジェクトライブラリ」のセクショ

ンを参照してください。ターゲットについての詳細は、この章の「LabVIEW プロジェクト」のセクションを参照してください。

クイックスケール VI

LabVIEW 7.1 以前では、「クイックスケール 1D（Quick Scale 1D）」VI または「クイックスケール 2D（Quick Scale 2D）」VI の **X** 入力がゼロの配列の場合、この VI は **0** および **Y(i)=X(i)/Max|X|** として **max|X|** を、または NaN の配列として **Yij=Xij/Max|X|** を返します。
LabVIEW 8.0 では、「クイックスケール（Quick Scale）」VI の **X** 入力がゼロの配列の場合、この VI は **0** および **Y(i)=X(i)/Max|X|** として **max|X|** を、またはゼロの配列として **Yij=Xij/Max|X|** を返します。

キー読み取り VI

In LabVIEW 7.*x* 以前では、「キー読み取り（Read Key）」VI を使用して、Shift-JIS でエンコードされた日本語のマルチバイト文字の文字を読み取ることができました。1 または <Shift-JIS> を、**マルチバイトエンコーディング**入力に配線する必要があります。LabVIEW 8.0 では、「キー読み取り」VI は、オペレーティングシステムの地域設定を適切なエンコーディングに設定すると、デフォルトでエンコードされた文字列の、マルチバイト文字を読み取ります。

スケール VI

LabVIEW 7.1 以前では、「スケール 1D（Scale 1D）」VI または「スケール 2D（Scale 2D）」VI の **X** 入力がゼロの配列の場合、この VI は **0** として**スケール**、**0** として**オフセット**、NaN の配列として **Y=(X− オフセット)/ スケール**を返します。LabVIEW 8.0 では、スケール VI の **X** 入力がゼロの配列の場合、この VI は **1** として**スケール**、**0** として**オフセット**、ゼロの配列として **Y=(X− オフセット)/ スケール**を返します。

セマフォ VI

LabVIEW 7.*x* では、**エラー入力**パラメータにエラーが含まれている場合でも、「セマフォ解放（Release Semaphore）」VI および「セマフォ取得（Acquire Semaphore）」VI を実行しようとはしませんでした。
LabVIEW 8.0 では、これらの VI は**エラー入力**パラメータにエラーがあっても実行しようとします。ただし、LabVIEW 7.*x* の VI を LabVIEW 8.0 で開くと、LabVIEW 7.*x* の機能が維持されます。

SMTP E メール VI

LabVIEW 7.*x* 以前では、SMTP E メール VI の**文字セット**入力に値を書き込むことで、文字セットを指定できます。これらの VI は、E メールを送信する前にメッセージを UTF-8 形式に暗号化します。SMTP E メール VI の**文字セット**または**字訳**パラメータがなくなりました。

複素数をソート VI

LabVIEW 7.*x* 以前では、「複素数をソート（Sort Complex Numbers）」VI の**メソッド**入力を**振幅**に設定すると、同じ振幅を持つ要素のシーケンスが変更されませんでした。LabVIEW 8.0 では、**メソッド**を**振幅**に設定すると、同じ振幅を持つ要素をまず実部に基づいてソートし、次に虚部に基づいてソートします。

単位ベクトル VI

LabVIEW 7.*x* 以前では、「単位ベクトル（Unit Vector）」VI によって入力ベクトルのノルムが以下の式に基づいて計算されます。

$$||X|| = \sqrt{x_0^2 + x_1^2 + \ldots + x_{n-1}^2}$$

LabVIEW 8.0 では、単位ベクトル VI によって入力ベクトルのノルムが以下の式に基づいて計算されます。

$$||X|| = \left(|x_0|^y + |x_1|^y + \ldots + |x_{n-1}|^y\right)^{\frac{1}{y}}$$

ここで、*X* は入力ベクトル、$||X||$ はノルム、*y* はノルムタイプです。

ユーザ VI

`labview¥help`、`labview¥project`、または `labview¥wizard` ディレクトリに配置する VI は、**ヘルプ**、**ツール**、および**ファイル**メニューにそれぞれ表示されます。LabVIEW 8.0 はこれらのディレクトリに配置する VI をプライベートアプリケーションインスタンスで開くため、LabVIEW 7.*x* では、これらの VI が LabVIEW 8.0 のように動作しないことがあります。

すべてのアプリケーションインスタンスでメモリ内のすべてのユーザ VI のリストを生成するには、`labview¥vi.lib¥Utility¥allVIsInMemory.llb` にある Memory VI の VIMemory Get VI を使用してください。現在のアプリケーションインスタンスへのリファレンスを作成するには、`labview¥vi.lib¥Utility¥allVIsInMemory.llb` にある Get User Application Reference VI を使用してください。アプリケーションインスタンスについての詳細は、このドキュメントの「アプリケーションインスタンスを使用する」のセクションを参照してください。

廃止予定の VI および関数

LabVIEW 8.0 は以下の VI および関数をサポートしていません。

- LabVIEW 7.1 以降では、「多項式実ゼロカウンタ（Polynomial Real Zero Counter）」VI はインストールされません。「多項式実ゼロ（複数）カウンタ（Polynomial Real Zeros Counter）」VI を代用してください。

- LabVIEW 7.1 以降では、PPC VI はインストールされません。TCP VI を代用してください。
- LabVIEW 8.0 は「QR 因子分解（QR Factorization）」VI をサポートしていません。「QR 分解（QR Decomposition）」VI を代用してください。
- LabVIEW 8.0 では、「レーベンバーグ・マルカート（Levenberg Marquardt）」VI または「非線形レーベンバーグ・マルカートフィット（Nonlinear Lev-Mar Fit）」VI をサポートしていません。「非線形カーブフィット（Nonlinear Curve Fit）」VI を代用してください。
- LabVIEW 8.0 では、**関数**パレット上に「VISA ステータス記述（VISA Status Description）」関数がありません。「シンプルエラー処理 (Simple Error Handler)」VI または「一般エラー処理 (General Error Handler)」VI を代用してください。
- LabVIEW 8.0 は、「カイ 2 乗分布（Chi Square Distribution）」、「F 分布（F Distribution）」、「正規分布（Normal Distribution）」、および「T 分布（T Distribution）」VI をサポートしません。代わりに、「連続 CDF（Continuous CDF）」VI のカイ 2 乗、F、正規、およびステューデント t インスタンスをそれぞれ使用してください。
- LabVIEW 8.0 は、「逆カイ 2 乗分布（Inv Chi Square Distribution）」、「逆 F 分布（Inv F Distribution）」、「逆正規分布（Inv Normal Distribution）」、および「逆 T 分布（Inv T Distribution）」VI をサポートしません。代わりに、「連続 CDF」VI のカイ 2 乗、F、正規、およびステューデント t インスタンスをそれぞれ使用してください。
- LabVIEW 8.0 では、「1D 線形評価（1D Linear Evaluation）」VI および「2D 線形評価（2D Linear Evaluation）」VI は、**関数**パレットから外されました。「線形評価（Linear Evaluation）」VI を代用してください。
- LabVIEW 8.0 では、「1D 多項式評価（1D Polynomial Evaluation）」VI および「2D 多項式評価（2D Polynomial Evaluation）」VI は、**関数**パレットから外されました。「多項式評価（Polynomial Evaluation）」VI を代用してください。
- LabVIEW 8.0 では、「1D 直交座標から極座標へ変換（1D Rectangular to Polar）」VI および「1D 極座標から直交座標へ変換（1D Polar to Rectangular）」VI が**関数**パレットから外されました。「直交座標を極座標に変換（Re/Im To Polar）」関数および「極座標を直交座標に変換（Polar To Re/Im）」関数を代用してください。
- LabVIEW 8.0 では、「高調波アナライザ（Harmonic Analyzer）」VI が**関数**パレットから外されました。**THD** または**コンポーネントレベル**出力を測定するには「高調波ひずみ解析（Harmonic Distortion Analyzer）」VI を使用するか、**SINAD** または **THD+ ノイズ**出力を測定するには「SINAD アナライザ（SINAD Analyzer）」VI を使用してください。

- LabVIEW 8.0 では、「ネットワーク関数（平均）（Network Functions (avg)）」VI は**関数**パレットから外されました。「周波数応答関数 (振幅 - 位相)（Frequency Response Function (Mag-Phase)）」、「周波数応答関数 (実 - 虚)（Frequency Response Function (Real-Im)）」、「クロススペクトル (振幅 - 位相)（Cross Spectrum (Mag-Phase)）」、または「クロススペクトル (実 - 虚)（Cross Spectrum (Real-Im)）」VI を使用してください。
- LabVIEW 8.0 では、「パルスパラメータ（Pulse Parameters）」VI が**関数**パレットから外されました。「過度計測（Transition Measurements）」VI を代用して**スルーレート**、**持続時間**、**オーバーシュート**または、**プレシュート**出力を測定し、「パルス計測（Pulse Measurements）」VI を使用して**周期** m **パルス持続時間**、または**デューティサイクル**出力を測定し、「振幅とレベル（Amplitude and Levels）」VI を使用して**振幅**、**高状態レベル**、または**低状態レベル**出力を測定します。
- LabVIEW 8.0 では、「伝達関数（Transfer Function）」VI が**関数**パレットから外されました。「周波数応答関数 (振幅 - 位相)」または「周波数応答関数 (実 - 虚)」VI を代用してください。
- LabVIEW 8.0 では、「NI DIAdem レポートウィザード（NI DIAdem Report Wizard）」Express VI が**関数**パレットに表示されなくなりました。「NI DIAdem レポート（NI DIAdem Report）」Express VI を代用してください。「NI DIAdem レポート」Express VI の詳細については、このドキュメントの「DIAdem レポート Express VI」セクションを参照してください。

メモ NI DIAdem レポート Express VI を使用するためには、DIAdem 9.1 Service Pack 2 以降がインストールされていなければなりません。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）にアクセスし、info code として `exf5ty` と入力すると、DIAdem の評価版をダウンロードまたは DIAdem を購入することができます。

- LabVIEW 8.0 では、VISA リソース名定数および IVI 論理名定数は**関数**パレットから外されました。VISA リソース名を指定するには、VISAVI の **VISA リソース名**入力を使用してください。IVI 論理名を指定するには、計測器を初期化する適切なドライバ VI の適切な入力を使用してください。
- LabVIEW 8.0 では、エラーリング定数が**関数**パレットから外されました。32 ビット符号付き整数定数を代用して、希望するエラーコードを入力します。
- **(Windows)** LabVIEW 8.0 では、サウンド VI は**サウンド**パレットに表示されますが、LabVIEW 7.*x* では**関数**パレットにありませんでした。LabVIEW 8.0 ではサウンド VI を代用してください。LabVIEW 7.*x* に付属されたサンプルは、LabVIEW 8.0 には付属されません。

- **(Mac OS および Linux)** LabVIEW 8.0 のサウンド VI は、LabVIEW 7.1 のサウンド VI と一緒に付属されます。LabVIEW 7.*x* に付属されたサンプルは、LabVIEW 8.0 には付属されません。

ファイル I/O VI および関数

LabVIEW 8.0 では、「ファイルから文字を読み取り（Read Characters From File）」VI が**関数**パレットから外されました。「テキストファイルから読み取り（Read from Text File）」関数を代用してください。

LabVIEW 8.0 では、「開く / 作成 / 置換ファイル（Open/Create/Replace File）」VI が**関数**パレットから外されました。「開く / 作成 / 置換ファイル（Open/Create/Replace File）」関数を代用してください。以下の関数には、LabVIEW 7.*x* 以前の「開く / 作成 / 置換ファイル（Open/Create/Replace File）」VI の一部の機能が含まれます。

- ファイルのサイズを調べるには、「ファイルサイズを取得（Get File Size）」関数を使用します。
- 「ファイルダイアログ（File Dialog）」Express VI を使用して、ファイルを参照できるダイアログボックスの開始パス、ファイルパターン、およびファイルまたはディレクトのデフォルト名を指定します。
- 「Refnum をパスに変換（Refnum to Path）」関数を使用して、リファレンスをパスに変換します。
- 「バイナリファイルに書き込み（Write to Binary File）」関数を使用して、プラットフォームに依存しないテキストファイルまたは他のタイプのバイナリファイルを作成し、「バイナリファイルから読み取り（Read from Binary File）」関数を使用して、結果のバイナリファイルを読み取ります。

LabVIEW 8.0 では、「ファイルから読み取り（Read File）」および「ファイルに書き込み（Write File）」関数が**関数**パレットから外されました。「バイナリファイルから読み取り」および「バイナリファイルに書き込み」関数を代用してください。

LabVIEW 8.0 では、「文字をファイルに書き込み（Write Characters To File）」VI が**関数**パレットから外されました。「テキストファイルに書き込み（Write to Text File）」関数を代用してください。

LabVIEW 8.0 では、「アクセス権（Access Right）」VI が**関数**パレットから外されました。「許可を取得（Get Permissions）」および「許可を選択（Set Permissions）」関数を代用してください。

LabVIEW 8.0 では、「EOF」関数が**関数**パレットから外されました。「ファイルサイズを取得（Get File Size）」および「ファイルサイズを設定（Set File Size）」関数を代用してください。

LabVIEW 8.0 では、「ディレクトリのリスト（List Directory）」VI が**関数**パレットから外されました。「リストフォルダ（List Folder）」関数を代用してください。

LabVIEW 8.0 では、「範囲をロック（Lock Range）」関数が**関数**パレットから外されました。「アクセスを拒否（Deny Access）」関数を代用してください。

ブロックダイアグラムに「新規ディレクトリ（New Directory）」関数を含む LabVIEW 7.*x* で作成した VI を開くと、LabVIEW 8.0 はその関数を「フォルダを作成（Create Folder）」関数に置換します。**パス**入力で指定したフォルダが存在しない場合は、「フォルダを作成」関数は、「新規ディレクトリ」関数と同じように、エラーを返す代わりにフォルダを作成します。

LabVIEW 8.0 では、「シーク（Seek）」関数が**関数**パレットから外されました。「ファイル位置を取得（Get File Position）」および「ファイル位置を設定（Set File Position）」関数を代用してください。

LabVIEW 8.0 では、「タイプと作成者（Type and Creator）」関数が**関数**パレットから外されました。「タイプと作成者を取得（Get Type and Creator）」および「タイプと作成者を設定（Set Type and Creator）」関数を代用してください。

LabVIEW 8.0 では、「ボリューム情報（Volume Info）」VI が**関数**パレットから外されました。「ボリューム情報を取得（Get Volume Info）」関数を代用してください。

LabVIEW 8.0 では、「ファイルを開く（Open File）」および「新規ファイル（New File）」関数が**関数**パレットから外されました。「ファイルから一行読み取り（Read Lines From File）」VI は**関数**パレットには表示されませんが、互換性のために LabVIEW に付属されます。

LabVIEW 8.0 では、「I16 ファイルから読み取り（Read From I16 File）」、「SGL ファイルから読み取り（Read From SGL File）」、「I16 ファイルに書き込み（Write To I16 File）」、および「SGL ファイルに書き込み（Write To SGL File）」VI は**関数**パレットにありません。「バイナリファイルから読み取り」および「バイナリファイルに書き込み」VI を代用してください。

廃止された VI を検索する

旧バージョンの LabVIEW で作成した VI にある廃止された VI を検索するには、以下の手順に従ってください。

1. 検索する VI をメモリにロードします。
2. **検索**ダイアログボックスを表示するには、**編集→検索して置換**を選択します。
3. **テキスト**ボタンをクリックします。

4. **アプリケーションインスタンス**プルダウンメニューから、検索する VI を含むアプリケーションインスタンスを選択します。アプリケーションインスタンスについての詳細は、このドキュメントの「アプリケーションインスタンスを使用する」のセクションを参照してください。
5. **検索範囲**プルダウンメニューから**アプリケーションインスタンス中のすべての VI** を選択します。
6. **階層ウィンドウ**チェックボックスをオンにします。
7. **検索**ボタンをクリックします。

プロパティ、メソッド、イベントの動作変更

以下のプロパティ、メソッド、イベントの動作は LabVIEW 8.0 で変更されました。

アプリケーションプロパティとメソッド

LabVIEW 8.0 では、一部のアプリケーションプロパティおよびメソッドの動作は、属するアプリケーションインスタンスによって異なります。たとえば、アプリケーション : メモリ内のすべての VI プロパティの動作は、使用するアプリケーションインスタンスによって異なります。このプロパティは、プロパティと同じアプリケーションインスタンスでメモリ内のすべての VI のリストを返します。ただし、アプリケーション : ディレクトリパス I プロパティの動作は、使用するアプリケーションインスタンスに依存しません。このプロパティは、アプリケーションがあるディレクトリへの絶対パスを返します。この情報は、各インスタンスによって変更しません。

アプリケーションインスタンスについての詳細は、このドキュメントの「アプリケーションインスタンスを使用する」のセクションを参照してください。

フロントパネル : 開く メソッド

LabVIEW 7.0 の FP を開くメソッドは、LabVIEW 7.1 で旧フロントパネルを開くに名前変更されました。LabVIEW 7.1 は、フロントパネルがすでに開いていると、エラーを返さない異なる FP を開くメソッドを含みます。LabVIEW 7.1 の FP を開くメソッドは、LabVIEW 8.0 でフロントパネル : 開くと名前変更されました。LabVIEW 7.0 から旧フロントパネルを開くメソッドを使用している VI がある場合は、メソッドをフロントパネル : 開くメソッドに置換してください。

キーダウンイベントとキー繰り返しイベント

VI のキーダウン、キーダウン？、キー繰り返し、キー繰り返し？の各イベントおよび制御器の **VKey** データフィールドは、英数字セクションの <Return> キーおよび数値キーパッドの <Enter> キーで異なる値が入力されます。LabVIEW 7.*x* 以前では、<Enter> キーまたは <Return> キーがイ

ベントの 1 つを生成すると、**VKey** データフィールドで **<Enter>** を返します。LabVIEW 8.0. では、<Enter> キーまたは <Return> キーがイベントの 1 つを生成すると、**VKey** データフィールドで、それぞれ、**<Enter>** または **<Return>** を返します。

(Mac OS) LabVIEW 8.0 では、ショートカットメニューの <Control>- クリックのみを受け入れるようになりましたが、<Command>- クリックのキーの組み合わせは受け入れません。イベントストラクチャで動作をエミュレートする場合は、VI を変更して新規の動作をエミュレートします。

リストボックスプロパティ

LabVIEW 7.*x* 以前では、リストボックスプロパティの最上行プロパティをリストボックスの一番下の項目より下の行に設定すると、表示されている一番下の行に強制的に変更されます。LabVIEW 8.0 では、リストボックスの表示項目数は、このプロパティに配線できる行数を制限しません。

LabVIEW 8.0 は、単数列リストボックスのダブルクリックプロパティをサポートしていません。ダブルクリックされた行を取得メソッドを代用してください。

所有 VI プロパティ

LabVIEW 7.*x* 以前では、所有 VI プロパティはオブジェクトが属する VI へのリファレンスを返しました。このリファレンスは、メモリ内の VI を維持します。LabVIEW 8.0 では、所有 VI プロパティが返すリファレンスは、メモリ内の VI を維持しません。VI がメモリから削除されると、このリファレンスは無効になります。リファレンスを閉じるまでメモリに残る VI へのリファレンスを取得するには、「VI リファレンスを開く（Open VI Reference）」関数を使用します。

テキストプロパティ

LabVIEW 7.*x* 以前では、テキストプロパティは標準表示で文字列を返します。LabVIEW 8.0 では、テキストプロパティは文字列をフロントパネルオブジェクトと同じテキスト表示で返します。たとえば、パスワード表示に文字列制御器を表示すると、テキストプロパティはパスワード表示で文字列を返します。

ツリー制御器プロパティ

LabVIEW 7.*x* 以前では、アクティブセルプロパティ : セルサイズ : 高さおよびアクティブセルプロパティ : セルサイズ : 幅プロパティは、ツリー制御器内に各行に 17 ピクセルを返しました。LabVIEW 8.0 では、アクティブセル : セルサイズ : 高さおよびアクティブセル : セルサイズ : 幅プロパティは、ツリー制御器内に各行に 16 ピクセルを返します。

VI 文字列メソッド

エクスポート VI 文字列メソッドを使用して前のバージョンの LabVIEW からエクスポートした文字列は、VI 文字列 : インポートメソッドを使用する場合、正しく LabVIEW 8.0 にインポートされない場合があります。

廃止予定のプロパティ、メソッド、イベント

LabVIEW 8.0 は以下のプロパティ、メソッド、イベントをサポートしていません。

カーソルプロパティ

LabVIEW 8.0 は、カーソルロックスタイルプロパティをサポートしません。カーソルモードプロパティを代用してください。

リストボックス、表、デジタル表、ツリー制御器プロパティ、およびイベント

LabVIEW 8.0 は、複数列リストボックスのセルの前景色プロパティをサポートしていません。アクティブ セル プロパティ : セルフォント : カラープロパティを代用してください。

LabVIEW 8.0 は、表またはデジタル表のセルの前景色プロパティをサポートしていません。表およびデジタル表については、アクティブセル : セルフォント : カラープロパティを代用してください。

LabVIEW 8.0 は、ツリー制御器のアクティブ セル プロパティ : 前景カラープロパティをサポートしていません。アクティブ セル プロパティ : セルフォント : カラープロパティを代用してください。

LabVIEW 8.0 は、ドラッグ、ドラッグ ?、ドロップ、ドロップ ? イベントをツリー制御器クラスでサポートしません。ドラッグ終了、ドラッグ入る、ドラッグ出る、ドラッグオーバー、ドラッグソースアップデート、ドラッグ開始、ドラッグ開始 ?、ドロップイベントを制御器クラスで代用してください。

名前付き数値プロパティ

LabVIEW 8.0 は、名前付き数値オブジェクトについて、名前付き数値カラー、名前付き数値カラー : 背景色、名前付き数値カラー : テキストカラーの各プロパティをサポートしていません。テキストカラー、テキストカラー : 背景色、テキストカラー : テキストカラーの各プロパティをそれぞれ代用してください。

パネルプロパティ

LabVIEW 8.0 は、パネルクラスのカラープロパティをサポートしません。このプロパティを LabVIEW 8.0 で使用すると、プロパティは左上隅のペーンのみに適用されます。ペーンカラープロパティをペーンクラスで代用してください。

サブパネルプロパティ

LabVIEW 8.0 では、サブパネルでサブ VI のペーンを使用し、サブパネル制御器のスクロールバー表示を構成して、サブパネル制御器のフロントパネルをスケールします。

LabVIEW 8.0 は、サブパネル制御器の X スクロールバー表示プロパティをサポートしていません。ペーンの水平スクロールバー表示プロパティを代用してください。

LabVIEW 8.0 は、サブパネル制御器の Y スクロールバー表示プロパティをサポートしていません。ペーンの垂直スクロールバー表示プロパティを代用してください。

LabVIEW 8.0 は、サブパネル制御器のパネルをスケールプロパティをサポートしていません。ペーンのスケールモードを設定メソッドを代用してください。

VI プロパティ、メソッド、イベント

LabVIEW 8.0 はフロントパネルウィンドウ : 自動中央揃えプロパティをサポートしていません。フロントパネル : 中心メソッドを代用してください。

LabVIEW 8.0 はフロントパネルウィンドウ : 画面サイズに合わせるプロパティをサポートしていません。フロントパネルウィンドウ : 状態プロパティを代用してください。

LabVIEW 8.0 は、VI クラスのフロントパネルウィンドウ : 基点プロパティをサポートしません。このプロパティを LabVIEW 8.0 で使用すると、プロパティは左上隅のペーンのみに適用されます。ペーンクラスの基点プロパティを代用してください。

LabVIEW 8.0 は、VI クラスのフロントパネルウィンドウ : スクロールバーを表示プロパティをサポートしません。このプロパティを LabVIEW 8.0 で使用すると、プロパティは左上隅のペーンのみに適用されます。ペーンクラスの水平スクロールバー表示および垂直スクロールバー表示プロパティを代用してください。

LabVIEW 8.0 は、VI クラスのフロントパネルスケールモードを取得またはフロントパネルスケールモードを設定メソッドをサポートしません。これらのメソッドを LabVIEW 8.0 で使用すると、メソッドは左上隅のペーンのみに適用されます。ペーンクラスのスケールモードを取得およびスケールモードを設定メソッドを代用してください。

LabVIEW 8.0 は、VI クラスのマウスダウン、マウスダウン ?、マウス境界内、マウス境界外、マウス移動、マウスアップイベントをサポートしません。ペーンクラスのマウスダウン、マウスダウン ?、マウス境界内、マウス境界外、マウス移動、マウスアップイベントを代用してください。

アプリケーション項目のタグ

以下のアプリケーションタグは、LabVIEW から削除されました。

- `APP_BUILD_STANDALONE_APP`
- `APP_DN_ASSEMBLY_REFS`
- `APP_EDIT_VI_LIBRARY`
- `APP_SAVE_WITH_OPTIONS`
- `APP_SHOW_CLIPBOARD`
- `APP_SRC_CODE_CTRL`
- `APP_SWITCH_EXEC_TARGET`
- `APP_UPDATE_VXI`
- `APP_VIEW_PRINTED_MANUALS`

前のバージョンの LabVIEW で保存され、削除されたタグが含まれているランタイムメニュー（`.rtm`）ファイルを使用すると、LabVIEW 8.0 は、**メニューエディタ**ダイアログボックスでファイルを保存すると、自動的に `.rtm` ファイルからタグを削除します。削除されたアプリケーション項目タグは、LabVIEW によって予約され、ユーザタグとして使用できません。

HiQ のサポート

ナショナルインスツルメンツは、LabVIEW 8.0 で HiQ 機能を提供していません。アプリケーションで HiQ VI を使用する場合、それを数学 VI および信号処理 VI と置換することを検討してください。数学 VI および信号処理 VI の使用については、『LabVIEW ヘルプ』を参照してください。

エラーリストウィンドウ

LabVIEW 7.*x* 以前では、**エラーリスト**ウィンドウの **VI リスト**にメモリ内にあるすべての VI のエラーが表示されます。LabVIEW 8.0 では、**エラーリスト**ウィンドウの**エラーがある項目**セクションに、VI およびライブラリなど、メモリ内にあるすべての項目のエラーが表示されます。2 つ以上の項目が同じ名前の場合、このセクションは各あいまいな項目について特定のアプリケーションインスタンスを表示します。アプリケーションインスタンスについての詳細は、このドキュメントの「アプリケーションインスタンスを使用する」のセクションを参照してください。

VI 文字列ファイルの構文

LabVIEW 8.0 では、**ツール→上級→文字列をインポート**で VI 文字列ファイルをインポートするか、または VI 文字列 : インポートメソッドを使用すると、`<GROUPER></GROUPER>` という新規のタグセットを検索します。このタグセットは一緒にグループ化されたフロントパネルオブジェクトを示します。したがって、LabVIEW 8.0 では、旧バージョンの LabVIEW で保存した VI 文字列ファイルをインポートすることはできません。

LabVIEW 7.1 以前では、プライベートデータの <ITEMS> セクションにリストボックスの文字列をリストします。LabVIEW 8.0 では、プライベートデータの <ITEMS> セクションにリストボックスの文字列をリストします。また、LabVIEW 7.1 以前では、リストボックスで 1 種類のフォントしか使用できず、そのフォントはプライベートデータの <LBLABEL> セクションにリストされます。LabVIEW 8.0 では、リストボックスで複数のフォントが使用できるようになり、それらのフォントはプライベートデータの <CELL_FONTS> セクションにリストされます。

LabVIEW 7.1 以前では、複数列リストボックスの文字列はデフォルトデータ内にリストされます。ただし、複数列リストボックスのデフォルトデータは整数か整数の配列です。LabVIEW 8.0 では、複数列リストボックスの文字列はプライベートデータ内にリストされます。

LabVIEW 7.1 以前では、ツリー制御器の文字列とフォントはエクスポートできませんでした。LabVIEW 8.0 では、ツリー制御器の文字列とフォントを、リストボックスや複数列リストボックスと同様の形式でエクスポートできます。

LabVIEW 8.0 では、エクスポートファイルの各行にプライベートデータとデフォルトデータの 2 つのタグ以上は含まれません。また、ネスティングレベルごとに項目がインデントされます。

VI 文字列ファイルを LabVIEW 8.0 形式に変換するには、以下の手順に従ってください。

1. 旧バージョンの LabVIEW にある VI 文字列ファイルをインポートします。
2. VI を保存します。
3. LabVIEW 8.0 で VI をロードします。
4. LabVIEW 8.0 形式で VI 文字列ファイルを保存するには、**ツール→上級→文字列をエクスポート**を選択します。

LabVIEW 7.*x* からタイプデスクリプタデータを双方向に変換する

タイプデスクリプタを保存する形式が LabVIEW 8.0 で変更されました。LabVIEW 7.*x* は、16 ビットの平坦化表記でタイプデスクリプタを保存します。LabVIEW 8.0 は、32 ビットの平坦化表記でタイプデスクリプタを保存します。この変更はタイプデスクリプタの 64KB サイズの制限を撤廃します。

LabVIEW 8.0 には、LabVIEW 7.*x* で書き込まれたタイプデスクリプタを読み取り、LabVIEW 7.*x* で読み取ることのできるタイプデスクリプタを書き込むメカニズムが備わっています。「文字列に平坦化（Flatten To String）」関数には **7.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。関数を右クリックしてこのメニュー項目を選択すると、この関数は、入力データを LabVIEW 7.*x* で書き込まれたデータのように処理でき

ます。**7.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場合に、**データ文字列**出力が配線されていると、LabVIEW 8.0 ではその関数上に赤い `7.x` グリフを描画して LabVIEW 7.*x* 形式に、あるいは LabVIEW 7.x 形式からデータを変換中であることを示します。データ変換を回避するには、**7.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チェックマークを外します。

LabVIEW 8.0 では、LabVIEW 7.*x* 以前のバージョンで保存した VI をロードすると、「文字列に平坦化（Flatten To String）」関数で **7.x データの変換**属性が自動的に設定されます。関数は、LabVIEW 7.*x* 以前と同じように操作されます。VI に LabVIEW 8.0 のタイプデスクリプタ形式を使用する場合は、「文字列に平坦化」関数を右クリックし、ショートカットメニューから **7.x データの変換**を選択してチェックマークを外します。LabVIEW 7.*x* 以前のバージョンで書き込まれたデータを含むファイルを VI で操作する必要がなく、LabVIEW 7.*x* 以前のバージョンで実行している VI にデータを送受信することがない場合には、LabVIEW 8.0 のタイプデスクリプタ形式を使用します。LabVIEW の今後のバージョンでは、前回のタイプデスクリプタ形式のサポートは継続されない可能性があります。

NaN 文字列を整数タイプへ変換する（Windows）

LabVIEW 7.*x* では、`NaN` を明示的または暗示的に整数に変換すると、その値はその整数のデータタイプの最小値となります。たとえば、`NaN` を符号付き 16 ビット整数に変換すると、16 ビット符号付き整数では最小値である −32,768 となります。

LabVIEW 8.0 では、`NaN` を明示的または暗示的に整数に変換すると、その値はその整数のデータタイプの最大値となります。たとえば、`NaN` を符号付き 16 ビット整数に変換すると、16 ビット符号付き整数では最大値である 32,767 となります。

ケーススストラクチャに配線されている定数

LabVIEW 7.*x* 以前では、定数をケーススストラクチャに配線し、実行しないケースにサブ VI を配置することによって、サブ VI をメモリに駐在しておけます。例えば、ケーススストラクチャに TRUE 定数を配線して、サブ VI をケーススストラクチャの FALSE ケースに配置する場合、呼び出す VI と一緒にサブ VI がロードされます。LabVIEW 8.0 では、実行しないすべてのコードが削除されます。したがって、ケーススストラクチャに配線された定数がある VI を、旧バージョンの LabVIEW で保存して LabVIEW 8.0 でロードすると、定数が非表示の制御器に変更され、旧バージョンの LabVIEW からの動作が維持されます。

オペレーティングシステムのメッセージを遅延する

LabVIEW 7.*x* では、LabVIEW は .NET および ActiveX イベントを処理するたのコールバック VI を実行中に、オペレーティングシステムのメッセージを処理しました。LabVIEW 8.0 では、LabVIEW は、コールバック VI の実行が停止するまで、またはユーザがモーダルダイアログボックスをロードするまで、オペレーティングシステムのメッセージの処理を遅延します。この遅延によって、コールバック VI が割り込みなしで実行され、他のイベントでイベントを発行してデットロック状態になる事態を回避できます。

コールバック VI から非モーダルダイアログボックスに対して同期を取った呼び出しを行うことはできません。非モーダルダイアログボックスで VI を実行メソッドを呼び出して、メソッドの**終了まで待機**入力に FALSE のブール定数を書き込むことで、コールバック VI から非モーダルダイアログボックスを非同期に呼び出す必要があります。

LabVIEW 7.*x* では、LabVIEW は DLL または共有ライブラリ関数を実行中に、オペレーティングシステムのメッセージを処理しました。LabVIEW 8.0 では、LabVIEW は、DLL 関数への呼び出しが終わるまで、またはユーザがモーダルダイアログボックスを DLL からロードするまで、オペレーティングシステムのメッセージの処理を遅延します。この遅延によって、DLL 関数が割り込みなしで実行し、LabVIEW が DLL 関数が実行中に同じ DLL を呼び出して、デットロック状態が発生することを避けることができます。

このデフォルトの動作を使用すると、DLL の実行中に非モダールダイアログボックスへの同期の呼び出しを行うことができません。非モーダルダイアログボックスで VI を実行メソッドを呼び出して、メソッドの**終了まで待機**入力に FALSE のブール定数を書き込むことで、DLL から非モーダルダイアログボックスを非同期に呼び出す必要があります。

作成する DLL でオペレーティングシステムのメッセージを遅延するかどうかを選択できます。**マイ DLL プロパティ**ダイアログボックスの**上級**ページで、**共有ライブラリでオペレーティングシステムメッセージを遅延**チェックボックスをオフにして、DLL の実行中にオペレーティングシステムのメッセージを処理します。

Resource Manager（Mac OS）

LabVIEW 7.*x* 以前では、Macintosh リソースファイルの読み取り / 書き込み機能（マニュアルへの記載なし）が使用できます。LabVIEW 8.0 では、これらのメソッドは存在しません。したがって、これらのマニュアルに記載されていなかった機能を利用したユーティリティは使用できません。Macintosh リソースの VI からの読み取りと書き込みはできなくなりました。

1 および 2 ボタンダイアログボックス

LabVIEW 7.*x* 以前では、1 ボタンダイアログまたは 2 ボタンダイアログを表示している VI をプログラムでは中止できません。LabVIEW 8.0 では、これらのダイアログボックスを表示している VI を VI を中断メソッドを使用してプログラムで中止できます。

プロパティおよびインポークノード

LabVIEW 7.*x* のカーソル凡例から暗示的にリンクされたプロパティノードまたはインポークノードを作成した場合に、LabVIEW 8.0 で VI を開くと、LabVIEW はノードを削除します。

共有ライブラリを更新する

LabVIEW 7.*x* 以前で `labview.lib` にリンクされた共有ライブラリ（DLL）を作成した場合は、LabVIEW では `labviewv.lib` に代わりにリンクします。LabVIEW 8.0 での共有らライブラリの使用の詳細については、このドキュメントの「複数のバージョンの LabVIEW で共有ライブラリを使用する」のセクションを参照してください。

印刷の余白値

LabVIEW 7.*x* 以前では、**オプション**ダイアログボックスの**印刷**ページの**余白**オプションでは、余白値にセンチメートルが使用されました。LabVIEW 8.0 の**余白**オプションでは、余白値にミリメートルが使用されます。

LabVIEW 6.*x* からのアップグレード

LabVIEW 6.*x* から LabVIEW 8.0 にアップグレードする際、以下のような互換性に関する問題が発生する可能性があります。アップグレードの際に起こる可能性のある他の問題点については、本書の「LabVIEW 7.*x* からのアップグレード」のセクションを参照してください。

各バージョンの新機能および変更点については、LabVIEW のバージョン 6.*x* から 8.0 までの各バージョンの『LabVIEW アップグレードノート』（`ni.com/jp/manuals`）を参照してください。

波形データタイプの変更点

LabVIEW 7.0 では、波形のデータタイプは、倍精度浮動小数点数ではなく **t0** コンポーネントのタイムスタンプのデータタイプを使用します。データを、LabVIEW 6.*x* のデータタイプに関する情報を含めずに波形データタイプとしてファイルに保存すると、LabVIEW 7*x* 以降でそのデータを取り出そうとするとエラーが発生する可能性があります。

LabVIEW 7.*x* 以降では、「ファイルから波形を読み取り（Read Waveform from File）」VI で、ファイル内の古い波形データタイプ形式を新しい形式に変換します。この VI は、変換を実行するかどうかを尋ね

るダイアログボックスを表示します。LabVIEW ランタイムエンジンでは、「ファイルから波形を読み取り」VI によるこの変換が行われず、エラーが返されます。LabVIEW 6.*x* から LabVIEW 7.*x* への波形データの移行の詳細については、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）にアクセスし、info code の `exd9zq` を入力して情報を参照してください。

シリアル互換性 VI

LabVIEW 7.*x* 以降では、シリアル互換性 VI は**関数**パレットには表示されません。VXI デバイスと通信する VI を作成するには、VISA VI および関数を使用します。

LabVIEW 7.*x* 以降では、LabVIEW はオペレーティングシステムのシリアルドライバと通信に、`serpdrv` ドライバを使用しません。LabVIEW には、VISA をベースにした互換性のある VI が含まれています。新規アプリケーションでは、VISA およびシリアル VI および関数を使用してシリアルデバイスを制御するようにしてください。旧バージョンの LabVIEW で作成したシリアル VI を含むすべての VI は、LabVIEW 7.1 以降でも動作します。

ポート番号のポートへのマッピングを再構成した場合は、それらのポートへのマッピングを指定する必要があります。シリアルポートのマッピングを指定するには、`labview¥vi.lib¥Instr¥_sersup.llb` にある「シリアルエイリアス設定（set serial alias ports）」VI を使用します。文字列配列を VI の **VISA エイリアス**入力に配線して、使用するポート名を入力配列に入力します。配列の各要素はポートに対応しています。たとえば、ポート 0 が VI エイリアスの MySerialPort にマップされる場合は、**VISA エイリアス**入力配列の最初の要素に `MySerialPort` と入力します。「VISA シリアルポート構成（VISA Configure Serial Port）」VI を呼び出す前に、「シリアルエイリアス設定」VI を呼び出す必要があります。

VISA VI および関数を使用してシリアル計測器を制御するサンプルについては、`labview¥examples¥instr¥smplserl.llb` を参照してください。

ループのデフォルトデータ

LabVIEW 6.0 以前では、ループが実行されない場合、For ループは未定義のデータを生成します。LabVIEW 6.1 以降では、For ループのカウント端子（N 端子）に `0` を配線した場合や、空の配列を入力として For ループに配線して自動指標付けを使用にした場合、For ループはデフォルトデータを生成します。ただし、これは出力のトンネルが指標付け不使用の場合です。

リモートフロントパネルライセンス

LabVIEW 開発システムとアプリケーションビルダには、1 つのクライアントがリモートでフロントパネルを表示および制御することができるリモートフロントパネルライセンスが含まれています。LabVIEW プロ

フェッショナル開発システムには、5 つのクライアントがフロントパネルを表示および制御することができるリモートフロントパネルライセンスが含まれています。

リモートフロントパネルライセンスをアップグレードすると、より多くのクライアントをサポートすることができます。

マルチスレッド割り当て

LabVIEW 7.1 以降では、LabVIEW 7.1 以前のバージョンと比較してより多くの VI 実行用のスレッドが割り当てられるようになりました。そのため、呼び出した DLL が実際には再入可能でない場合に「ライブラリ関数呼び出しノード（Call Library Function Node）」を誤って「再入可能」に設定すると、マルチスレッドでエラーが発生することがあります。「ライブラリ関数呼び出しノード」および再入可能の詳細については、『LabVIEW ヘルプ』を参照してください。

LabVIEW でスレッドの割り当て方法を変更するには、`labview¥vi.lib¥Utility¥sysinfo.llb` にある threadconfig VI を使用します。VI の再入可能を無効にするには、**ファイル→ VI プロパティ**を選択して、**カテゴリ**プルダウンリストから**実行**を選択して、**再入実行**チェックボックスをオフにすることもできます。

スレッド割り当ての詳細については、『LabVIEW ヘルプ』を参照してください。

計測器ドライバ

LabVIEW 7.*x* 以降の LabVIEW パッケージには、計測器ドライバを含む LabVIEW 計測器ドライバライブラリ CD は含まれていません。計測器ドライバは、ナショナルインスツルメンツの Instrument Driver Network（`ni.com/idnet`）からダウンロードできます。ナショナルインスツルメンツのデバイスドライバ CD には、NI-DAQ、NI-VISA、および他のナショナルインスツルメンツドライバが収録されています。

単位および変換ファクタ

LabVIEW 7.*x* 以降では、「複合演算（Compound Arithmetic）」関数を使用した後に、「単位変換（Convert Unit）」関数を使用して余分な端子を削除する必要がありません。

LabVIEW 7.1 以降の単位変換ファクタは、米国商務省標準技術局（NIST）発行の *Guide for the Use of the International System of Units（SI）*でのガイドラインにさらに準拠しています。また、カロリーの単位はカロリー（熱）となり、馬力は馬力（電気）となりました。これらの単位の略語は以前と同じです。次の表は、LabVIEW 6.1 と 8.0 での単位変換ファクタの変更の詳細を示しています。

表 1　単位および変換ファクタ

単位	6.1 の定義	8.0 の定義
天文単位（AU）	149,498,845,000 m	149,597,900,000 m
英温度単位 (平均)	1055.79 J	1055.87 J
電子ボルト（eV）	1.602e−19 J	1.60217642e−19 J
フィートキャンドル	10.764 lx	10.7639 lx
馬力 v.s. 馬力（電気）	745.7 W	746 W。新しい変換は厳密です。
英ガロン	4.54596 l	4.54609 l
光年	9.4605 Pm	9.46073 Pm
重量ポンド	4.448 N	4.448222 N
ロッド	502.92 cm	5.029210 m
スラグ	32.174 lb	14.59390 kg
統一された原子質量（u）	1.66057e−27 kg	1.66053873e−27 kg

パネルアップデートを延期プロパティ

LabVIEW 6.1 以前では、「パネルアップデートを延期」プロパティが FALSE になるまで、変更待ちのフロントパネルオブジェクトは再描画されませんでした。LabVIEW 8.0 では、このプロパティを TRUE に設定すると、変更待ちになっているすべてのフロントパネルオブジェクトを再描画して、フロントパネル更新の新しい要求をすべて保留にします。場合によっては、この変更によってフロントパネルの変更された要素が余分な回数再描画されることがあります。

数値制御器のデータ範囲

LabVIEW 6.1 以前では、いくつかの数値制御器のデフォルト設定が最小値 `0.00`、最大値 `0.00`、増分値 `0.00`、範囲外の動作が**無視**に設定されています。LabVIEW 8.0 では、これらの数値制御器はそのデータタイプのデフォルトのデータ範囲値を使用します。

強制ドットとタイプ定義

LabVIEW 6.1 以前では、タイプ定義に関する情報がワイヤに含まれていたため、ブロックダイアグラム上に強制ドットが多く発生することがありました。タイプ定義を、タイプ定義端子でない VI または関数の端子に配線すると、強制ドットが表示されます。タイプ定義である出力端子をタイプ定義でない表示器に配線したときにも、強制ドットは表示されます。そのような強制ドットは、VI でのタイプ定義の使用が一貫していないことを示します。この場合、強制ドットは実行時の性能に影響しません。

「文字列に平坦化（Flatten To String）」関数を使用してタイプ定義を平坦化する方法については、『LabVIEW ヘルプ』を参照してください。

ファイルダイアログボックスのボタンラベル

LabVIEW 6.1 以前では、「ファイルダイアログ（File Dialog）」関数によって表示されるファイルダイアログボックスには、ユーザが新規ファイル名を入力すると、**保存**というボタンラベルが表示されます。選択モードの入力パラメータによっては、ボタンラベルは**開く**になっています。LabVIEW 8.0 では、「ファイルダイアログ（File Dialog）」Express VI が表示するファイルダイアログボックスのボタンのラベルが、変更されない限りすべてのケースにおいて **OK** になりました。ボタンのラベルを変更するには、「ファイルダイアログ」Express VI の**ボタンラベル**入力を使用します。既存の VI で「ファイルダイアログ」Express VI を使用する場合は、VI の動作を再確認して、デフォルトラベルの **OK** が VI の機能面で適切かどうかを判断する必要があります。

オンラインヘルプを制御関数

「オンラインヘルプを制御（Control Online Help）」関数の**ヘルプファイルへのパス**入力は必須入力となりました。その入力には、コンパイル済みのヘルプファイル名（`.chm` または `.hlp`）またはコンパイル済みのヘルプファイルへの完全なパスを配線できます。コンパイル済みのヘルプファイルのみを配線した場合、LabVIEW は `labview¥help` ディレクトリ内でそのファイルを検索します。

VI を実行メソッド

LabVIEW 7.1 では、VI を実行メソッドの**リファレンスを自動破棄**入力を TRUE に設定すると、VI の実行を停止した後にリファレンスが自動的に破棄されます。LabVIEW 8.0 では、メソッドがエラーを返すと、ただちにリファレンスを破棄します。ブロックダイアグラムの一部がそのリファレンスに依存している場合、この動作によって VI が壊れることがあります。

ロード時にフロントパネルを表示する

LabVIEW 8.0 では、VI がロードされたときにフロントパネルを表示するよう VI が構成されていても、ユーザが VI サーバを使用して VI をロードした場合は、フロントパネルは表示されません。その場合は、フロントパネル : 開くメソッドを使用してプログラムでフロントパネルを表示する必要があります。

VI リファレンスを開く関数

LabVIEW 6.1 以前では、「VI リファレンスを開く（Open VI Reference）」関数の**オプション**パラメータに値を配線しないと、テンプレートがメモリにない場合はテンプレートから VI が作成されます。テンプレートがメモリにある場合は、テンプレートへのリファレンスが開きます。LabVIEW 7.0 以降では、、「VI リファレンスを開く」関数を使用して

メモリに既存のテンプレートへのリファレンスを作成すると、**オプション**パラメータに `0x02` を指定しないと関数によってエラーが返されます。

指数表記

LabVIEW 6.0 以前では、^ 演算子はフォーミュラノードで指数を表します。LabVIEW 6.1 以降のバージョンでは、指数を表す新しい演算子は `**` で、`x**y` のように使用します。^ 演算子は、ビット単位の排他的論理和 (XOR) 演算を表します。

IVI Configuration Store ファイル

IVI Configuration Store ファイルでは、すべて大文字小文字を区別するようになりました。論理名、ドライバセッション名、または仮想名をアプリケーションで使用する場合、使用する名前は IVI Configuration Store ファイルで定義されている名前と完全に一致する必要があり、大文字小文字も同じでなくてはなりません。

技術サポートフォーム

LabVIEW 7.*x* 以降では、LabVIEW のインストールプログラムは `techsup.llb` をインストールしません。インストール、構成、およびアプリケーションに関する問題や質問は、ナショナルインスツルメンツのウェブサイト（`ni.com/support/ja`）を参照してください。

LabVIEW 5.*x* からのアップグレード

LabVIEW 5.*x* から LabVIEW 8.0 にアップグレードする際、以下のような互換性に関する問題が発生する可能性があります。アップグレードの際に起こり得る他の問題点は、本書の「LabVIEW 6.*x* からのアップグレード」、「LabVIEW 7.*x* からのアップグレード」の各セクションを参照してください。

LabVIEW の各バージョンの新機能および変更点については、バージョン 5.*x* から 8.0 までの『LabVIEW アップグレードノート』および『LabVIEW 5.1 追補マニュアル』（`ni.com/jp/manuals`）を参照してください。

データログファイルの変換

LabVIEW 8.0 はデータログファイルのタイプ定義をチェックして、変換が必要かどうかを確認します。LabVIEW 6.0 以前で保存されたデータログファイル、または LabVIEW 7.1 以前で保存され波形データタイプが含まれているデータログファイルは、LabVIEW 8.0 で読み取り、追加するときに変換されます。それ以外の場合は、データログファイルを読み取ったりデータログファイルに追加しても、データログファイルは変換されません。

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 8.0 ではそのファイルを LabVIEW 8.0 形式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」を選択すると、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換されます。ファイルを変換しないを選択すると、エラーが返されファイルは開きません。

メモ 今後も古いデータを LabVIEW 6.1 以前のバージョンで使用することがある場合は、変換の前にデータログファイルのバックアップコピーを取っておいてください。変換したデータログファイルは、LabVIEW 6.1 以前に戻したり、LabVIEW 6.1 で読み取ったりすることはできません。

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 環境設定ファイルに次の行を追加します。

```
silentDatalogConvert=True
```

(Mac OS) 次の行を追加します。

```
silentDatalogConvert:True
```

(Linux) 次の行を追加します。

```
labview.silentDatalogConvert:True
```

データログファイルを開いたときに自動的にデータログファイルを変換したくない場合は、環境設定を `FALSE` に設定します。

VI サーバへの接続

LabVIEW 5.*x* のアプリケーションでは LabVIEW 8.0 VI サーバプロトコルの特徴の一部を認識しないため、LabVIEW 8.0 クライアントから LabVIEW 5.*x* アプリケーションの VI サーバに接続することはできません。ただし、LabVIEW 5.*x* クライアントから LabVIEW 8.0 アプリケーションの VI サーバへ接続できます。

UDP 関数

LabVIEW 6.*x* 以降では、UDP VI は**関数**パレットにありませんが、互換性 VI として `labview¥vi.lib¥_oldvers¥_oldvers.llb` に存在しました。ネットワーク通信については、標準の UDP 関数を代用してください。

LabVIEW 4.*x* からのアップグレード

LabVIEW 4.*x* から LabVIEW 8.0 にアップグレードする際、以下のような互換性に関する問題が発生する可能性があります。アップグレードの際に起こり得る他の問題点については、本書の「LabVIEW 5.*x* からのアップグレード」「LabVIEW 6.*x* からのアップグレード」および「LabVIEW 7.*x* からのアップグレード」の各セクションを参照してください。

LabVIEW の各バージョンの新機能および変更点については、バージョン 4.*x* から 8.0 までの『LabVIEW アップグレードノート』および

『LabVIEW 5.1 追補マニュアル』（ni.com/jp/manuals）を参照してください。

ブールデータの LabVIEW 4.*x* から / への変換

LabVIEW 4.*x* と LabVIEW 5.*x* データでのブールデータの格納形式は、LabVIEW 4.*x* と LabVIEW 5.*x* の間で変更されました。LabVIEW 4.*x* では、配列でないブールデータは 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 8.0 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ログ関数が増え、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した。ブール値のデータ形式について、LabVIEW 8.0 では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 8.0 では既存の CIN との間に互換性がもたらされました。

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 8.0 で同じデータを書き込む場合とは異なります。LabVIEW 8.0 には、LabVIEW 4.*x* で書き込まれたバイナリデータを読み取り、LabVIEW 4.*x* で読み取ることのでき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。「タイプキャスト（Type Cast）」、「文字列に平坦化（Flatten To String）」、および「文字列から非平坦化（Unflatten From String）」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。関数を右クリックしてこのメニュー項目を選択すると、この関数は、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処理できます。LabVIEW 4.*x* 用にフォーマットされたデータを作成するには、「バイナリファイルに書き込み」、「文字列に平坦化」、または「タイプキャスト」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取るには、「バイナリファイルから読み取り」、「文字列から非平坦化」、または「タイプキャスト」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 8.0 ではその関数上に赤い `4.x` グリフを配置して 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。データ変換を回避するには、**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チェックマークを外します。

ブール値を持つデータファイルが複数ある場合、これらのファイルを開いて LabVIEW 8.0 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* 以前のバージョンで保存した VI を LabVIEW 8.0 でロードすると、「ファイルに書き込み」、「ファイルから読み取り」、「タイプキャスト」、「文字列に平坦化」、および「文字列から非平坦化」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動作します。VI に LabVIEW 8.0 のブールデータ形式を使用する場合は、関数を右クリックし、ショートカットメニューから **4.x データ**

の変換を選択してチェックマークを外します。LabVIEW 4.*x* 以前のバージョンに書き込みを行ったブールデータを含むファイルを VI で操作する必要がなく、またブールデータを含むデータを LabVIEW 4.*x* 以前のバージョンで実行している VI とデータを送受信することがない場合には、LabVIEW 8.0 のブールデータ形式を使用します。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。

VI 制御 VI

LabVIEW 5.*x* 以降では、VI 制御 VI は**関数**パレットにありませんが、互換性 VI として `labview¥vi.liblity¥victl.llb` に存在しました。VI 制御 VI の代わりに、VI サーバ関数の「VI リファレンスを開く（Open VI Reference）」、「リファレンス呼び出しノード（Call By Reference Node）」、「プロパティノード（Property Node）」、および「インボークノード（Invoke Node）」を使用してください。

VI 制御 VI が返すエラーコードは、LabVIEW 8.0 では異なるものがいくつかあります。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 を返します。これに対し、LabVIEW 8.0 の VI 制御 VI はエラーコード 1004 および 1003 を返します。LabVIEW 4.*x* で作成された VI がエラーコード 7 および 1000 をチェックする場合には、その VI が LabVIEW 8.0 で動作するように修正する必要があります。

DDE VI （Windows）

LabVIEW 5.*x* 以降では、DDE VI は**関数**パレットにありませんが、互換性 VI として `labview¥vi.lib¥platform¥dde.llb` に存在しました。

LabVIEW 3.*x* 以前のバージョンからのアップグレード

VI 変換キットを使用した LabVIEW 3.*x* 以前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。アップグレードの際に起こる可能性のある他の問題点については、本書の「LabVIEW 4.*x* からのアップグレード」、「LabVIEW 5.*x* からのアップグレード」、「LabVIEW 6.*x* からのアップグレード」、および「LabVIEW 7.*x* からのアップグレード」の各セクションを参照してください。

LabVIEW の各バージョンの新機能および変更点については、バージョン 3.*x* から 8.0 までの『LabVIEW アップグレードノート』および『LabVIEW 5.1 追補マニュアル』（`ni.com/jp/manuals`）を参照してください。

LabVIEW 8.0 の機能および変更点

プログラミングの概念、操作手順、および LabVIEW 8.0 の機能のリファレンス情報については、『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→ LabVIEW ヘルプを検索**を選択します。

LabVIEW 8.0 の機能および変更点の詳細については、`labview` ディレクトリにある `readme.html` を参照してください。

LabVIEW ライセンスをアクティブ化する

(Windows) LabVIEW はライセンスのアクティブ化に依存します。LabVIEW を実行する前に有効な LabVIEW ライセンスをアクティブ化する必要があります。LabVIEW をアクティブ化するには、インストールパッケージの一部として受領したシリアル番号を使用します。以下の方法で LabVIEW ライセンスをアクティブ化できます。

- インストール時にシリアル番号を入力し、インストールの最後に**アクティブ化ウィザード**を実行するように選択します。
- 評価モードで LabVIEW を起動したら、**アクティブ化スタートアップ**ダイアログボックスで**アクティブ化**を選択します。
- 評価モードで LabVIEW を実行中に、**ヘルプ→ LabVIEW をアクティブ化**を選択します。ライセンスのアクティブ化は、LabVIEW を再起動するまで有効になりません。

LabVIEW でのライセンス詳細については、『LabVIEW 8.0 リリースノート』を参照してください。

LabVIEW を起動する

LabVIEW を起動すると、**スタートアップ**ウィンドウが表示されます。このウィンドウでは、新しい VI やプロジェクトの作成、最近開いた LabVIEW ファイルの選択、サンプルの検索、『LabVIEW ヘルプ』へのアクセスを実行できます。

スタートアップウィンドウは、既存のファイルを開いたり新規のファイルを作成すると閉じます。**表示→スタートアップウィンドウ**を選択すると、このウィンドウを表示することができます。**スタートアップ**ウィンドウは、開いているすべてのフロントパネルとブロックダイアグラムを閉じたときにも表示されます。

LabVIEW をログインおよびログアウトする

ネットワークでセキュリティドメインが構成されている場合は、LabVIEW 環境にログインできます。ログイン情報は、強制アクセス制御リストを持つ、リモート VI サーバと通信する際に、LabVIEW を認識します。NI セキュリティ : ログイン、NI セキュリティ : ログアウトメソッド、および NI セキュリティユーザ変更イベントをそれぞれ使用することによって、LabVIEW をプログラムでログインおよびログアウトし、ブロックダイアグラムからユーザのアカウントの変更を監視することができます。

ドメインアカウントマネージャ

ツール→セキュリティ→ドメインアカウントマネージャを選択して、ドメインを作成および破棄、ドメインポリシーを実施して、ローカルまたはリモードでドメイン内のユーザとグループのアカウントを管理します。ドメインアカウントマネージャは、離れた場所からドメインをホストするリモードマシンでリモートドメインを管理する場合に特に便利です。認証プロトコルは、標準のデータ暗号化およびデータ統合により保護されています。

新規ダイアログボックスの機能強化

スタートアップウィンドウで**新規**リンクをクリックして、**新規**ダイアログボックスを表示します。**新規**ダイアログボックスを使用して、新規 VI、プロジェクト、プロジェクトライブラリ、カスタム制御器、シェア変数などを作成できます。**プロジェクトへ追加**チェックボックスをオンにして、新規項目を開いているプロジェクトに追加します。

In LabVIEW 7.*x* 以前では、**小さいダイアログ**または**大きいダイアログ**ボタンを選択して、**新規**ダイアログボックスのサイズを縮小または拡張することができました。LabVIEW 8.0 では、**小さいダイアログ**または**大きいダイアログ**ボタンがなくなりました。ただし、**新規**ダイアログボックスのサイズを変更すると、次にアクセスするときに LabVIEW によってダイアログボックスのサイズが記憶されます。

LabVIEW 7.*x* 以前のバージョンでは、**新規**ダイアログボックスは選択した VI のフロントパネルとブロックダイアグラムのプレビューを表示しました。LabVIEW 8.0 では、ブロックダイアグラムのプレビューは**新規**ダイアログボックスの**説明**セクションに表示されます。

新規作成リストには、LabVIEW 7.*x* 以前からの 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の項目が含まれなくなりました。VI のフロントパネルまたはブロックダイアグラム、またはグローバル変数で、**ファイル→別名で保存**を選択してファイルダイアログボックスを表示し、**テンプレート VI** を**ファイルの種類**プルダウンメニューから選択して、項目をテンプレートとして保存します。カスタム制御器のフロントパネルで、**ファイル→別名で保存**を選択してファイルダイアログボックスを表示し、**テンプレート制御器**を**ファイルの種類**プルダウンメニューから選択して、制御器をテンプレートとして保存します。

RT、FPGA、ならびに PDA ターゲット

RT、FPGA、または PDA ターゲットで作業するには、LabVIEW プロジェクトを作成する必要があります。LabVIEW Real-Time、FPGA、および PDA モジュールとプロジェクトの使用の詳細については、特定のモジュールのドキュメントを参照してください。

実行ターゲットを切り替える

LabVIEW 7.*x* 以前では、**LabVIEW** ダイアログボックスの**実行ターゲット**プルダウンメニューからターゲットを選択して、VI を開発するターゲットを選択する必要がありました。VI の作業中に実行ターゲットを変更する必要がある場合、**操作→実行ターゲットの切り替え**を選択して、切り替えるターゲットを選択できます。LabVIEW 8.0 では、プロジェクト環境を使用して実行ターゲット上で作業できます。**ターゲットとデバイスを追加**ダイアログボックスを表示するには、**プロジェクトエクスプローラ**ウィンドウでプロジェクトルートを右クリックして、ショートカットメニューから**新規→ターゲットとデバイス**を選択します。プロジェクトのターゲットが他のターゲットをサポートする場合、ターゲットを右クリックして、ショートカットメニューから**新規→ターゲットとデバイス**を選択し、既存のターゲットの下にターゲットを追加することもできます。ターゲットの下で VI を追加および開発できます。そして、これらのターゲットの下で VI を切り替えることによって、実行ターゲットを切り替えることができます。LabVIEW Real-Time、FPGA、および PDA モジュールの実行ターゲットの切り替えの詳細については、モジュールのドキュメントを参照してください。

LabVIEW プロジェクト

プロジェクトを使用して、LabVIEW ファイルと非 LabVIEW ファイルを一緒にグループ化してビルド仕様を作成し、ファイルをターゲットへ配置またはダウンロードします。ターゲットは、VI を実行するデバイスまたはマシンです。プロジェクトを保存すると、プロジェクトの構成情報、ビルド情報、配置情報、ファイルへのリファレンスなどを含む、プロジェクトファイル（`.lvproj`）が作成されます。

スタンドアロンアプリケーションおよび共有ライブラリを作成するために、プロジェクトを使用する必要があります。RT、FPGA、または PDA ターゲットで作業するには、プロジェクトを使用する必要があります。LabVIEW Real-Time、FPGA、および PDA モジュールとプロジェクトの使用の詳細については、特定のモジュールのドキュメントを参照してください。

プロジェクトスタイルの LabVIEW プラグアンドプレイ計測器ドライバは、LabVIEW 8.0 のプロジェクトおよびプロジェクトライブラリの機能を使用します。以前の LabVIEW プラグアンドプレイドライバでも同様にプロジェクトスタイルのドライバを使用できます。

プロジェクトエクスプローラウィンドウ

プロジェクトエクスプローラウィンドウを使用して、プロジェクトを作成および編集します。**ファイル→新規プロジェクト**を選択して**プロジェクトエクスプローラ**ウィンドウを表示します。**プロジェクト→新規プロジェクト**または**ファイル→新規**を選択して、**新規**ダイアログボックスで**空のプロ**

ジェクトを選択して、**プロジェクトエクスプローラ**ウィンドウを表示することもできます。

プロジェクトエクスプローラウィンドウには、デフォルトで以下の項目が含まれています。

- **プロジェクトルート**—**プロジェクトエクスプローラ**ウィンドウに他のすべての項目を含みます。プロジェクトのルート上のラベルには、プロジェクトのファイル名が含まれています。
  - **マイコンピュータ**—プロジェクト内でローカルコンピュータをターゲットとして表します。
  - **依存性**—ターゲットの下にある VI が必要とする項目が含まれます。
  - **ビルド仕様**—ソース配布のビルド構成、および LabVIEW ツールキットとモジュールで使用できる他のビルドの種類が含まれます。LabVIEW プロフェッショナル開発システムまたはアプリケーションビルダがインストールされている場合、**ビルド仕様**を使用して、スタンドアロンアプリケーション（EXE）、共有ライブラリ（DLL）、および Zip ファイルを構成します。**(Windows)** **ビルド仕様**を使用してインストーラを構成することもできます。

プロジェクトに他のターゲットを追加すると、**プロジェクトエクスプローラ**ウィンドウでターゲットを表す追加項目が作成されます。各ターゲットには、**依存性**および**ビルド仕様**も含まれます。各ターゲットの下にファイルを追加できます。

プロジェクトエクスプローラウィンドウから、他の開かれている VI のブロックダイアグラムに VI をドラッグして、VI をサブ VI として使用できます。

既存のファイルでプロジェクトを使用する

既存ファイルのグループをプロジェクトに追加できます。以下の既存ファイルをプロジェクトに追加するオプションから 1 つを選択してください。

- 既存ファイルがスタンドアプリケーションの一部である場合は、**ツール→ビルドスクリプトを変換**を選択して、アプリケーションをプロジェクに変換します。
- 既存ファイルがスタンドアプリケーションの一部でない場合は、新規プロジェクトを作成し、**プロジェクトエクスプローラ**ウィンドウにファイルを追加します。

プロジェクトエクスプローラウィンドウツールバー

プロジェクトで操作を実行するには、標準、プロジェクト、ビルド、およびソース管理ツールバーのボタンを使用します。ツールバーは、**プロジェクトエクスプローラ**ウィンドウの上部にあります。**プロジェクトエクスプ**

ローラウィンドウのサイズを変更して、すべてのツールバーを表示することもできます。

表示→ツールバーを選択する、および表示または非表示にするツールバーを選択することによって、ツールバーを表示または非表示にすることができます。ツールバー上の空白領域で右クリックして、表示または非表示にするツールバーを選択します。

プロジェクトに項目を追加する

VI や LLB などの LabVIEW ファイル、およびテキストファイルやスプレッドシートなどの非 LabVIEW ファイルをプロジェクトのターゲットへ追加するには、**プロジェクトエクスプローラ**ウィンドウを使用します。プロジェクトで項目の構造を作成することができます。

以下の手順で、プロジェクトでターゲットの下に項目を追加できます。

- ターゲットまたはターゲットの下にあるフォルダを右クリックし、ショートカットメニューから**ファイルを追加**を選択して、ファイルダイアログボックスから追加するファイルを選択します。また、ターゲットを選択して**プロジェクト→プロジェクトへ追加→ファイルを追加**を選択し、ファイルダイアログボックスから追加するファイルを選択することもできます。
- 新規のブランク VI を追加するには、ターゲットを右クリックしてショートカットメニューから**新規→ VI** を選択します。また、**ファイル→新規 VI** または**プロジェクト→プロジェクトへ追加→新規 VI** を選択して、新規のブランク VI を追加することもできます。
- フロントパネルの右上隅またはブロックダイアグラムウィンドウからにある VI アイコンを選択し、このアイコンをターゲットへドラッグします。
- **(Windows および Mac OS)** ディスク上の項目またはディレクトリを選択して、ターゲットにドラッグします。

また、**新規**ダイアログボックスからプロジェクトへ新規 LabVIEW ファイルを追加することもできます。**新規**ダイアログボックスを表示するには、**ファイル→新規**または**プロジェクト→プロジェクトへ追加→新規**を選択します。**新規**ダイアログボックスでは、追加する項目を選択して**プロジェクトへ追加**チェックボックスをオンにします。複数のプロジェクトを開く場合は、**プロジェクト**リストから項目を追加するプロジェクトを選択します。

フォルダをプロジェクトに追加する

プロジェクトエクスプローラウィンドウを使用して、ディスク上のディレクトリをフォルダとしてプロジェクトで追加します。ディスク上のディレクトリからファイルを追加するには、ターゲットまたはターゲットの下にあるフォルダを右クリックし、ショートカットメニューから**フォルダを追加**を選択します。ディスク上のディレクトリを選択すると、ファイルとサブフォルダの内容を含む、ディレクトリ全体の内容を表すプロジェクト項

目が作成されます。.LabVIEW は、ディスク上のディレクトリと同じ名前の新規仮想フォルダをプロジェクトに作成します。

また、ターゲットを選択して**プロジェクト→プロジェクトへ追加→フォルダを追加**を選択し、ディレクトリからファイルを追加することもできます。

メモ プロジェクトにディスク上のディレクトリを追加した後に、ディレクトリに変更を加えた場合、LabVIEW は自動的にプロジェクトのフォルダを更新しません。ディスク上のサブディレクトリのストラクチャを変更しても、プロジェクトのサブフォルダは自動的に更新されません。プロジェクトのフォルダがディスク上のディレクトリに一致するようにしたい場合は、フォルダを手動で更新する必要があります。

また、新規フォルダを作成してプロジェクトで項目を整理することもできます。ターゲットを右クリックして、ショートカットメニューから**新規→フォルダ**を選択して、ターゲットの下に新規フォルダを追加します。また、既存フォルダを右クリックして、ショートカットメニューから**新規→フォルダ**を選択することによって、新規サブフォルダを作成することもできます。

プロジェクトルートを右クリックし、ショートカットメニューから**新規→フォルダ**を選択して、新規フォルダを追加してプロジェクトのターゲットを整理できます。

プロジェクトに LLB を追加する

フォルダまたはファイルとして、プロジェクトに LLB を追加できます。フォルダとして LLB を追加する場合、LLB のファイル名を使用してフォルダに名前をつけ、新規フォルダ内で項目として LLB にある VI を追加します。ターゲットまたはターゲットの下のフォルダを右クリックし、ショートカットメニューから**フォルダを追加**を選択して、追加する LLB を検索するファイルダイアログボックスを使用します。**プロジェクトエクスプローラ**ウィンドウにあるフォルダから項目を追加または削除しても、ディスク上の `.llb` ファイルには影響しません。ディスク上の LLB を編集するには、**ツール→ LLB マネージャ**を選択します。

また、ファイルとしてプロジェクトに LLB を追加することもできます。ファイルとして LLB を追加する場合、LLB にある VI は**プロジェクトエクスプローラ**ウィンドウには表示されません。ターゲットまたはターゲットの下のフォルダを右クリックし、ショートカットメニューから**ファイルを追加**を選択して、追加する LLB を検索するファイルダイアログボックスを使用します。ピリオドの右側に **VI** と開いたフォルダのアイコンを選択し、**選択**ボタンをクリックします。

プロジェクトから項目を削除する

プロジェクトエクスプローラウィンドウを使用して、プロジェクトから項目を削除します。次の方法で項目を削除できます。

- 削除する項目を右クリックし、ショートカットメニューから**削除**を選択します。
- 削除する項目を選択して、<Delete> キーを押します。
- 削除する項目を選択して、**標準**ツールバーの**削除**ボタンをクリックします。

メモ プロジェクトから項目を取り除いても、ディスク上の対応する項目は削除されません。

プロジェクトで依存データを表示します。

依存性を使用して、ターゲットの下の VI に必要な項目が表示されます。各ターゲットには**依存データ**我含まれます。

依存性には直接項目を追加できません。**依存性**を右クリックして、ショートカットメニューから**更新**を選択すると、ターゲットの下の VI の依存データが追加されます。たとえば、サブ VI が含まれる VI をターゲットに追加し、**更新**を選択すると、サブ VI が**依存性**に追加されます。ただし、依存項目をターゲットの下に追加しても、**依存性**の下に項目は表示されません。たとえば、サブ VI が含まれる VI をターゲットの下に追加し、**更新**を選択しても、サブ VI は**依存性**には追加されません。

依存データには、VI、DLL、および VI が静的に呼び出すプロジェクトライブラリが含まれます。

メモ VI が動的に呼び出す項目は**依存性**には表示されません。これらの項目をターゲットの下に追加して、プロジェクト内で管理する必要があります。

LabVIEW はサブ VI を回帰的に記録します。LabVIEW は DLL を回帰的に記録しません。たとえば、`a.vi` が `b.dll` を静的に呼び出し、`b.dll` が `c.dll` を静的に呼び出すと、LabVIEW は `b.dll` のみを依存項目と見なします。`c.dll` をプロジェクトで管理するには、`c.dll` をターゲットの下に追加する必要があります。

依存項目がプロジェクトライブラリの一部である場合、LabVIEW はプロジェクトライブラリ全体を**依存性**に追加します。

依存性の下には新規項目を作成できません。**プロジェクトエクスプローラ**ウィンドウの他の場所から**依存性**に項目をドラッグすることはできません。

依存性から項目を削除するには、その項目を右クリックし、ショートカットメニューから**削除**を選択します。このオプションを使用して項目を削除

する場合、**依存性**を右クリックしてショートカットメニューから**更新**を選択すると、項目が再度表示されます。

プロジェクトを保存する際、LabVIEW はその依存データをプロジェクトの一部として保存しません。プロジェクトを開く場合、**依存性**を右クリックして、ショートカットメニューから**更新**を選択して依存データを表示する必要があります。

ターゲットをプロジェクトに追加する (Windows)

ターゲットとデバイスを追加ダイアログボックスを使用して、プロジェクトにターゲットまたはデバイスを追加します。

メモ このダイアログボックスを表示するには、ターゲットをサポートするモジュールまたはドライバがインストールされている必要があります。

プロジェクトルートを右クリックして、ショートカットメニューから**新規→ターゲットとデバイス**を選択して、**ターゲットとデバイスを追加**ダイアログボックスを表示します。プロジェクトのターゲットが他のターゲットをサポートする場合、ターゲットを右クリックして、ショートカットメニューから**新規→ターゲットとデバイス**を選択し、既存のターゲットの下にターゲットを追加することもできます。たとえば、コンピュータに NI PCI デバイスがインストールされている場合、**マイコンピュータ**ターゲットの下にデバイスを追加できます。

ファイルをターゲットに配置する

プロジェクトエクスプローラウィンドウで、ターゲットの下で VI を開いて実行し、ターゲットにファイルを配置します。VI を実行すると、ソースファイル、ビルド出力、およびハードウェア構成がターゲットに配置されます。

複数のターゲットでパレットを使用する

LabVIEW は各ターゲット用に別々のパレットセットをロードします。プロジェクトの複数のターゲットで VI を開く場合は、複数のパレット設定がメモリ内に存在します。ターゲット間で切り替える場合、アクティブなターゲットで各ピン付けされたパレットを対応するパレット部分にマップしようとします。ピン付けされたパレットに対応するパレットがない場合は、アクティブなターゲットのトップレベルのパレットが表示されます。

アプリケーションインスタンスを使用する

LabVIEW は、LabVIEW プロジェクトで各ターゲットのアプリケーションインスタンスまたは LabVIEW のインスタンスを作成します。**プロジェクトエクスプローラ**ウィンドウから VI を開くと、VI はターゲットのアプリケーションインスタンスで開かれます。また、LabVIEW は、プロジェ

クトの一部ではない開かれている VI およびプロジェクトから開かれていない VI の、主なアプリケーションインスタンスを持っています。

フロントパネルおよびブロックダイアグラムの左下隅に表示されるアプリケーションインスタンス名を使用して、VI が属するアプリケーションインスタンスを認識します。複数のプロジェクトが開かれている場合は、アプリケーションインスタンス名には、**Project 1.lvproj/My Computer** のように、ターゲット名が後に続くプロジェクト名が含まれます。それ以外の場合は、アプリケーションインスタンスの名前にはターゲット名のみが含まれます。

VI が開いているときに**プロジェクトエクスプローラ**ウィンドウから VI を削除すると、VI は同じアプリケーションインスタンス内に残ります。VI と閉じて、再度開くと、LabVIEW は現在のアプリケーションインスタンスにある VI を開きます。

複数のアプリケーションインスタンスで VI を編集する

複数のアプリケーションインスタンスのディスク上で同じ VI を同時に開くことができます。たとえば、同じ VI を 2 つの異なるプロジェクト、またはプロジェクト内の 2 つの異なるターゲットから開くことができます。

1 つのアプリケーションインスタンスのみで開かれている VI を編集して、VI を他のアプリケーションインスタンスで開くと、開いた VI には編集された VI のインスタンスからの最新の変更が含まれています。ただし、複数のアプリケーションインスタンスで既に開かれている VI を編集すると、LabVIEW は他のアプリケーションインスタンスでの VI への変更を自動的に適用しません。他のアプリケーションインスタンスの VI の編集、実行、または保存は、VI がすべてのアプリケーションインスタンスで同じになるまでできません。

VI をすべてのアプリケーションインスタンスで同じにするためには、以下の方法のいずれかを実行してください。

- VI のツールバー上にある**他のアプリケーションインスタンスと同期**ボタンをクリックして、すべてのアプリケーションインスタンスの VI に変更を適用します。
- 変更を含むアプリケーションインスタンスの VI を保存します。VI を保存すると、LabVIEW は自動的に他のすべてのアプリケーションインスタンスで VI に変更を適用します。
- 変更を含むアプリケーションインスタンスの VI への編集を取り消します。

メモ VI を保存または同期した後に VI への変更を取り消すことはできません。

実行中、または他のアプリケーションインスタンスで実行するために予約されている VI を編集できません。VI がプロジェクトライブラリの一部である場合は、VI を実行すると、すべてのアプリケーションインスタンス

でプロジェクトライブラリも一時的にロックされます。VI の実行中にプロジェクトライブラリを編集できません。VI が停止した後、プロジェクトライブラリのロックが解除されます。

プロジェクトで VI のウェブサーバを使用する

プロジェクトの一部である VI またはフロントパネルを表示する場合は、ウェブサーバを使用します。**操作→リモートパネルに接続**を選択する場合、プロジェクト名、プロジェクトライブラリ、そして該当する場合はターゲットを **VI 名**に含めます。

たとえば、ターゲットのマイコンピュータの下にある `MyProject.lvproj` というプロジェクトに `MyVI.vi` が存在する場合、**VI 名**は `MyProject.lvproj¥My Computer¥MyVI.vi` と入力します。`MyLibrary` というプロジェクトライブラリが VI を所有する場合、`MyProject.lvproj¥My Computer¥MyLibrary:MyVI.vi` として **VI 名**にプロジェクトライブラリも含めます。VI がプロジェクトになく、プロジェクトライブラリが VI を所有しない場合は、追加情報なしで **VI 名**を入力します。

LabVIEW プロジェクトライブラリ

LabVIEW プロジェクトライブラリは、他のプロジェクトライブラリを含む、VI、タイプ定義、シェア変数、パレットメニューファイル、およびその他のファイルの集合です。新規プロジェクトライブラリを作成および保存すると、プロジェクトライブラリファイル（`.lvlib`）が作成されます。このプロジェクトライブラリファイルには、プロジェクトライブラリのプロパティ、およびプロジェクトライブラリが所有するファイルのリファレンスが含まれます。

プロジェクトライブラリは、単一の項目の階層へのファイルの整理、起こり得る VI 名の重複の回避、特定のファイルに対するパブリックアクセスの制限、ファイルの集合に対する編集許可の制限、および VI のグループに対するデフォルトのパレットメニューの設定を行う場合に便利です。

プロジェクトエクスプローラウィンドウまたはスタンドアロンのプロジェクトライブラリウィンドウで、プロジェクトライブラリの構成を表示できます。**プロジェクトエクスプローラ**ウィンドウが開かれていない場合は、プロジェクトライブラリファイルを参照して、プロジェクトライブラリウィンドウで開きます。

LabVIEW プロジェクトでプロジェクトライブラリを作成できます。**マイコンピュータ**を右クリックして、ショートカットメニューから**新規→ライブラリ**を選択します。**マイ コンピュータ**の下に表示されるプロジェクトライブラリファイルが作成されます。

また、プロジェクトでフォルダからプロジェクトライブラリを作成することもできます。**プロジェクトエクスプローラ**ウィンドウで、フォルダを右

クリックしてショートカットメニューから**ライブラリに変換**を選択します。

プロジェクトライブラリを使用して、項目の仮想階層や論理階層を整理します。プロジェクトライブラリファイルは、VI を含む物理ファイルである LLB とは異なり、所有する実際のファイルを含みません。プロジェクトライブラリが所有するファイルは、ファイルを保存したディレクトリに、ディスク上でそれぞれ表示され続けます。プロジェクトライブラリは、ディスク上のファイルとは異なる構成になる可能性があります。

プロジェクトライブラリを使用して、VI 名およびその他の LabVIEW ファイル名を限定します。ファイル名で VI が識別されます。すでにメモリ内にある VI と同じ名前で VI をロードして、すでにメモリ内にある VI を使用すると、クロスリンクの問題が発生します。VI がプロジェクトライブラリの一部である場合、プロジェクトライブラリ名で VI 名を限定してクロスリンクを回避します。限定されたファイル名には、ファイル名および所有するプロジェクトライブラリのファイル名が含まれます。

たとえば、`library1.lvlib` が所有する `init.vi` という名前のサブ VI を含む `caller.vi` という名前の VI を作成する場合、`library2.lvlib` が所有する `init.vi` という名前の異なるサブ VI を含むことで、クロスリンクの問題を回避できます。`caller.vi` を保存する場合に LabVIEW に記録する限定されたファイル名は、それぞれ、`library1.lvlib:init.vi` と `library2.lvlib:init.vi` です。

注意 プロジェクトライブラリの名前を変更するには、プロジェクトライブラリを右クリックしてショートカットメニューから**名前の変更**を選択する必要があります。LabVIEW 以外でプロジェクトライブラリ名を変更すると、プロジェクトライブラリが壊れる可能性があります。

プロジェクトライブラリを右クリックして、ショートカットメニューから**プロパティ**を選択して、特定のファイルのタイプにパブリックアクセスを制限し、プロジェクトライブラリの編集権限を制限、そしてプロジェクトライブラリに所有されるパレットメニューファイル（`.mnu`）を、プロジェクトライブラリが所有するすべての VI のデフォルトパレットメニューに設定します。

デフォルトパレットメニューを選択すると、プロジェクトライブラリが所有する VI を呼び出すサブ VI を右クリックして、ショートカットメニューからそのプロジェクトライブラリのデフォルトパレットを表示できます。

シェア変数を使用してライブデータを共有する

LabVIEW 7.*x* 以前では、ローカルおよびグローバル変数を使用して、ワイヤで接続できないアプリケーションに情報を渡しました。アプリケーションの目的によって、変数のタイプを選択しました。LabVIEW 8.0 ではシェア変数が可能になり、プロジェクトまたはネットワークを介して複数の VI でライブデータの読み取りや書き込みに使用できます。

シェア変数は、DataSocket などの既存の LabVIEW データ転送技術の機能を組み合わせたもので、プロジェクトエクスプローラウィンドウでシェア変数を管理できます。フロントパネルまたはブロックダイアグラムから、シェア変数データの読み取りおよび書き込みを行えます。

シェア変数は、VI 間でデータを送信するソフトウェア項目で構成されます。シェア変数を使用して、VI 間、またはワイヤに配線できないアプリケーションの位置の間で、データを共有します。シェア変数は値または I/O ポイントを表すことができます。シェア変数を使用する VI のブロックダイアグラムを編集することなく、その目的、**変数タイプ**、その他すべてのシェア変数のプロパティを変更できます。

メモ シェア変数をファイアウォールまたはルータを介して転送する場合は、ファイアウォールおよびネットワークアドレス変換（NAT）を構成する必要があります。

シェア変数を作成する

シェア変数を作成するには、プロジェクトが開かれている必要があります。プロジェクトにシェア変数を追加するには、**プロジェクトエクスプローラ**ウィンドウのターゲット、プロジェクトライブラリ、またはプロジェクトライブラリ内のフォルダを右クリックし、ショートカットメニューから**新規→変数**を選択します。

シェア変数はプロジェクトライブラリの一部である必要があります。プロジェクトライブラリの一部ではないターゲットまたはフォルダからシェア変数を作成すると、LabVIEW は新規プロジェクトライブラリを作成し、シェア変数を所有します。

シェア変数 Refnum および変数プロパティを使用して、プロジェクトでシェア変数をプログラムで構成します。変数リファレンスプロパティから表示器を作成して、シェア変数 Refnum を作成します。単一プロセスのシェア変数には、この Refnum で変更を加えることができません。

フロントパネルでシェア変数値の読み取りおよび書き込みを行う

フロントパネルオブジェクト内のライブデータの読み取りまたは書き込みを行うには、フロントパネルのデータバインディングを使用します。

メモ ネットワークで共有のシェア変数に対してのみフロントパネルオブジェクトを連結できます。

シェア変数に制御器の境界を作成するには、**プロジェクトエクスプローラ**ウィンドウからシェア変数を VI のフロントパネルへドラッグします。制御器を右クリックして、ショートカットメニューから**プロパティ**を選択 h して、**データバインディング**ページにあるオプションを使用して、制御器をシェア変数へ、またはネットワーク上にある

NI Publish-Subscribe Protocol（NI-PSP）データ項目へ連結することもできます。

メモ LabVIEW 8.0 では制御器を右クリックして、ショートカットメニューから**データ操作→ DataSocket 接続**を選択できません。`dstp`、`opc`、`ftp`、`http`、`file` プロトコルを使用してフロントパネルオブジェクトを接続する代わりに、制御器**プロパティ**ダイアログボックスの**データバインディングセレクション**プルダウンメニューから **DataSocket** を選択します。

制御器のデータバインディングを有効にする場合、制御器の値を変更すると、制御器をバインドしたシェア変数の値も変更されます。

ブロックダイアグラムでシェア変数値の読み取りおよび書き込みを行う

シェア変数ノードはブロックダイアグラムオブジェクトで、**プロジェクトエクスプローラ**ウィンドウで対応するシェア変数を指し示します。シェア変数ノードは、シェア変数の値の読み取りおよび書き込みを行い、シェア変数データのタイムスタンプを読み取ります。シェア変数ノードを作成するには、**プロジェクトエクスプローラ**ウィンドウからシェア変数を、同じプロジェクト内の VI のブロックダイアグラムへドラッグします。

また、**関数**パレットからシェア変数ノードを選択して、ブロックダイアグラムに配置することもできます。ブロックダイアグラムのシェア変数ノードをアクティブプロジェクトにあるシェア変数に連結するには、シェア変数ノードをダブルクリックして**変数を選択**を表示します。シェア変数ノードを右クリックして、ショートカットメニューから**変数を選択**を選択することもできます。

シェア変数を 1 つの書き込みに制限する

デフォルトでは、複数のアプリケーションがシェア変数に書き込みを行うことができます。ただし、ネットワークで共有のシェア変数を設定して、一度に 1 つのターゲットのみから値の変更を受け入れるようにすることができます。**シェア変数プロパティ**ダイアログボックスの**変数**ページの**1 つの書き込み**チェックボックスをオンにします。これによって、シェア変数の書き込み操作が他の書き込みによって影響されないことを確認できます。シェア変数エンジンは、書き込みを 1 台のコンピュータの単一の VI に制限します。シェア変数に接続された最初の書き込みを行う書き込み側は値を書き込むことができますが、後続の書き込み側は書き込みできません。最初のライターが接続解除されると、キュー内の次のライターがシェア変数に値を書き込むことができます。LabVIEW は、ライターにシェア変数への書き込みが許可されていないことを通知します。

シェア変数のバッファを有効にする

シェア変数、または psp や dstpURL を使用してデータをプログラムで共有する場合は、LabVIEW はデフォルトですべての読み取りに対して最新値のみを書き込みます。あるクライアントが他のクライアントが値を読み取る前に値を書き込んだ場合、新しい値が、古い未処理の値をクライアントが読み取る前に上書きします。読み取り側が、以下の値を受け取る前に値を受け取らないと、データは損失されます。この未処理データの損失は、サーバまたはクライアントで発生します。データの損失は、データを読み取っている状態でサーバに書き込まれた最新値のみを受け取る場合には、問題になりません。ただし、サーバに書き込まれたすべてのシェア変数値を受け取りたい場合、または損失の可能性がある転送が不十分な場合は、クライアント上でデータをバッファ処理する必要があります。**シェア変数プロパティ**ダイアログボックスの**変数**ページの**バッファを使用**チェックボックスをオンにして、シェア変数のバッファを有効にします。

シェア変数を管理する

ツール→シェア変数→変数マネージャを選択して、**変数マネージャ**ダイアログボックスを表示します。このダイアログボックスを使用して、プロジェクト環境外にあるシェア変数を変数、作成、監視します。また、このダイアログボックスを使用してシェア変数エンジンの使用期間を管理できます。

ソース配布を作成する

ブロックダイアグラムの削除およびパスワード設定など、以前は**オプション付き保存**ダイアログボックスで使用可能だった VI の配布オプションは、ソース配布の作成時に使用可能になりました。

プロジェクトエクスプローラウィンドウの**ビルド仕様**を使用して、ソース配布のビルド仕様を作成します。**ソース配布プロパティ**ダイアログボックスを表示するには、**ビルド仕様**を右クリックしてショートカットメニューから**新規→ソース配布**を選択します。

旧バージョンのアプリケーションビルダからの変更点

アプリケーションビルダは**プロジェクトエクスプローラ**ウィンドウに統合されています。LabVIEW ベースパッケージまたは開発システムをご使用の場合は、ナショナルインスツルメンツのウェブサイト、`ni.com/jp` で info code の `rdlv21` を入力して、アプリケーションビルダを別途購入していただけます。

スタンドアロンアプリケーション（EXE）、共有ライブラリ（DLL）、および Zip ファイルのビルド仕様をビルドおよび作成するには、**プロジェクトエクスプローラ**ウィンドウの**ビルド仕様**を使用します。**(Windows) ビルド仕様**を使用して、インストーラのビルド仕様をビルドおよび作成することもできます。

ビルド仕様は旧バージョンのアプリケーションビルダの .bld ファイルと同等ですが、個別のファイルではなく、LabVIEW プロジェクトの一部となっています。

メモ プロジェクトを作成して、アプリケーションビルダツールを使用する必要があります。開いている VI から**実行ファイルビルドウィザード**を使用すると、アプリケーションのプロジェクトを作成できます。操作方法を案内する**実行ファイルビルドウィザード**を起動するには、**ツール→実行ファイルをビルド**を選択します。

.bld ファイルを新規プロジェクトのビルド仕様に変換できます。**ツール→ビルドスクリプト**を変換を選択し、変換する .bld ファイルを検索および選択します。LabVIEW は、このファイルを使用して、ソースファイルおよびビルド仕様を含むプロジェクトを作成します。

LabVIEW 8.0 では、アプリケーションビルダの以下の機能強化および変更が導入されました。

- 各種のビルド仕様には、ビルド仕様設定の構成および編集に使用できる複数ページのダイアログボックスがあります。ダイアログボックスにアクセスするには、ビルド仕様を右クリックしてショートカットメニューから**プロパティ**を選択します。
- **(Windows)** インストーラのビルド仕様は他の種類のビルド仕様とは異なります。1 つのインストーラビルドに、複数のアプリケーションまたは共有ライブラリのビルドを含められます。インストーラのビルド仕様を右クリックして、ショートカットメニューから**プロパティ**を選択し、**ソースファイル**ページを参照して、インストーラに含めるアプリケーション、共有ライブラリ、ソース配布ビルドを選択します。
- アプリケーションまたは共有ライブラリを作成する前に、ビルド仕様でデバッグを有効にすると、アプリケーションおよび共有ライブラリをデバッグできます。ビルド仕様を右クリックして、ショートカットメニューから**プロパティ**を選択して**上級**ページを参照し、**デバッギングを許可**チェックボックスをオンにします。

メモ アプリケーションおよび共有ライブラリをデバッグする場合、「VI リファレンスを開く（Open VI Reference）」関数を作成する再入可能フロントパネルをデバッグすることはできません。また、LabVIEW で共有ライブラリを作成したエントリポイントとなる再入可能フロントパネルをデバッグすることもできません。

- **(Windows)** アプリケーションおよび共有ライブラリのビルド仕様でバージョン番号を指定できます。アプリケーションビルダ仕様を右クリックして、ショートカットメニューから**プロパティ**を選択し、**アプリケーション情報**ページを参照して、アプリケーションのバージョン番号を設定します。共有ライブラリビルド仕様を右クリックして、ショートカットメニューから**プロパティ**を選択し、**共有ライブラリ情**

報ページを参照して、共有ライブラリのバージョン番号を設定します。

- **(Windows)** インストーラのビルドにナショナルインスツルメンツ製のインストーラまたはドライバを含められます。インストーラビルド仕様を右クリックして、ショートカットメニューから**プロパティ**を選択し、**追加インストーラ**ページを参照して、ドライバ、LabVIEW ランタイムエンジンサポート、およびその他のナショナルインスツルメンツの製品のインストーラをインストーラビルドに追加します。
- ビルド仕様で、アプリケーションまたは共有ライブラリのランタイム言語を構成できます。アプリケーションまたは共有ライブラリのビルド仕様を右クリックして、ショートカットメニューから**プロパティ**を選択し、**ランタイム言語**ページを参照して、アプリケーションまたは共有ライブラリの言語の環境設定を行います。
- LabVIEW ランタイムエンジンは多言語対応のため、アプリケーション、共有ライブラリ、またはインストーラとともに複数バージョンのランタイムエンジンを含める必要はありません。
- **(Windows)** インストーラのビルド仕様を構成すると、インストーラにレジストリ入力を含めて複数のファイルのショートカットを作成できます。インストーラビルド仕様を右クリックして、ショートカットメニューから**プロパティ**を選択して、**レジストリ**ページを参照し、カスタムレジストリキーを作成します。ショートカットを作成および構成するには、**ショートカット**ページを使用します。
- ストレージ VI を使用するアプリケーションまたは共有ライブラリのインストーラを作成する場合は、NI USI インストーラを含める必要があります。インストーラビルド仕様を右クリックして、ショートカットメニューから**プロパティ**を選択し、**追加インストーラ**ページを参照して、**含む National Instruments インストーラ**リストから **NI USI** を選択して NI USI インストーラを含めます。

新しいパレットの構成

LabVIEW 8.0 では、**制御器**および**関数**パレットについては、以下の機能強化が行われています。

- パレット項目はカテゴリに従って整理されるようになりました。**標準**および**標準（アイコンまたはテキスト）**形式は、それぞれ、**カテゴリ（標準）**および**カテゴリ（アイコンとテキスト）**形式に置換されました。以下の表は、LabVIEW 7.1 の**制御器**パレットでのトップレベルパレット、および LabVIEW 8.0 でのそのカテゴリ位置を示します。

制御器パレット

表 2　制御器パレットの構成

7.1 パレット	8.0 カテゴリ
Express	Express
数値	モダン
ブール	モダン
文字列 & パス	モダン
配列 & クラスタ	モダン
リスト & 表	モダン
グラフ	モダン
リング & 列挙体	モダン
コンテナ	モダン
I/O	モダン
Refnum	モダン
装飾体	モダン
ダイアログ制御器	システム
旧バージョンの制御器	クラシック

関数パレット

表 3　関数パレットの構成

7.1 パレット	8.0 カテゴリ
Express	Express
ストラクチャ	プログラミング
数値	プログラミング
ブール	プログラミング
文字列	プログラミング
配列	プログラミング
クラスタ	プログラミング
比較	プログラミング
時間 & ダイアログ	プログラミング
ファイル I/O	プログラミング

表 3　関数パレットの構成（続き）

7.1 パレット	8.0 カテゴリ
NI Measurements	測定 I/O
波形	プログラミング
解析	数学および信号処理
計測器 I/O	計測器 I/O
アプリケーション制御	プログラミング
グラフィック & サウンド	プログラミング
通信	接続性およびデータ通信
レポート生成	プログラミング
上級	接続性およびデータ通信
装飾体	プログラミング

- **カテゴリ（標準）**および**カテゴリ（アイコンとテキスト）**形式では、カテゴリを右クリックしてショートカットメニューから**このカテゴリを上に移動**または**このカテゴリを下に移動**を選択すると、パレットに表示されるカテゴリの順番を変更できます。また、カテゴリの左側の 2 重線をクリックし、カテゴリを表示したい場所に矢印をドラッグすることもできます。
- デフォルトでは、旧バージョンの LabVIEW で**上級**パレットに更新して使用していない限り、短縮パレットの表示でインストールされます。短縮パレットの表示では、トップレベルパレットで一部の上級カテゴリを非表示にしています。パレットツールバーで**表示**ボタンをクリックし、ショートカットメニューから**常に表示するカテゴリ→すべてのカテゴリを表示**を選択すると、**制御器**または**関数**パレットですべてのカテゴリが表示されます。
- 現在のパレットまたはすべてのパレットの形式を変更できます。**制御器**または**関数**パレットツールバーで**表示**ボタンをクリックし、**このパレットを別の形式で表示**ショートカットメニューから形式を選択すると、現在のパレットの形式が変更されます。**ツール→オプション**を選択して、制御器 / 関数パレットページを選択して、すべてのパレットの形式を選択します。
- **ツリー形式**—パレット項目をツリー制御器にテキストとして表示します。**制御器**または**関数**パレットツールバーで**表示**ボタンをクリックし、ショートカットメニューから**このパレットを別の形式で表示→ツリー**を選択すると、現在のパレットをツリー形式に設定できます。
- **テキスト**および**ツリー**形式では、**制御器**または**関数**パレットのツールバーで**表示**ボタンをクリックし、ショートカットメニューから**アルファベット順にソート**を選択して、同じレベルにある項目をアルファベット順にソートします。

- **お気に入り**カテゴリを使用して、頻繁にアクセスする項目を**関数**パレットでグループ化します。**カテゴリ（標準）**、**カテゴリ（アイコンとテキスト）**、**アイコン**、**アイコンとテキスト**形式を使用して、**お気に入り**カテゴリに項目を追加できます。ピン付けされた**関数**パレットで、オブジェクトを右クリックしてショートカットメニューから**項目をお気に入りに追加**を選択し、**お気に入り**カテゴリにオブジェクトを追加します。**カテゴリ（標準）**および**カテゴリ（アイコンとテキスト）**形式ではパレットを展開してサブパレットを表示できるので、サブパレットのタイトルを右クリックしてショートカットメニューから**サブパレットをお気に入りに追加**を選択することもできます。

移動および名前が変更されたパレット

以下のパレットは、LabVIEW 8.0 で移動または名前が変更されました。

- **VISA 上級**パレットの**インタフェース専用**および**上位レベルレジスタアクセス**パレットが、**バス / インタフェース特有**および**レジスタアクセス**にそれぞれ名前が変更されました。
- **上級フォーミュラ解析**パレットの VI は、**フォーミュラ解析**および **1D & 2D 評価 VI** パレットに配置されています。
- **クラスタ**および**バリアント**パレットの VI は、**1D & 2D 評価 VI** パレットに配置されています。
- **配列操作**、**カーブフィット**、**フォーミュラ**、**対数関数**、**特殊および数値関数**パレットはそれぞれ、**信号操作**、**フィッティング**、**スクリプト & フォーミュラ**、**指数関数**、**基本 & 特殊関数**パレットと名前変更されました。カーブフィット VI の一部は、**補間 & 補外**パレットにあります。
- **Express 数値定数**パレットは **Express 数学 & 科学定数** パレットに名前変更されました。
- **その他の数値定数**パレットは **Express 数学 & 科学定数** パレットに名前変更されました。
- **Express 対数関数**パレットは **Express 指数関数** パレットに名前変更されました。
- **配列操作ポイントバイポイント**パレットにあった VI は、**幾何学ポイントバイポイント**、**多項式ポイントバイポイント**、**確率と統計ポイントバイポイント**、**信号操作ポイントバイポイント**パレットにあります。
- **カーブフィットポイントバイポイント**パレットにあった VI は、**フィッテングポイントバイポイント**および**補間ポイントバイポイント**パレットにあります。
- **上級**パレットは**ライブラリ & 実行可能ファイル**パレットに名前変更されました。
- **時間 & ダイアログ**パレットにあった VI は、**タイミング**および**ダイアログ & ユーザインタフェース**パレットにあります。
- **周波数領域**パレットにあった VI は、**スペクトル解析**および**変換**パレットにあります。

- **周波数領域ポイントバイポイント**パレットにあった VI は、**スペクトル解析ポイントバイポイント**および**変換ポイントバイポイント**パレットにあります。
- **時間領域**パレットにあった VI は、**信号操作**および**微分積分**パレットにあります。
- **時間領域ポイントバイポイント**パレットにあった VI は、**信号操作ポイントバイポイント**および**微分積分ポイントバイポイント**パレットにあります。
- **タイミングループ**パレットは、**タイミングストラクチャ**パレットに名前変更されました。
- **すべての VI および関数**パレットは**関数**パレットから外されました。

パレット編集の機能強化

LabVIEW 8.0 のパレット編集については、以下の機能強化が行われています。

- LabVIEW 7.*x* 以前では、**制御器**または**関数**パレットのカスタム表示を作成できます。LabVIEW 8.0 はカスタムパレットセットをサポートしていません。カスタムパレットセットを使用せずにパレットセットを編集できます。複数のターゲットで作業している場合は、各ターゲットのパレットセットがロードされます。特定のターゲットのパレットセットを編集するには、変更したいパレットセットのターゲット上の VI から**制御器および関数パレットセットを編集**ダイアログボックスを表示します。**ツール→上級→パレットセットを編集**を選択し、**制御器および関数パレットセットを編集**ダイアログボックスを表示させます。
- パレットセットへの変更は、デフォルトデータディレクトリの `8.0\Palettes` フォルダに保存されます。
- ディレクトリに同期する項目を非表示にするには、その項目を右クリックし、ショートカットメニューから**同期項目を隠す**を選択します。
- 項目がディレクトリに同期する場合は、項目を右クリックしてショートカットメニューから**同期パスを表示**を選択すると、ソースディレクトリへのパスを表示できます。
- 行または列がパレットの終わりに表示され、非表示の同期項目や LabVIEW で検索できない項目のみが含まれる場合、この行または列はパレット編集時にのみ表示されます。
- 空のサブパレットまたは LabVIEW で検索できない項目のみを含むサブパレットは、パレット編集時にのみ表示されます。

メニューの再構成

LabVIEW では多くのメニューや項目が移動または名前変更されました。LabVIEW メニューでは、次の項目が変更されました。

- **ファイル**—プロジェクトに関する基本的な操作（作成、開く、保存、閉じるなど）のメニュー項目が追加されました。

- **編集**—**操作**メニュー項目のいくつかが**編集**メニューに移動され、また新しく**すべてを選択**項目が追加されました。
- **表示**—ウィンドウやパレットの表示、クラスの参照、**プロジェクトエクスプローラ**ウィンドウ表示などを実行するメニュー項目が新しく追加されました。**ウィンドウ**メニュー項目の一部は、**表示**メニューの一部になりました。**参照**メニューが廃止され、**参照**に含まれていたメニュー項目のいくつかは**表示**メニューに追加されました。**参照**メニュー項目を表示するには、**表示→関係を参照**を選択します。
- **プロジェクト**—プロジェクトへの VI やファイルの追加、ビルド仕様の作成、プロジェクト情報の表示を行う項目を含む、新しいメニューが追加されました。
- **操作**—**操作**メニューでは、VI のデバッグの項目が追加または移動されました。**参照**メニューが廃止され、**参照**に含まれていたメニュー項目のいくつかは**操作**メニューに追加されました。
- **ツール**—**ツール**メニューには複数の項目が追加されました。既存の**ツール**メニュー項目は、これまでと異なるサブメニューに配置されています。
- **ウィンドウ**—パレット、**ナビゲーション**ウィンドウ、**エラーリスト**ウィンドウを表示するための項目が削除されました。これらの項目は**表示**メニューに移動されました。
- **ヘルプ**—計測器ドライバを検索し、LabVIEW をアクティブ化する項目が追加されました。

メモ メニュー項目の中には、特定のオペレーティングシステムにインストールされている特定の LabVIEW 開発システムのにおいて、**プロジェクトエクスプローラ**ウィンドウで項目が選択されているとき、または VI が選択されているときにだけ有効になるものがあります。

新しいメニューの場所、名前、使用方法については、『LabVIEW ヘルプ』を参照してください。

ソース管理を使用する

Perforce または Microsoft Visual SourceSafe などの他社のソース管理プロバイダをインストールした場合、LabVIEW のプロジェクト内またはプロジェクトの一部ではない個々のファイルでソース管理を使用できます。VI、プロジェクト内のファイル、プロジェクトライブラリなどのプロジェクトファイルで、ソース管理を実行できます。LabVIEW は、他社のソース管理プロバイダに接続するためのインフラストラクチャを提供します。LabVIEW でソース管理を使用するためには、他社のソース管理プロバイダをインストールする必要があります。**(Windows)** LabVIEW は Microsoft のソースコード管理インタフェースをサポートするソース管理プロバイダを統合します。**(Mac OS および Linux)** Perforce のコマンドラインインタフェースを使用できます。

LabVIEW 8.0 では、ソース管理の機能が以下のように変更されました。

メモ ソース管理はプロフェッショナル開発システムでのみ使用できます。

- LabVIEW 7.*x* 以前のビルトインソース管理プロバイダは、LabVIEW 8.0 では使用できません。旧バージョンのソース管理ファイルの移行の詳細については、「LabVIEW ビルトインソース管理プロバイダから移行する」を参照してください。
- 他社のソース管理プロバイダを選択およびインストールしたら、LabVIEW を構成してそのプロバイダを操作できるようにする必要があります。**ツール→ソース管理→ソース管理**を構成を選択し、**オプション**ダイアログボックスの**ソース管理**ページを表示します。
- LabVIEW 内でソース管理を使用して、編集用にファイルをチェックアウト、編集したファイルをチェックイン、ソース管理から最新バージョンのファイルを取得、そしてソース管理で最新バージョンのファイルを比較できます。**ツール→ソース管理**および実行するソース管理操作を選択します。
- ファイルのソース管理履歴を表示できます。**ツール→ソース管理→履歴**を表示を選択すると、ソース管理でファイルリビジョン履歴を含むプロバイダ特定のダイアログボックスが表示されます。
- ソース管理 VI を使用して、プログラムでソース管理操作を実行します。
- LabVIEW で Perforce の構成ファイルを作成することはできません。

LabVIEW ビルトインソース管理プロバイダから移行する

LabVIEW 7.*x* 以前のビルトインソース管理プロバイダは、LabVIEW 8.0 では使用できません。LabVIEW でソース管理を使用する場合は、他社のソース管理プロバイダを選択する必要があります。旧バージョンでビルトインのプロバイダを使用していた場合は、ファイルを他のプロバイダに移行して LabVIEW でソース管理を使用する必要があります。LabVIEW でサポートしている他社のソース管理プロバイダの最新リストについては、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）にアクセスし、info code の `rdbp01` を参照してください。

ファイルを新規ソース管理プロバイダに移行すると、ビルトインプロバイダに格納されたリビジョン履歴が失われます。旧バージョンのファイルを新規プロバイダに転送することはできません。

ビルトインソース管理プロバイダから他社のソース管理プロバイダにファイルを移行するには、以下の手順に従ってください。

1. 旧バージョンの LabVIEW で、LabVIEW のビルトインソース管理プロバイダに含まれているファイルがすべてのユーザによってチェックインされていることを確認します。

2. 新規ソース管理プロバイダにファイルを追加するコンピュータで、ビルトインプロバイダを使用してすべてのファイルの最新バージョンを取得します。
3. ビルトインプロバイダを使用して、ソース管理からファイルをチェックアウトします。
4. 他社のソース管理プロバイダで、新規ソース管理プロジェクトで使用する設定を構成します。
5. LabVIEW を構成して他社のソース管理プロバイダと使用できるようにします。
6. LabVIEW プロジェクトを作成します。ビルトインプロバイダに含まれるファイルをプロジェクトに追加します。LabVIEW がプロンプトを表示したら、ソース管理にファイルを追加します。また、他社のプロバイダで直接ファイルを追加することもできます。

VI および関数の機能強化

LabVIEW 8.0 では、VI および関数の以下の機能強化および変更が導入されました。

解析 VI の機能強化

LabVIEW 7.*x* 以前の**解析**パレットの VI は、**数学**、**信号処理**、**プログラミング**の 3 つのパレットに分かれて表示されるようになりました。これらの VI のアイコンの多くが変更されました。各アイコンの上部にある黄色のバナーは、VI が位置するサブパレットを示します。

機能向上および変更が行われた VI

以下のとおり、数学、信号処理、プログラミング VI は、LabVIEW 8.0 で機能向上および変更が行われています。

数学パレット

フィッティング VI の機能向上および変更は以下のとおりです。

- 「指数フィット（Exponential Fit）」、「指数フィット係数（Exponential Fit Coefficients）」、「線形フィット（Linear Fit）」、「線形フィット係数（Linear Fit Coefficients）」VI の**メソッド**入力は、使用するフィッティングのメソッドを指定します。最小 2 乗、最小絶対残差、二重平方のフィッティングのメソッドを使用できます。また、これらの VI には**許容範囲**入力があり、選択したメソッドが反復を必要とする場合、出力パラメータの反復調整を停止するタイミングを判断します。これらの VI の**重み付け**入力は、観測データ値の重み付けを指定します。
- 「指数フィット（Exponential Fit）」VI および「線形フィット（Linear Fit）」VI の**残差**は、近似モデルの加重平均の誤差を返します。

- 「指数フィット」VI および「指数フィット係数」VI の**精密化？**入力は、近似モデルの**振幅**と**ダンピング**を更に精密化するかどうかを指定します。
- LabVIEW 7.*x* 以前では、「指数フィット」VI に**標準偏差**入力または **mse** 出力はありません。
- LabVIEW 7.*x* 以前では、「線形フィット」VI に **mse** 出力はありません。
- 「一般 LS 線形フィット（General LS Linear Fit）」VI の**重み付け**入力は、観測データ値の重み付けを指定します。また、この VI には最適フィットを計算する新規アルゴリズムの**行列の階 H の SVD** があります。このアルゴリズムは、入力行列の階数に問題があり、他のすべてのアルゴリズムが失敗した場合にのみ使用されます。LabVIEW 7.*x* 以前では、この VI に**標準偏差**入力はありません。
- 「一般多項式フィット（General Polynomial Fit）」VI の**係数制約**入力は、特定の順序の多項式係数で制約を指定します。この VI の**多項式フィット係数**出力は**多項式係数**に名前変更されました。

微分積分 VI の機能向上および変更は以下のとおりです。

- 「数値積分（Numeric Integration）」VI は、2 次元および 3 次元の数値積分を実行できます。

線形代数 VI の機能向上および変更は以下のとおりです。

- LabVIEW 7.x では、「固有値とベクトル（Eigenvalues and Vectors）」VI の**固有ベクトル**出力は正規化されたので、その最大要素が常に 1 になります。LabVIEW 8.0 では、**固有ベクトル**出力はユークリッドノルムが 1 になるように正規化されます。
- 「QR 分解（QR Decomposition）」VI は「QR 因子分解（QR Factorization ）」VI の代替です。「QR 分解」VI には 2 つの新規入力があります。**ピボット？**はこの VI でピボット列を使用して入力行列 **A** を分解するかどうかを指定します。**Q オプション**は、この VI で直交行列 **Q** を生成する方法を指定します。この VI は、フルサイズ、エコノミーサイズ、または直交でない行列を生成できます。この VI の **P** 出力は置換行列を返します。
- 「QZ 分解（QZ Decomposition）」VI には 2 つの新規入力があります。**分解タイプ**は一般ヘッセンベルグ分解または一般シューア分解を実行するかどうかを指定します。**次数**は、一般化固有値の α と β を順序付ける方法を指定します。
- 「シューア分解（Schur Decomposition）」VI の**次数**入力は、**固有値**とそれに対応する**シューア型**と**シューアベクトル**の順序を指定します。

多項式 VI の機能向上および変更は以下のとおりです。

- 「多項式根（Polynomial Roots）」VI は、複素多項式の根を検索できます。また、この VI は**上級精度**の根を求めるオプションを規定します。
- 「0 係数を削除（Remove Zero Coefficients）」VI の **0 係数の数**出力は、LabVIEW が削除した 0 に近い後続の係数の数値を返します。

確率と統計 VI の機能向上および変更は以下のとおりです。

- 「最頻値（Mode）」VI は多態性で、ユニモーダルおよびマルチモーダル解析の両方を実行します。この VI の**間隔**入力は正でない値を受け取ることができます。**間隔**が正でない場合、「最頻値」VI は入力シーケンスの正確な最頻値のみを計算します。
- 「標準偏差と分散（Standard Deviation and Variance）」VI の **X** 入力は、必須入力ではなくなり、推奨入力になりました。

有理多項式 VI の機能向上および変更は以下のとおりです。

- 「パディ近似（Pade Approximation）」VI の **c(0..n+m)** 入力は、**微分**に名前変更されました。また、**A(0..m)** 出力は**分子**、**B(0..n)** 表示器は**分母**に名前変更されました。

信号処理パレット

上級 FIR フィルタ VI の機能向上および変更は以下のとおりです。

- 「FIR ウィンドウ係数（FIR Windowed Coefficients）」VI の**ウィンドウパラメータ**入力は、カイザー、ガウス、ドルフ・チェビシェフウィンドウの特定のパラメータ値を指定します。この VI の**ウィンドウ**入力のデータタイプは、列挙体から 32 ビット符号なし整数に変更されました。

上級 IIR フィルタ VI の機能向上および変更は以下のとおりです。

- 「カスケード→直接（Cascade → Direct Coefficients）」VI は「カスケードからダイレクト係数（Cascade To Direct Coefficients）」VI に名前変更されました。
- 「IIR カスケードフィルタ（IIR Cascade Filter）」、「IIR カスケードフィルタ初期状態付」、「IIR フィルタ」、「IIR フィルタ初期状態付（IIR Filter with I.C.）」VI は、多態性で複素数入力をサポートします。

フィルタ VI の機能向上および変更は以下のとおりです。

- 「FIR ウィンドウフィルタ（FIR Windowed Filter）」VI の**ウィンドウパラメータ**入力は、カイザー、ガウス、ドルフ・チェビシェフウィンドウの特定のパラメータ値を指定します。この VI の**ウィンドウ**入力のデータタイプは、列挙体から 32 ビット符号なし整数に変更されました。
- 「中央値フィルタ（Median Filter）」VI を除く、**フィルタ**パレットのすべての VI は多態性で複素数入力をサポートします。

信号生成 VI の機能向上および変更は以下のとおりです。

- 「バイナリ M 系列（Binary MLS）」VI には**エラーコード**出力があります。
- 「ランプパターン」VI の**最後を除外？**入力は、生成された配列の最後の値を含むかどうかを指定します。
- 「持続時間による信号生成（Signal Generator by Duration）」VI の**信号タイプ**入力は、必須入力ではなくなり、推奨入力になりました。

信号生成ポイントバイポイント VI の機能向上および変更は以下のとおりです。

- 「ノコギリ波 PtByPt（Sawtooth Wave PtByPt）」VI は最適化され、以下のとおり更新された式を含みます。

  ノコギリ波（sawtooth wave） = 振幅 × ノコギリ (*q*)

  ここで、

$$\text{ノコギリ波}(q) = \begin{cases} \dfrac{q}{180} & q \le 180 \text{ の場合} \\ \dfrac{q}{180} - 2 & \text{その他} \end{cases}$$

  および

  q =(360 · **f** · **時間**) + **位相**) mod (360)、位相（度）になります。

信号操作 VI の機能向上および変更は以下のとおりです。

- 「自己相関（Autocorrelation）」、「コンボリューション（Convolution）」、「相互相関（CrossCorrelation）」VI は、多態性で実数および複素数入力の 2 次元配列をサポートします。
- 「自己相関」および「相互相関」VI の 1 次元インスタンスの**正規化**入力は、自己相関と相互相関の計算にそれぞれ使用する正規化メソッドを指定します。
- 「コンボリューション」VI の 2 次元インスタンスの**出力サイズ**入力は、コンボリューションのサイズを指定します。
- 「正規化ベクトル（Normalize Vector）」VI および「正規化行列（Normalize Matrix）」VI は、「正規化（Normalize）」多態性 VI を構成します。
- 「クイックスケール 1D（Quick Scale 1D）」VI および「クイックスケール 2D（Quick Scale 2D）」VI が、「クイックスケール（Quick Scale）」多態性 VI を構成するようになりました。「クイックスケール 2D（Quick Scale 2D）」VI の **Yij=Xij/Max{X}** パラメータは、**Yij=Xij/Max|X|** に名前変更されました。
- 「リサンプル (定数から定数)（Resample (constant to constant)）」および「リサンプル (定数から変数)（Resample (constant to variable)）」VI は複素数入力をサポートします。
- 「クイックスケール 1D（Quick Scale 1D）」VI および「クイックスケール 2D（Quick Scale 2D）」VI は、「スケール（Scale）」多態性 VI を構成します。
- 「単位ベクトル（Unit Vector）」VI は、多態性で 2 つの新規入力があります。***ノルムタイプ***は、ノルムの計算に使用されるノルムのタイプを示し、**ユーザ定義ノルム**は、***ノルムタイプ***が**ユーザ定義**かどうかを示します。

- 「ゼロパッド（Zero Padder）」VI は、多態性で複素数入力をサポートします。この VI の **2 の累乗以外のみ？**入力は、配列サイズがすでに有効な 2 の累乗の場合に入力配列のサイズを 2 倍にするかどうかを指定します。

スペクトル解析 VI の機能向上および変更は以下のとおりです。

- 「パワースペクトル（Power Spectrum）」VI は、多態性で複素数入力をサポートします。
- 「STFT スペクトログラム（STFT Spectrogram）」VI は「STFT スペクトログラム（STFT Spectrograms）」と名前変更され、2 つの新規入力があります。**時間 - 周波数サンプリング情報**は、時間 / 周波数共同領域内での信号のサンプルに使用する密度を指定し、結果として生じる 2 次元時間 / 周波数配列の大きさを定義します。**ウィンドウ情報**は、短時間フーリエ変換の計算に使用するウィンドウ情報を指定します。ブラックマン・ハリス、完全ブラックマン、フラットトップ、4 次ブラックマン・ハリス、7 次ブラックマン・ハリス、ローサイドロープ、ブラックマン・ナットール、三角ウィンドウが使用できるようになりました。LabVIEW 7.*x* 以前では、この VI に**時間間隔**、**ウィンドウ幅**、**ウィンドウ選択**入力はありません。

変換 VI の機能向上および変更は以下のとおりです。

- 「FFT」および「逆 FFT（Inverse FFT）」VI は、実数および複素数入力の 2 次元配列をサポートします。
- 「FFT」および「逆 FFT」VI の**シフト？**入力は、高速フーリエ変換の中央に DC コンポーネントをシフトするかどうかを指定します。
- 「FFT」VI の 1 次元インスタンスの **FFT サイズ**入力は、実行する FFT の長さを指定します。

変換ポイントバイポイント VI の機能向上および変更は以下のとおりです。

- 「実数 FFT PtByPt（Real FFT PtByPt）」VI および「複素 FFT PtByPt（FFT PtByPt）」は、「FFT PtByPt」多態性 VI を構成します。
- 「実逆 FFT PtByPt（Inverse Real FFT PtByPt）」VI および「複素逆 FFT PtByPt（Inverse Complex FFT PtByPt）」VI は、「逆 FFT PtByPt」多態性 VI を構成します。

波形調節 VI の機能向上および変更は以下のとおりです。

- 「スケールされたウィンドウ（Scaled Window）」は複素数入力をサポートしています。この VI の**ウィンドウパラメータ**入力は、カイザー、ガウス、ドルフ・チェビシェフウィンドウの特定のパラメータ値を指定します。この VI の**ウィンドウ**入力のデータタイプは、列挙体から 32 ビット符号なし整数に変更されました。

波形測定 VI の機能向上および変更は以下のとおりです。

- **ウィンドウ**入力のデータタイプが列挙体から 32 ビット符号なし整数に変更されました。この変更が影響する VI は、「クロススペクトル（振幅 - 位相）（Cross Spectrum (Mag-Phase)）」、「クロススペクト

ル（実 - 虚）（Cross Spectrum (Real-Im)）」、「FFT パワースペクトル密度（FFT Power Spectral Density）」、「FFT パワースペクトル（FFT Power Spectrum）」、「FFT スペクトル（振幅 - 位相）（FFT Spectrum (Mag-Phase)）」、「FFT スペクトル（実 - 虚）（FFT Spectrum (Real-Im)）」、「周波数応答関数（振幅 - 位相）（Frequency Response Function (Mag-Phase)）」、「周波数応答関数（実 - 虚）（Frequency Response Function (Real-Im)）」です。

ウィンドウ VI の機能向上および変更は以下のとおりです。

- 「余弦テーパーウィンドウ（Cosine Tapered Window）」VI の**比率**入力は、信号全体の長さに対するテーパーセクションの長さの比率を指定します。この VI の**余弦 α 漸減 {X}** 出力は、**ウィンドウ処理後の X** に名前変更されました。
- 「スケール時間領域ウィンドウ（Scaled Time Domain Window）」VI の**ウィンドウパラメータ**は、適用するウィンドウのパラメータを指定します。また、この VI には新規の**エラー**出力があります。この VI の**波形**入力、**ウィンドウ処理済み波形**出力、**ウィンドウ定数**出力は、**X**、**ウィンドウ処理後の X**、**ウィンドウプロパティ**にそれぞれ名前変更されました。また、「スケール時間領域ウィンドウ」VI は、ブラックマン・ナットール、三角、カイザー、ドルフ・チェビシェフ、ガウスウィンドウを適用するようになりました。この VI の**ウィンドウ**入力のデータタイプは、列挙体から 32 ビット符号なし整数に変更されました。
- 「ウィンドウプロパティ（Window Properties）」VI を除いて、**ウィンドウ**パレット上のすべての VI は多態性で複素数入力をサポートします。

プログラミングパレット

比較 VI の機能向上および変更は以下のとおりです。

- 「等しい？（Equal?）」および「等しくない？（Not Equal?）」比較関数は、2 つの VI サーバリファレンスが同じオブジェクトを参照するかどうかを判断できます。

新規の VI および関数

LabVIEW 8.0 では、以下のとおり新規に数学、信号処理、プログラミング VI が追加されています。

数学パレット

上級カーブフィットサブパレットには、以下の新しい VI があります。

- 「指数フィット間隔（Exponential Fit Intervals）」VI
- 「ガウスピークフィット係数（Gaussian Peak Fit Coefficients）」VI
- 「ガウスピークフィット間隔（Gaussian Peak Fit Intervals）」VI
- 「フィットの適合度（Goodness of Fit）」VI

- 「線形フィット間隔（Linear Fit Intervals）」VI
- 「対数フィット係数（Logarithm Fit Coefficients）」VI
- 「対数フィット間隔（Logarithm Fit Intervals）」VI
- 「累乗フィット係数（Power Fit Coefficients）」VI
- 「累乗フィット間隔（Power Fit Intervals）」VI
- 「外れ値を削除（Remove Outliers）」VI

微分方程式サブパレットには、新しい「ODE ソルバー（ODE Solver）」VI があります。

離散数学サブパレットには、以下の新しい VI があります。

- 「GCD」VI
- 「LCM」VI
- 「順列（Permute）」VI

指数関数サブパレットには、以下の新しい VI があります。「x の y 番目のルート（y-th root of x）」関数

フィッティングサブパレットには、以下の新しい VI があります。

- 「3 次スプラインフィット（Cubic Spline Fit）」VI
- 「ガウスピークフィット（Gaussian Peak Fit）」VI
- 「対数フィット（Logarithm Fit）」VI
- 「非線形カーブフィット（Nonlinear Curve Fit）」VI
- 「累乗フィット（Power Fit）」VI

幾何学サブパレットには、以下の新しい VI があります。

- 「2D 直交座標移動（2D Cartesian Coordinate Shift）」VI
- 「2D 直交座標回転（2D Cartesian Coordinate Rotation）」VI
- 「3D 直交座標移動（3D Cartesian Coordinate Shift）」VI
- 「3D 直交座標回転 (オイラー)（3D Cartesian Coordinate Rotation (Euler)）」VI
- 「3D 直交座標回転 (方向)（3D Cartesian Coordinate Rotation (Direction)）」VI
- 「3D 座標変換（3D Coordinate Conversion）」VI
- 「方向余弦からオイラー角に変換（Direction Cosines to Euler Angles）」VI
- 「オイラー角から方向余弦に変換（Euler Angles to Direction Cosines）」VI

双曲線関数サブパレットには、以下の新しい VI があります。

- 「Hyperbolic Cosecant」関数
- 「Hyperbolic Cotangent」関数

- 「Hyperbolic Secant」関数
- 「Inverse Hyperbolic Cosecant」関数
- 「Inverse Hyperbolic Cotangent」関数
- 「Inverse Hyperbolic Secant」関数

仮説検定サブパレットには、以下の新しい VI があります。

- 「相関検定（Correlation Test)」VI
- 「符号検定（Sign Test)」VI
- 「T 検定（T Test)」VI
- 「ウィルコクスンの符号付き順位検定（Wilcoxon Signed Rank Test)」VI
- 「Z 検定（Z Test)」VI

微分積分サブパレットには、新しい「ロバット求積法（Lobatto Quadrature)」VI があります。

補間 & 補外サブパレットには、以下の新しい VI があります。

- 「補間多項式を作成（Create Interpolating Polynomial)」VI
- 「メッシュグリッドを作成 (2D)（Create Mesh Grid (2D))」VI
- 「補間多項式を評価（Evaluate Interpolating Polynomial)」VI
- 「エルミート補間 1D（Hermite Interpolation 1D)」VI
- 「1D 補間（Interpolate 1D)」VI
- 「1D フーリエ補間（Interpolate 1D Fourier)」VI
- 「2D 補間（Interpolate 2D)」VI
- 「順序付けられた表を検索（Search Ordered Table)」VI
- 「スプライン補間 1D（Spline Interpolation 1D)」VI

数値サブパレットには、以下の新しい関数があります。

- マシンイプシロン（Machine Epsilon）定数
- 「方形波（Square)」関数

最適化サブパレットには、以下の新しい VI があります。

- 「制約付き非線形最適化（Constrained Nonlinear Optimization)」VI
- 「2 次計画（Quadratic Programming)」VI
- 「制約のない最適化（Unconstrained Optimization)」VI

多項式サブパレットには、新しい「多項式プロット（Polynomial Plot)」VI があります。

確率サブパレットには、以下の新しい VI があります。

- 「連続 CDF（Continuous CDF）」VI
- 「連続逆 CDF（Continuous Inverse CDF）」VI
- 「連続モーメント（Continuous Moments）」VI
- 「連続 PDF（Continuous PDF）」VI
- 「連続乱数（Continuous Random）」VI
- 「離散 CDF（Discrete CDF）」VI
- 「離散逆 CDF（Discrete Inverse CDF）」VI
- 「離散モーメント（Discrete Moments）」VI
- 「離散 PF（Discrete PF）」VI
- 「離散乱数（Discrete Random）」VI

確率と統計サブパレットには、以下の新しい VI があります。

- 「相関係数（Correlation Coefficient)」VI
- 「相関係数 (ケンドールの τ)（Correlation Coefficient (Kendall's Tau)）」VI
- 「相関係数（スピアマン）（Correlation Coefficient (Spearman)）」VI
- 「共分散行列（Covariance Matrix）」VI
- 「平均の測定（Measures of Mean）」VI
- 「分散の測定（Measures of Spread）」VI
- 「パーセント（Percentiles）」VI

三角関数サブパレットには、以下の新しい VI があります。

- 「Inverse Cosecant」関数
- 「Inverse Cotangent」関数
- 「Inverse Secant」関数

信号処理パレット

フィルタサブパレットには、新しい「ゼロ位相フィルタ（Zero Phase Filter）」VI があります。

信号操作サブパレットには、新しい「リフル（Riffle）」VI があります。

変換サブパレットには、以下の新しい VI があります。

- 「チャープ Z 変換（Chirp Z Transform）」VI
- 「DCT」VI
- 「DST」VI
- 「逆 DCT（Inverse DCT）」VI
- 「逆 DST（Inverse DST）」VI

波形調節サブパレットには、新しい「連続コンボリューション（FIR）（Continuous Convolution (FIR)）」VI があります。

波形測定サブパレットには、新しい「マルチトーン情報抽出（Extract Multiple Tone Information）」VI があります。

ウィンドウサブパレットには、以下の新しい VI があります。

- 「ブラックマン・ナットールウィンドウ（Blackman-Nuttall Window）」VI
- 「チェビシェフウィンドウ（Chebyshev Window）」VI
- 「ガウスウィンドウ（Gaussian Window）」VI
- 「対称ウィンドウ（Symmetric Window）」VI
- 「ウィンドウプロパティ（Window Properties）」VI

プログラミングパレット

比較サブパレットには、以下の新しい関数があります。

- 「空の配列？（Empty Array?）」関数
- 「極座標を直交座標に変換（Polar to Re/Im）」関数
- 「直交座標を極座標に変換（Re/Im to Polar）」関数

デジタル波形 VI の機能強化

デジタル波形 VI を使用して、デジタル波形（DWDT）またはデジタル表（DTbl）の一部の置換、デジタル波形データまたはデジタルデータからアナログ波形またはバイナリデータ形式への変換、アナログ波形、バイナリ形式、またはスプレッドシート文字列からデジタル波形データまたはデジタルデータへの変換を行います。「デジタルパターン発生器（Digital Pattern Generator）」VI で、さまざまな塗りつぶしパターンを使用してデジタルパターンを生成します。「空のデジタルデータ（Empty Digital Data）」および「空のデジタル波形（Empty Digital Waveform）」定数で空のデジタル波形データまたはデジタルデータを返します。これは、シフトレジスタの初期化または波形作成に重要です。

機能向上および変更が行われた VI

以下のとおり、デジタル波形 VI は、LabVIEW 8.0 で機能向上および変更が行われています。

- 「デジタル反転（Invert Digital）」VI には、2 つの新規入力があります。**信号指標**は、データの反転を開始する信号を指定します。**信号数**は、反転させる信号数を指定します。
- 「デジタルパターン検索（Search for Digital Pattern）」VI の**比較モード**入力は、X の値を検索時にどのように扱うかを指定します。
- 「アナログからデジタル（Analog to Digital）」VI は、多態性で、アナログ波形をデジタルデータに変換できます。

- 「デジタルからアナログ（Digital to Analog）」VI は、多態性で、デジタルデータをアナログ波形に変換できます。

新規の VI および関数

LabVIEW 8.0 では、以下のとおり新しいデジタル波形 VI および関数が導入されました。

- 「バイナリからデジタル（Binary to Digital）」VI
- 「デジタルパターン発生器（Digital Pattern Generator）」VI
- デジタルリング（Digital Ring）定数
- 「デジタルからバイナリ（Digital to Binary）」VI
- 空のデジタルデータ（Empty Digital Data）定数
- 空のデジタル波形（Empty Digital Waveform）定数
- 「部分置換（Replace Subset）」VI
- 「スプレッドシート文字列からデジタル（Spreadsheet String to Digital）」VI

無効ストラクチャ

条件無効ストラクチャを使用し、ストラクチャの各サブダイアグラムを記号と値から構成される表記に割り当ててコードのセクションを無効にします。この記号と値は、ユーザ定義および LabVIEW 定義の両方で、これにより LabVIEW でどのサブダイアグラムを使用するかを判断します。

メモ 実行しないコードは、ブロックダイアグラムでコンパイルされません。たとえば、条件無効ストラクチャの非アクティブなサブダイアグラム内にあるコードはコンパイルされません。また、定数が配線されたケースストラクチャで、ケースストラクチャ内の実行されないケース内のコードはコンパイルされません。

ダイアグラム無効ストラクチャを使用して、コンパイルしたくないブロックダイアグラムの部分を定義します。

Express VI の機能強化

Express VI および関数を使用して、一般的な計測タスクを作成します。

機能向上および変更が行われた VI

以下のとおり、Express VI は、LabVIEW 8.0 で機能向上および変更が行われています。

- 「比較（Comparison）」Express VI には、2 つの新規入力があります。**比較する項目**では、データポイント、タイムスタンプ、ポイント間の時間、データポイント数、入力信号の信号名を比較できます。**比較定数値（Constant Value to Compare Against）**は、**オペラン**

ド 1 入力の比較対象とする定数値を指定します。また、この VI は最適化され、最初の空の入力信号のケースでより適切な結果を返します。

- 「LabVIEW 計測ファイル読み取り（Read LabVIEW Measurement File）」および「LabVIEW 計測ファイル書き込み（Write LabVIEW Measurement File）」Express VI は、それぞれ、「計測ファイルから読み取り（Read From Measurement File）」および「計測ファイルへ書き込み（Write To Measurement File）」に名前が変更されました。これらの Express VI では、バイナリ計測ファイル (`.tdm`) と同様にテキストベースの計測ファイル (`.lvm`) をサポートします。ファイル形式をファイルへ設定するには、構成ダイアログボックスで**バイナリ（TDM）**オプションを選択します。
- 「計測ファイルから読み取り」、「計測ファイルへ書き込み」、「ユーザ入力プロンプト（Prompt User for Input）」、および「レポート（Report）」Express VI は、ターゲットにより動作が変更されます。現在のターゲットに接続されたホストコンピュータがない（またはない可能性がある）場合、構成ダイアログボックスでは、ホストがない場合は無効となるオプションの隣りに警告が表示されます。ホストコンピュータが接続されていない LabVIEW Real-Time モジュールなど、入力用にプロンプトを出すようにこれらの VI を構成し、ターゲット上でユーザインタフェースなしで VI を実行すると、これらの VI はエラーを返します。

  Windows プラットフォーム以外に「レポート」VI のターゲットを指定すると、構成ダイアログボックスは出力先として **HTML（ウェブページ）**のみをサポートします。
- 「リレー（Relay）」Express VI は、**有効**ブロックダイアグラム入力に値を配線しない場合、**有効**入力の構成ダイアログボックス値を使用します。
- 「信号選択（Select Signals）」Express VI の**セレクタ入力**入力は、**信号**入力から信号の一部を選択できます。
- 「データの書き込み（Write Data）」および「計測ファイルへ書き込み（Write To Measurement File）」Express VI は最適化され、固有のチャンネル名になります。ファイルまたはチャンネル名が繰り返される場合、それぞれを固有の名前にするために、名前の最後に整数値が追加されます。たとえば、正弦、正弦、方形、方形、正弦という名前を規定すると、LabVIEW ではこれらの名前をそれぞれ正弦、正弦 1、方形、方形 1、正弦 2 に更新します。

新規の VI および関数

LabVIEW 8.0 では、以下のとおり新しい Express VI が導入されました。

- 「サウンドを集録（Acquire Sound）」Express VI
- 「デュアルチャンネルスペクトル計測（Dual Channel Spectral Measurement）」Express VI
- 「波形を再生（Play Waveform）」Express VI
- 「NI DIAdem レポート（DIAdem Report）」Express VI

DIAdem レポート Express VI

LabVIEW のデータから直接レポートを生成するには、「DIAdem レポート（DIAdem Report）」Express VI を使用します。レポートは、PDF ファイルや HTML ファイルとしてエクスポートしたり、印刷したり、自動的に表示したりできます。レポートの設定は、すべて LabVIEW 内で行います。

メモ 「DIAdem レポート（DIAdem Report）」Express VI を使用するためには、DIAdem 9.1 Service Pack 2 以降がインストールされていなければなりません。DIAdem の評価版をダウンロードまたは DIAdem を購入するには、ナショナルインスツルメンツのウェブサイト `ni.com/jp` にアクセスしてください。

ファイル I/O VI および関数の機能強化

ファイル I/O パレットには更新された VI、および新規や更新された関数が含まれ、パス入力の機能強化や新規ダイアログ関数など、ファイルのデータ読み取りと書き込みがより簡単になりました。また、**ファイル I/O** パレットには、Zip ファイルでの作業に使用できる新しい VI も含まれます。

パス入力

入力としてパスまたは Refnum データを受け入れるファイル I/O 関数には、**パス（ダイアログを使用）**入力などの位相（ダイアログを使用）を持つ入力が含まれます。この入力を配線しないと、ファイルの実行時に、ユーザにファイルまたはディレクトリを選択するように促すプロンプトを表示するダイアログボックスが表示されます。これらの関数はそれぞれ**キャンセルされた**出力を含みます。ユーザがダイアログボックスで**キャンセル**ボタンをクリックすると、**キャンセルされた**は TRUE を返し、関数がエラーを返します。パスをファイル I/O 関数に配線すると、関数は操作の実行に必要な最小限の権限付きでファイルを開き、ファイルを明示的に開いたり閉じたりする必要がないようにします。

データログ関数

データログ関数を使用してデータログファイルの書き込みおよび読み取りを行います。**ダイアログ**パレットには以下の新しい関数が含まれます。

- 「開く / 作成 / 置換データログ（Open/Create/Replace Datalog）」関数
- 「データログ位置を所得（Get Datalog Position）」関数
- 「レコード数を取得（Get Number of Records）」関数
- 「データログ読み取り（Read Datalog）」関数
- 「データログ位置を設定（Set Datalog Position）」関数
- 「レコード数を設定（Set Number of Records）」関数
- 「データログに書き込み（Write Datalog）」関数

ファイル I/O VI および関数

ファイル I/O VI および関数を使用して、ファイルの開閉、ファイルからの読み取りやファイルへの書き込み、ディレクトリやファイルの作成、ディレクトリ情報の取り出しを行います。また、文字列、数値、配列、およびクラスタのファイルへの書き込みを行います。**ファイル I/O** パレットには以下の新しい関数が含まれます。

- 「開く / 作成 / 置換ファイル（Open/Create/Replace File）」関数
- 「バイナリファイルから読み取り（Read from Binary File）」関数
- 「テキストファイルから読み取り（Read from Text File）」関数
- 「バイナリファイルに書き込み（Write to Binary File）」関数
- 「テキストファイルに書き込み（Write to Text File）」関数

Zip VI

Zip VI を使用して、新しい Zip ファイルを作成したり、Zip ファイルにファイルを追加したり、Zip ファイルを閉じることができます。

その他の機能強化と変更点

LabVIEW 8.0 では、その他のファイル I/O VI および関数の以下の機能強化および変更が導入されました。

- 「コピー（Copy）」関数には、**上書き**と**プロンプト**入力、および**キャンセル**出力が含まれます。
- 「削除（Delete）」関数には**階層全体**入力が含まれ、これが TRUE の場合、指定した**パス**またはパスの下の階層全体を削除します。
- 「ファイルダイアログ（File Dialog）」関数は、「ファイルダイアログ（File Dialog）」Express VI になりました。この VI をダブルクリックして**選択モード**ダイアログボックスを表示し、ユーザがファイルダイアログボックスで選択できるファイルまたはディレクトリのタイプを構成します。
- 「ファイルを閉じる（Close File）」、「コピー（Copy）」、「フォルダを作成（Create Folder）」、「削除（Delete）」、「ファイル / ディレクトリ情報（File/Directory Info）」、「ファイル更新（Flush File）」、「移動（Move）」関数には、新しいコネクタペーンの構成があります。新しい構成は、旧バージョンの LabVIEW で作成したブロックダイアグラム上のワイヤ位置に影響することがあります。
- 「移動（Move）」および「コピー（Copy）」関数を使用すると、既存のファイルおよびディレクトリを上書きできます。
- 「ファイル / ディレクトリ情報（File/Directory Info）」関数には、新しく 2 つの出力が追加されました。**ショートカット**はファイルまたはディレクトリがショートカットかどうかを示し、**解決されたパス**はショートカットのターゲットへのパスを返します。

- 「キー読み取り（Read Key）」VI の文字列インスタンスには、LabVIEW 7.*x* 以前の**マルチバイトエンコーディング（0: なし）**パラメータは含まれません。

計測器 I/O アシスタントの機能強化

計測器 I/O アシスタントについては、以下の機能強化が行われています。

- **計測器を選択**の**終了文字**プルダウンメニューから、計測器のコマンド終了文字を示す文字を選択できます。また、< **カスタム** > を選択して**終了文字**テキストボックスに終了文字を入力することもできます。
- **書き込み**ステップの**コマンド入力**でコマンドを選択し、**パラメータを追加**ボタンをクリックして、計測器コマンドに入力パラメータを追加できます。また、コマンドを選択して右クリックして、ショートカットメニューから**パラメータを追加**を選択することもできます。
- **クエリして解析**ステップを**書き込み**と**読み込んでから解析**ステップに分割すると、パラメータ化された入力を使用できます。**クエリして解析**ステップを分割するには、**ステップのシーケンス**ウィンドウでステップを右クリックし、ショートカットメニューから**ステップを書き込みと読み込みステップに分解**を選択します。
- 計測器 I/O アシスタントは、トークンのデータを、解析するデータがなくなるまですべて構文解析するように設定できます。計測器 I/O アシスタントを構成してデータの終わりまで構文解析するには、**クエリして解析**ステップの**カウント**プルダウンメニューから**データの終わりまで**項目を選択するか、**カウント**プルダウンメニューの下で**データの終わりまで**をクリックします。また、計測器 I/O アシスタントは、**自動解析**ボタンをクリックした場合も、データの終わりまで構文解析します。**データの終わりまで**項目と操作は、構成したトークンが最後のトークンの場合にのみ利用できます。

サウンド VI の機能強化

サウンドパレットには、ご使用の VI にサウンドを組み入れたり、VI でサウンドを録音する場合に使用する VI が含まれます。

メモ **(Windows)** サウンド VI を使用するには DirectX 8.0 以降が必要です。

これらの VI には、以下のような機能強化が行われています。

- VI は最大 24 ビットサウンドまでサポートします。
- VI は、複数チャンネルサウンドファイルに加えて、モノおよびステレオサウンドをサポートします。
- 波形はサウンドデータを表します。符号なし 8 ビット、符号付き 16 ビット、または符号付き 32 ビットの整数、あるいは単精度および倍精度のデータタイプの要素を使用すると、Y 配列データを表すことができます。各波形は 1 つのチャンネルを定義します。
- サウンドデータの形式は、PCM（Pulse Code Modulated）です。

- VI は、連続するサウンド出力を生成できます。
- VI は、波形ファイルのストリーミング表示を可能にします。
- VI は、エラーチェックの機能が向上しています。

文字列関数の機能強化

PCRE（Perl Compatible Regular Expression）構文を使用する複雑な正規パターンの一致には、「正規表現で一致（Match Regular Expression）」および「文字列の検索と置換（Search and Replace String）」関数を使用します「文字列の検索と置換」関数を右クリックして、ショートカットメニューから**正規表現**を選択して、上級正規表現の検索および置換文字列でのサブマッチの置換を行うために関数を構成します。文字列で検索されたサブマッチを表示するには、「正規表現と一致（Match Regular Expression）」関数のサイズを変更します。「パターンで一致（Match Pattern）」関数または「正規表現で一致」関数を使用して、文字列で正規表現を検索することができます。「正規表現で一致」関数は、「パターンで一致」関数より多くの文字列一致のためのオプションがありますが、速度は遅くなります。

VI 階層ウィンドウの機能強化

表示→ VI 階層を選択して **VI 階層**ウィンドウを表示します。この VI 階層ウィンドウは、メモリ内のすべての VI の呼び出している階層だけでなく、開いている LabVIEW オブジェクトとターゲットを表示します。**VI 階層**ウィンドウには、メインの LabVIEW アプリケーションインスタンスを表すトップレベルアイコンが表示されます。その下に、プロジェクトに含まれない開いている VI がすべて表示されます。**VI 階層**ウィンドウには、メモリ内の各プロジェクトがトップレベルアイコンとして表示されます。追加する各ターゲットはプロジェクトの下に表示されます。

フロントパネルの機能強化

LabVIEW 8.0 では、フロントパネルの以下の機能強化と変更が行われました。

グラフカーソルの変更点

LabVIEW 8.0 では、グラフカーソルが以下のように変更されました。

- カーソルを作成するには、カーソル凡例を右クリックして、**カーソルを作成**を選択し、ショートカットメニューからカーソルモードを選択します。カーソルには以下のモードが含まれています。

表 4　LabVIEW 8.0 のカーソルモードの変更点

LabVIEW 7.1 のカーソルモード	LabVIEW 8.0 のカーソルモード	LabVIEW 8.0 カーソルモードの詳細
解放	**解放**	プロットの位置に関係なく、プロット領域内で自由にカーソルを移動します。
ポイントにスナップと**プロットにロック**	**シングルプロット**	カーソルに関連付けられたプロットのみにカーソルを配置します。このカーソルを関連プロットとともに移動できます。カーソルの凡例の行を右クリックして、ショートカットメニューから**スナップ**を選択して、1 つまたは複数のプロットをカーソルと関連付けます。
—	**マルチプロット**	プロット領域の特定のデータポイントのみにカーソルを配置します。マルチプロットのカーソルは、カーソルが関連するすべてのプロットについて、指定された x 値における値をレポートします。カーソルはプロット領域内のすべてのプロットに配置できます。カーソルの凡例の行を右クリックして、ショートカットメニューから**スナップ**を選択して、1 つまたは複数のプロットをカーソルと関連付けます。このモードはミックスドシグナルのグラフのみに有効です。

- 作成後にカーソルのモードを変更できません。カーソルを削除して、他のカーソルを作成する必要があります。カーソルの凡例の行を右クリックして、ショートカットメニューから**カーソルを削除**を選択してカーソルを削除します。
- **カーソルムーバセレクタ**、**フォーマットメニュー**、および**ロックメニュー**ボタンは、カーソル凡例の右側に表示されません。カーソルの凡例の行を右クリックして、ショートカットメニューからオプションを選択して、カーソルをカスタマイズします。旧バージョンの LabVIEW でカーソル凡例ボタンを使用してアクセスしたすべてのオプションが、ショートカットメニューに表示されるようになりました。
- カーソルの X スケールをプログラムで設定するには、X スケールプロパティを使用します。このプロパティは、カーソルモードが**解放**の場合のみに有効です。このプロパティは、グラフのカーソル凡例のショートカットメニューにある **X スケール**項目に似ています。

- カーソルの Y スケールをプログラムで設定するには、Y スケールプロパティを使用します。このプロパティは、カーソルモードが**解放**の場合のみに有効です。このプロパティは、グラフのカーソル凡例のショートカットメニューにある **Y スケール**項目に似ています。
- プロット領域のプロットをプログラムで監視するには、プロットを監視プロパティと全てのプロットを監視プロパティを使用します。これらのプロパティは、複数のプロットに関連付けられているカーソルでのみ有効です。

グラフの注釈を使用する

プロット領域でデータ点をハイライトするには、グラフで注釈を使用します。注釈には、注釈とデータ点を識別するラベルおよび矢印が含まれます。注釈の外観の設定やプロット領域でのプロットへのスナップの仕方を設定することもできます。注釈を作成するには、グラフを右クリックし、ショートカットメニューから**データ操作→注釈を作成**を選択して**注釈を作成**ダイアログボックスを開きます。注釈の設定を変更するには、注釈を右クリックしてショートカットメニューからオプションを選択します。注釈リストプロパティを使用して、プロット領域のすべての注釈に関する情報の配列をプログラムで取得します。

グラフのプロット領域で画像を描画する

グラフの背景、前景、中間プロット領域に画像を描画できます。プロット領域を背景、前景、中間のプロット画像にそれぞれ設定するには、波形グラフ、強度グラフ、ミックスドシグナルグラフのプロット画像 : 背景、プロット画像 : 前景、プロット画像 : 中間プロパティを使用します。プロット画像 : 前景プロパティは強度グラフにのみ使用できます。プロット画像は、前景と中間プロット画像の間に表示されます。グリッドは、中間プロット画像と背景の間に表示されます。背景、前景、または中間のプロット領域から既存の画像を削除する場合、空の画像を適切なプロパティに配線するか、画像を透明度にする必要があります。

メモ グラフをサイズ変更しても、プロット領域の画像はそれに従って調整されません。VI を再実行して、画像を再描画する必要があります。

グラフの座標をフロントパネルの座標にマップし、グラフのプロット領域で適切に画像を描画する必要があります。グラフまたはチャートの座標をフロントパネルの座標にマップするには、座標を XY にマップメソッドを使用します。このメソッドは、フロントパネルの原点に対するプロットの方向を決定する左と上のプロット境界のみを調整します。

プロット画像プロパティおよび座標を XY にマップメソッドを使用するサンプルについては、`labview¥examples¥general¥graphs¥Graph Pictures.llb` を参照してください。

グラフ、チャート、表の画像をエクスポートする

グラフ、チャート、表、デジタルデータ、デジタル波形制御器と表示器の画像をエクスポートできます。

(Windows) `.emf`、`.bmp`、および `.eps` ファイル

(Mac OS) `.pict`、`.bmp`、および `.eps` ファイル

(Linux) `.bmp` および `.eps` ファイル

画像をエクスポートするには、制御器を右クリックしてショートカットメニューから**データ操作→簡略化された画像をエクスポート**を選択します。**簡略化された画像をエクスポート**ダイアログボックスでは、画像形式と、画像をクリップボードとディスクのどちらに保存するかを選択できます。また、画像をエクスポートメソッドを使用すると、画像をプログラムでエクスポートすることもできます。

XY グラフでペーンを表示する

XY グラフでナイキスト平面、ニコルズ平面、S 平面、または Z 平面を表示することができます。これらの平面を表示するには、XY グラフを右クリックしてショートカットメニューから**任意の平面**を選択します。**任意の平面**メニュー項目には、ナイキスト平面、ニコルズ平面、S 平面、または Z 平面など、XY グラフに表示できるさまざまなタイプの平面を選択できます。任意の平面を 1 つ選択すると、**任意の平面**メニューには、平面線、平面ラベル、直交線を表示または非表示にするためのメニューが表示されます。

また、任意平面 : 平面タイププロパティを使用して、プログラムで XY グラフ上のこれらの平面を表示することもできます。プログラムで XY グラフの平面ラインおよび平面ラベルを表示または非表示にするには、任意平面 : ライン表示プロパティおよび任意平面 : ラベル表示プロパティを使用します。プログラムによって直交線を XY グラフに表示するには、直交線 : 表示プロパティを使用します。

また、XY グラフは、x 軸で実部がプロットされて y 軸で虚部がプロットされた複素データの配列も受け入れます。

ミックスドシグナルグラフ

共通の x スケールを共有する異なるタイプのデータをプロットするには、ミックスドシグナルグラフを使用します。ミックスドシグナルグラフは、波形グラフ、XY グラフ、デジタル波形グラフにより受け入れられたすべてのデータタイプを受け入れます。また、これらのデータタイプの組み合わせを含む任意のクラスタも受け入れます。

複数のプロット領域をミックスドシグナルグラフに追加すると、各プロット領域には固有の y スケールができます。アナログおよびデジタルデー

タの組み合わせに対応する必要がある場合、ミックスドシグナルグラフは自動的にプロット領域を作成します。

ミックスドシグナルグラフ上のプロット凡例はツリー制御器から構成され、グラフのプロット領域の左に表示されます。各ツリー制御器は 1 つのプロット領域を表します。プロット凡例を使用すると、プロット領域間でプロットを移動できます。

ミックスドシグナルグラフで複数プロットカーソルを使用すると、特定の x 値で複数プロットの値をレポートできます。

キー操作

プロパティダイアログボックスの**キー操作**ページを表示するには、フロントパネルで制御器を右クリックし、ショートカットメニューから**上級→キー操作**を選択します。また、このページは制御器を右クリックしてショートカットメニューから**プロパティ**を選択し、**キー操作**タブをクリックしても表示されます。**キー操作**ページには、次の変更点があります。

- **キー定義**セクションの名前が**フォーカス**に変更されました。**フォーカス**リストには、<Delete>、<Insert>、<Mute>、<Play>、<Volume Up>、<Volume Down>、F13、F14、F15、F16、F17、F18、F19、F20、F21、F22、F23、F24 などのショートカットキーがあります。
- **現在のキー定義**セクションの名前が**既存の連結**に変更されました。
- **キー操作**ページには、ブール制御器をトグルするショートカットを割り当てる**トグル**セクションがあります。トグルキーバインディングプロパティを使用してブール制御器をトグルするショートカットを割り当てることもできます。
- **キー操作**ページには、数値、スライド、リング、列挙体制御器の値を増分または減分するための**増分**セクションと**減分**セクションがあります。また、キーバインディングを増分プロパティとキーバインディングを減分プロパティによって数値制御器の値を増分、減分するためのショートカットキーをそれぞれ割り当てることもできます。

キー操作プロパティの名前は、フォーカスキーバインディングに変更されました。ブール制御器に対しては、このプロパティではフォーカスキーボードショートカットだけを設定し、トグルキーボードショートカットはトグルキーバインディングプロパティで設定します。

最小フロントパネルサイズ

LabVIEW 7.*x* 以前では、フロントパネルの最小サイズはそのペーンの内容領域を参照し、スクロールバーは含まれません。LabVIEW 8.0 では、1 つのペーンで構成されるフロントパネルについて、最小サイズは同様にそのペーンの内容領域を参照しますが、スクロールバーは含まれません。ただし、複数のペーンで構成されるフロントパネルについて、最小サイズは表示されているスクロールバーも含めて、フロントパネル全体を参照します。

スプリッタバーおよびペーン

フロントパネルを別々にスクロールできる複数のペーンに分割するには、**コンテナ**パレットでスプリッタバーを使用します。これらのペーンを使用すると、複数のカスタマイズ可能な領域でユーザインタフェースを設計できます。また、スプリッタバーを使用すると、ツールバーまたはステータスバーとしてペーンを指定できます。

プログラムでペーンを制御するには、ペーンのプロパティ、メソッド、イベントを使用します。

プログラムでスプリッタバーを制御するには、スプリッタのプロパティおよびイベントを使用します。

LabVIEW 8.0 では、LabVIEW 7.*x* 以前の**ウィンドウの外観をカスタマイズ**ダイアログボックスにある**スクロールバーを表示**オプションは、2 つの新規オプションの**水平スクロールバーを表示**および**垂直スクロールバーを表示**から構成されます。スプリッタバーをフロントパネルに追加すると、ダイアログボックスでこれらのオプションを変更することができません。オプションは、フロントパネルにスプリッタバーがある場合に左上隅のペーンの設定を反映します。

ランタイムショートカットメニューをカスタマイズする

VI にある各制御器のランタイムショートカットメニューをカスタマイズできます。ショートカットメニューをカスタマイズするには、制御器を右クリックして、ショートカットメニューから**上級→ランタイムショートカットメニュー→編集**を選択し、**ショートカットメニューエディタ**ダイアログボックスを表示します。デフォルトのショートカットメニューまたはカスタマイズされたショートカットメニューファイル（`.rtm`）を制御器に関連付けるにはこのダイアログボックスを使用します。プログラムでショートカットメニューをカスタマイズするには、ショートカットメニューを起動？イベント、「メニュー項目を挿入（Insert Menu Items）」関数、および「メニュー項目を削除（Delete Menu Items）」関数を使用します。

また、フロントパネルにショートカットメニューを追加することもできます。フロントパネルにショートカットメニューを追加するには、ショートカットメニュー起動およびショートカットメニュー選択ペーンイベントを使用します。

また、制御器を右クリックして、ショートカットメニューから**上級→ランタイムショートカットメニュー→無効**を選択して、制御器でランタイムショートカットメニューを無効にします。

すべての制御器のデフォルトショートカットメニューを無効にするには、**ファイル→ VI プロパティ**を選択して**カテゴリ**プルダウンメニューから**ウィンドウの外観**を選択し、**カスタマイズ**ボタンをクリックして**デフォル**

トランタイムショートカットメニュー使用可能チェックボックスをオフにします。デフォルトのランタイムショートカットメニューを無効にしても、カスタマイズしたショートカットメニューを含むことができます。

XControl の使用

複雑な制御器や表示器を設計、作成するには、XControl を使用します。XControl には、次のような機能があります。

- LabVIEW の標準の制御器と表示器を組み合わせて、それらの機能を統合された XControl を作成できます。
- VI で XControl を使用すると、XControl には制御器の働きが含まれるため VI のダイアグラムが単純化されます。
- XControl の動作、外観、機能、プロパティ、メソッドを定義するには、XControl ライブラリを使用します。
- XControl のショートカットメニューをカスタマイズして、XControl の作成に使用した新しいメニュー項目や LabVIEW の既存の標準制御器または表示器を追加できます。

XControl を作成するには、**プロジェクトエクスプローラ**ウィンドウで**マイ コンピュータ**を右クリックして、ショートカットメニューから**新規→ XControl** を選択します。

メモ XControl の作成および編集は、LabVIEW プロフェッショナル開発システムのみで実行できます。XControl がある VI は、すべての LabVIEW パッケージで実行できます。

XControl を使用するサンプルについては、`labview¥examples¥general¥xcontrols¥Dual Mode Thermometer` ディレクトリの Run Dual Mode Thermometer XControl プロジェクトを参照してください。

.NET コントロールを使用する

ActiveX コントロールと同じように、.NET コントロールをフロントパネルに使用できます。新規コントロールを作成を .NET コンテナで作成するか、後で使用するために .NET コントロールを制御器パレットに追加します。ブロックダイアグラムで .NET コントロールをプロパティノード、インボークノード、またはレジスタイベントバックノード二配線して、プロパティを設定、メソッドを呼び出し、またはコントロールのイベントを処理します。

新規 .NET コントロールを作成するには、.NET コンテナをフロントパネルに配置します。.NET コンテナを右クリックし、ショートカットメニューから **.NET コントロールを挿入**を選択して、**.NET コントロールを選択**ダイアログボックスを表示します。作成するを選択して**開く**ボタンをクリックします。

.NET コントロールを**制御器**パレットに追加するには、**ツール→ .NET & ActiveX → .NET コントロールをパレットに追加**を選択して、**.NET コントロールをパレットに追加**ダイアログボックスを表示します。追加する .NET コントロールを選択して、**OK** ボタンをクリックします。デフォルトで `labview¥menus¥Controls¥DotNet & ActiveX` ディレクトリにあるすべてのファイルが **.NET & ActiveX** パレットに自動的に表示されるため、LabVIEW はこのディレクトリに制御器を保存します。

リストボックス、ツリー制御器、および表

LabVIEW 8.0 には、リストボックス、表、デジタルデータ制御器、ツリー制御器に新しいプロパティとメソッドがいくつか追加されるなど、新しい機能が導入されています。これらの 4 種類の制御器は、多くの共通する動作と機能を備えています。

- 複数列リストボックスまたはツリー制御器を右クリックして、ショートカットメニューから**表示項目→行ヘッダ**を選択して行ヘッダを表示します。また、行ヘッダをプログラムで表示または非表示にするには、複数列リストボックスまたはツリー制御器の表示項目 : 行ヘッダ表示プロパティを使用することもできます。
- 垂直ラインや水平ラインを表示するには、複数列リストボックス、表、ツリー制御器のいずれかを右クリックして、ショートカットメニューから**表示項目→ 垂直ライン表示**または**表示項目→水平ライン表示**を選択します。また、垂直ラインをプログラムで表示または非表示にするには、複数列リストボックス、表、ツリー制御器の表示項目 : 垂直ライン表示プロパティを使用することもできます。水平ラインをプログラムで表示するには、単数列リストボックス、複数列リストボックス、表、ツリー制御器の表示項目 : 水平ライン表示プロパティを使用します。
- 色の設定ツールを使用して、表、リストボックス、ツリー制御器のすべてのセル、すべての行ヘッダ、すべての列ヘッダ、左上ヘッダセルに個別に色を設定できます。
- テキストがセルに入りきらない場合、VI を実行して操作ツールのカーソルをセル上に移動すると、全文が表示されたヒントラベルが表示されます。
- リストボックスとツリー制御器には、4 つの新しい項目記号があります。これらのチェックボックス記号の外観は、使用されているオペレーティングシステムによって異なります。記号をすべて表示するには、制御器内の項目を右クリックして、ショートカットメニューから**項目記号**を選択します。また、リストボックスにも、ツリー制御器と同様のカスタム記号の機能があります。
- <Shift-Enter> キーを押して、リストボックスとツリー制御器の現在の項目の下に項目を入力します。ツリー制御器では、このキーの組み合わせは、新規項目を現在の項目と同じ階層のレベルに配置し、新規項目の下のすべての項目は 1 つ下の行に移動されます。

- 実行時にユーザがセルを編集できるようにするには、リストボックスまたはツリー制御器を右クリックして、ショートカットメニューから**編集可能なセル**を選択します。また、セルの編集を許可プロパティを使用して、単数列リストボックス、複数列リストボックス、またはツリー制御器でユーザがプログラムでセルを編集できるようにします。セルを編集イベントを使用して、単数列リストボックス、複数列リストボックス、またはツリー制御器のセル内でユーザがテキストを変更するイベントを処理します。
- 項目の選択時に行全体をハイライトするには、リストボックスまたはツリー制御器を右クリックして、ショートカットメニューから**行全体をハイライト**を選択します。
- リストボックスとツリー制御器でのキーボード選択は、使用されているオペレーティングシステムのものと一致します。

メモ <PageUp>、<PageDown>、<Home>、<End> キーを制御器のプロパティダイアログボックスの**キー操作**ページでショートカットキーとして割り当てると、そのキーはリストボックスまたはツリー制御器で項目を選択するためには使用できません。

リストボックス

LabVIEW 8.0 では、リストボックスの以下の機能強化および変更が導入されました。

- 単数列リストボックスを右クリックしてショートカットメニューから**表示項目→水平スクロールバー**を選択し、水平スクロールバーを表示します。また、表示項目 : 水平スクロールバー表示プロパティを使用すると、プログラムで水平スクロールバーを表示または非表示にすることもできます。単数列リストボックスのショートカットメニューに表示されていた**表示項目→スクロールバー**項目は、**垂直スクロールバー**に変わりました。スクロールバー表示プロパティは、表示項目 : 垂直スクロールバー表示に名前変更されました。
- 単数列リストボックスを右クリックしてショートカットメニューから**表示項目→列ヘッダ**を選択して、列ヘッダを表示できます。また、表示項目 : 列ヘッダ表示プロパティを使用して、プログラムで列ヘッダを表示または非表示にすることもできます。
- 単数列リストボックスを右クリックし、ショートカットメニューから**前に行を挿入**を選択して、選択した行の前に 1 行追加することができます。
- 単数列リストボックスを右クリックし、ショートカットメニューから**行の高さを自動サイズ調整**を選択して、フォントの変更や複数行のテキストエントリに合わせて行の高さを調節できます。また、行の高さを自動サイズ調整プロパティを使用して行の高さをプログラムで変更することもできます。
- 単数列リストボックスを右クリックし、ショートカットメニューから**複数行入力**を選択して、セルでの複数行のエントリを可能にできま

す。複数ライン入力プロパティを使用して、プログラムでセルの複数行入力を可能にすることもできます。

- 複数列リストボックスを右クリックし、ショートカットメニューから**スムーズスクロール**を選択して、ユーザがリストボックスを水平方向に 1 ピクセルずつスクロールできるように設定できます。また、スムーズ水平スクロールバープロパティを使用して、ユーザがリストボックスを水平方向に 1 ピクセルずつスクロールできるようにプログラムすることもできます。

ツリー制御器

LabVIEW 8.0 では、ツリー制御器の以下の機能強化および変更が導入されました。

- 数値キーパッドの + キーと - キーを押してツリー制御器の項目を開く / 閉じることができます。また、ツリー制御器を右クリックし、ショートカットメニューから**すべての項目を開く**または**すべての項目を閉じる**を選択してツリー制御器の項目をプログラムで開く / 閉じることもできます。
- デフォルトでは、LabVIEW はツリー制御器の複数項目の対話式選択を同じ親を持つ項目に制限します。異なる親を持つ項目の選択を可能にするには、ツリー制御器を右クリックして、ショートカットメニューで**選択モード→兄弟にマルチ選択を制限**項目からチェックマークを外します。また、兄弟マルチ選択プロパティを使用して、プログラムで複数の項目の選択を兄弟に制限するか、異なる親を持つ項目を選択できます

メモ 複数の非兄弟項目が既に選択されている場合に、ツリー制御器で**兄弟にマルチ選択を制限**ショートカットメニュー項目を有効にすると、後続の選択の動作がわかりにくくなることがあります。選択を有効な選択内容にリセットすると、この選択動作は通常の状態に戻ります。

- ツリー制御器を右クリックし、ショートカットメニューの**展開 / 収縮記号タイプ→インデントレベル 0 で表示**項目のチェックマークを外して、階層の一番上のレベルにある項目の展開 / 収縮記号を非表示にします。インデントレベル 0 の項目は、ツリー制御器の左端に揃えられます。このオプションのとなりにあるチェックマークを外しても、項目をダブルクリックすると、インデントレベル 0 で項目を展開または縮小することができます。このダブルクリック機能を無効にするには、項目を閉じる？イベントの**破棄？**パラメータに TRUE 定数を配線します。また、展開 / 収縮記号 : インデントレベル 0 で表示プロパティを使用して、インデントレベル 0 の展開 / 収縮記号の表示と非表示をプログラムで切り替えられます。

表

表の水平スクロールバーが非表示の場合、ランタイムに <Shift- 右矢印キー > を押しても右方向へスクロールできません。垂直スクロールバー

が非表示の場合、ランタイムに <Shift- 下矢印キー > を押しても下方向へスクロールできません。上方向または左方向へは、スクロールバーが表示されていなくてもキーボードのキーを押してスクロールできます。

文字列およびツリー制御器でドラッグアンドドロップする

文字列制御器とツリー制御器は、LabVIEW 8.0 で標準のドラッグアンドドロップをサポートします。文字列制御器からドラッグするには、操作ツールを使用してテキストを選択し、別の文字列制御器やツリー制御器などのテキストを受け入れる制御器に対してその選択をドラッグします。ツリー制御器からドラッグするには、操作ツールを使用して項目を選択し、別のツリー制御器や文字列制御器などのツリーデータを受け入れる制御器に対してツリーからその項目をドラッグします。

ショートカットメニュー、イベント、プロパティ、メソッドを使用して、制御器でドラッグアンドドロップする項目を設定できます。「ドラッグドロップデータを取得（Get Drag Drop Data）」関数を使用して、ドラッグおよびドロップ操作を行う際に LabVIEW からドラッグデータを返すことができます。

文字列制御器および表示器で折り返しを無効にする

文字列制御器または表示器で改行を無効にするには、文字列を右クリックして、ショートカットメニュー項目の**折り返し有効化**の隣にあるチェックマークを外します。改行を無効にすると、文字列は代わりに改行で禁則処理を行います。折り返しを無効にするには、文字列が標準表示モードである必要があります。折り返しを無効にした後、文字列制御器または表示器を右クリックし、ショートカットメニューから**表示項目→水平スクロールバー**を選択すると、水平スクロールバーを表示できます。

また、折り返し有効化プロパティを使用して、単語での折り返しをプログラム的に無効にすることもできます。また、水平スクロールバー表示プロパティを使用して、プログラムで水平スクロールバーを表示することもできます。

パス制御器

パスプロパティダイアログボックスの**参照オプション**ページは、**参照オプション**ダイアログボックスに置換しました。**参照オプション**ページを表示するには、パス制御器を右クリックし、ショートカットメニューから**参照オプション**を選択します。このページには以下の機能強化が行われています。

- ボタンラベルを変更するには、**ボタンテキスト**テキストボックスを使用します。参照オプション : ボタンテキストプロパティを使用して、プログラムでボタンラベルを変更することもできます。

- **LLB をフォルダとして扱う**にチェックマークを付けると、ユーザは LLB にあるファイルを選択できるようになります。
- **選択モード**プルダウンメニューに置換したオプションは、**ファイルのみ**、**フォルダのみ**、**ファイルまたはフォルダ**、**既存のみ**、**新規のみ**、**新規または既存**です。

スクロールバー制御器

水平および垂直スクロールバー制御器を使用して、スクロール可能なデータを持つ制御器にカスタムスクロールバーを追加します。スクロールバーの値を変更するには、操作ツールを使用してスクロールボックスを新しい位置にクリックまたはドラッグする、増分矢印と減分矢印をクリックする、またはスクロールボックスと矢印の間のスペースをクリックします。

プログラムでページサイズ、最小および最大のデータ範囲制限を指定するには、Doc Min、Doc Max、増分、ページサイズ、最小値に強制、最大値に強制の各プロパティを使用できます。

配列スクロールバー

フロントパネルの制御器または表示器を右クリックし、ショートカットメニューから**表示項目→垂直スクロールバー**または**表示項目→水平スクロールバー**を選択すると、配列の垂直および水平スクロールバーを表示できます。配列は、1 次元より大きい場合にのみ垂直と水平の両方のスクロールバーを表示できます。1D 配列の場合、垂直スクロールバーを追加するには垂直方向に、水平スクロールバーを追加するには水平方向にサイズ変更します。

また、垂直スクロールバー表示プロパティおよび水平スクロールバー表示プロパティを使用して、プログラムで垂直および水平スクロールバーを表示することもできます。

その他の機能強化と変更点

LabVIEW 8.0 では、以下のその他のフロントパネルの機能強化と変更が行われました。

- グラフまたはチャートを右クリックし、ショートカットメニューから**上級→プロット凡例を無効にする**を選択して、ランタイムにプロット凡例の有効 / 無効を切り替えます。また、凡例 : 無効プロパティを使用して、波形チャート、波形グラフ、ミックスドシグナルグラフのプロット凡例の有効 / 無効をプログラムで切り替えることもできます。プロット凡例を無効にすると、ランタイムでグラフまたはチャートでのプロットの表示方法をカスタマイズできなくなります。
- LabVIEW 7.*x* 以前では、デジタル波形グラフの y スケールは、デフォルトで軸ラベルを表示します。LabVIEW 8.0 では、y スケールはデフォルトでは軸ラベルを表示しません。デフォルトの軸ラベルテキストは（表示するよう設定した場合）、**振幅**ではなく**デジタルプロット**と表示されるようになりました。

- リストボックス、複数列リストボックス、表、デジタルデータ、ツリー制御器は、垂直スクロールにマウスホイールをサポートします。
- カスタマイズする制御器または表示器の構成要素を右クリックし、ショートカットメニューから**上級→カスタマイズ**を選択して、制御器または表示器の構成要素のみを含む**制御器エディタ**ウィンドウを表示します。
- スライダまたはスライドまたは回転式制御器の指針を移動すると、制御器の値がヒントラベルに表示されます。ヒントラベルでは、制御器のデジタル表示と同じ形式と精度が使用されます。制御器に複数のデジタル表示がある場合、ヒントラベルでは現在アクティブな指針やスライダに対応するデジタル表示の形式と精度が使用されます。ヒントラベルを無効にするには、制御器を右クリックしてショートカットメニューから**プロパティ**を選択し、**外観**タブをクリックして**値ヒントラベルを表示**チェックボックスをオフにします。プログラムによって値のヒントラベルを無効にするには、値ヒントラベルを表示プロパティを使用します。
- 数値制御器および表示器は、c、d、da、h の各 SI 接頭辞を受け入れます。

LabVIEW MathScript（Windows）

LabVIEW MathScript ウィンドウは、数学的コマンドの編集と実行、数学的スクリプトの作成、数値表現とグラフィカル表現を使用した変数の表示を行います。このウィンドウを表示するには、**ツール→ MathScript ウィンドウ**を選択します。**LabVIEW MathScript ウィンドウ**は、呼び出したコマンドの出力の生成と履歴の保存、定義した変数のリスト表示、選択した変数の表示を行います。LabVIEW MathScript では、通常、MATLAB® 言語構文で書き込まれたスクリプトを実行できます。ただし、MathScript エンジンはスクリプトを実行しますが、MATLAB ソフトウェアがサポートする関数の一部をサポートしていません。

メモ MathScript は LabVIEW プロフェッショナル開発システムのみで利用できます。

LabVIEW MathScript ウィンドウまたは MathScript ノードで使用する関数およびスクリプトを書き込むことができます。

MathScript ノードは、ブロックダイアグラム上で MATLAB 言語構文に似た数式および表記を評価します。**LabVIEW MathScript ウィンドウ**で作成するスクリプトを保存したり、MathScript ノードでスクリプトをロードできます。また、MathScript ノードで作成したスクリプトを保存し、**LabVIEW MathScript ウィンドウ**でそのスクリプトをロードすることもできます。

行列データタイプ

行列データタイプを使用して、数学関連の問題のモデル化を容易で効率的に処理します。行列演算の既存の多態性 VI および関数は、行列データタ

イプを認識し、それに従って行列特有のアルゴリズムを実行します。複素行列の演算に使用する数学 VI および関数は、入力として行列を受け入れ、結果として行列を生成します。

メモ LabVIEW 7.*x* 以前では、「行列指数（Matrix Exp）」、「行列対数（Matrix Logarithm）」、「行列乗（Matrix Power）」、「行列平方根（Matrix Square Root）」VI は、LabVIEW 開発システムおよび LabVIEW プロフェッショナル開発システムでのみ使用できます。LabVIEW 8.0 では、これらの VI を LabVIEW ベースパッケージでも使用できます。

行列データタイプは行列演算（特に一部の線形代数の演算）の実数または複素数のスカラデータの行または列を保存するため、2 次元配列の代わりに行列データタイプを使用して行列データを表します。数学演算を実行する数学 VI および関数は、行列データタイプを受け入れて行列の結果を返します。これは、データフローで後続の多態性 VI および関数を有効にして、数学特有の演算を実行します。数学 VI または関数が数学演算を実行しない状態で行列データタイプを受け入れると、この VI または関数は自動的に行列データタイプを 2 次元配列に変換します。2 次元配列をデフォルトで行列演算を実行する VI または関数に配線すると、2 次元配列のデータタイプによって、VI または関数は自動的に 2 次元配列を実数または複素数の行列に変換します。

ほとんどの数値関数は行列データタイプおよび行列演算をサポートしています。たとえば、「積（Multiply）」関数を使用すると、他の行列または他の数値で行列を乗算できます。基本的な数値データタイプおよび複素線形代数関数を組み合わせて、正確な行列演算を実行する数値アルゴリズムを作成します。

LabVIEW 7.*x* 以前では、行列で数学演算を実行するには、線形代数 VI を使用する必要があります。2 次元配列は行列データを保存できますが、LabVIEW では線形代数関数などの一部の主要な数値関数について、行列制御器で行列データを処理する方法とは異なる方法で配列を処理します。LabVIEW 7.*x* 以前では 2 次元配列と行列を区別することができないため、数学関数を実行する前に配列の次元サイズと配列要素を確認し、行列データが行列演算に有効であることを確認する必要があります。行列データタイプを使用する場合は、数値関数を実行する前に配列の次元サイズを確認して配列要素を解析する必要はありません。2×3 行列と 4×4 行列など、配線する 2 つの行列の次元が異なる場合、VI または関数は空の行列を返します。

メモ VI または関数がデータを行列または 2 次元配列（またはその逆）に変換すると、強制ドットが VI または関数に表示されます。LabVIEW では 2 次元配列を保存する方法と同じように行列を保存するため、この種のデータ変換は実行には影響しません。

ブロックダイアグラムでは、行列データタイプは、異なる配線パターンを持つ実数 2 次元配列または複素 2 次元配列のデータタイプのように見えます。行列データタイプを入力として配線する場合、行列データタイプを受け入れる VI および関数は、自動的に行列特有の演算をサポートします。

行列データタイプを入力として以下の関数の 1 つに配線すると、サブ VI を含み、行列データタイプと動作する VI は、関数を置換します。

- 「等しい？（Equal?）」関数
- 「等しくない？（Not Equal?）」関数
- 「絶対値（Absolute Value）」関数
- 「和（Add）」関数
- 「積（Multiply）」関数
- 「平方根（Square Root）」関数
- 「差（Subtract）」関数
- 「指数関数（Exponential）」関数
- 「自然対数（Natural Logarithm）」関数
- 「X の Y 乗（Power Of X）」関数
- 「直交座標を複素数に変換（Re/Im To Complex）」関数
- 「複素数を直交座標に変換（Complex To Re/Im）」関数
- 「極座標を複素数に変換（Polar To Complex）」関数
- 「複素数を極座標に変換（Complex To Polar）」関数

結果として得られる VI は、同じアイコンで表示されますが、行列固有のアルゴリズムを持ちます。行列からこの入力の配線を解除しても、ノードは VI のままです。他のデータタイプを入力として配線すると、元の関数が復元されます。データタイプを関数に配線し、そのデータタイプが原因で基本的な数学演算が実行できなくなる場合は、関数は空の行列または **NaN** を返します。たとえば、2×3 の次元を持つ行列を「積（Multiply）」関数の入力に配線し、4×4 行列を他の入力に配線すると、関数は空の行列を返します。

メモ LabVIEW 7.*x* から LabVIEW 8.0 に VI をロードするとき、その VI が行列データタイプを使用できる関数に接続されている数学 VI を含む場合、LabVIEW はタイプ定義を無効にし、LabVIEW 7.*x* と同じ動作を行います。赤い 7.*x* グリフが関数に追加され、この関数が LabVIEW 7.*x* と同じ動作をすることを示します。

LabVIEW 7.0 以前では、1 次元配列のみがメモリ内で整列されます。これは、配列演算のパフォーマンスを向上させます。LabVIEW 7.1 以降では、メモリ内で 1 次元および 2 次元配列が整列されます。これは、線形代数の演算および行列データタイプに関連する演算のパフォーマンスを向上させます。

オプションダイアログボックスの機能強化

ツール→オプションを選択して**オプション**ダイアログボックスを表示します。**オプション**ダイアログボックスの左側にある**カテゴリ**リストから選択すると、ダイアログボックスでページを参照できます。**オプション**ダイアログボックスには以下の機能強化が行われています。

- **その他**ページは**環境**ページに名前変更されました。**最大取り消し回数**オプションは**ブロックダイアグラム**ページから**環境**ページへ移動しました。
- **VI サーバ :TCP/IP アクセス**ページは、**VI サーバ : マシンアクセス**ページに名前変更され、オプションも名前変更されました。
- **VI サーバ : ユーザアクセス**ページは、VI サーバを介しての VI に対するユーザアクセスを制御します。
- **メニューショートカット**ページでは、LabVIEW メニュー項目のキーボードショートカットを設定します。
- LaVIEW プロフェッショナル開発システムのみで使用可能な**ソース管理**ページでは、他社のソース管理プロバイダのソース管理を構成したり、LabVIEW のソース管理オプションを設定できます。
- **セキュリティ**ページは、LabVIEW のセキュリティオプションを設定します。
- **シェア変数エンジン**ページでは、シェア変数の時間の同期を設定します。
- **フロントパネル**および**ブロックダイアグラム**ページの**制御器でプリセット位置にスナップされているラベル / 端子でプリセット位置にスナップされているラベル**、**デフォルトでロックされているラベル**、**デフォルトラベル位置**オプションは、新規オブジェクトのラベルやキャプションのスナップ、ロック、位置決めを行います。
- ツリー制御器でパレット項目を表示するには、**制御器 / 関数パレット**ページの**形式**プルダウンメニューにある**ツリー**オプションを選択します。**パレット項目をソート**チェックボックスをオンにして、**形式**プルダウンメニューで**テキスト**または**ツリー**を選択した場合に、アルファベット順にパレットの項目をソートします。
- **VI サーバ : 構成**ページにある**サービス名**は、VI サーバが要求をリスンする TCP/IP サービス名を設定します。

別名で保存ダイアログボックスおよび旧バージョン用に保存ダイアログボックス

別名で保存ダイアログボックスを表示するには、以前に保存した VI で**ファイル→別名で保存**を選択します。**旧バージョン用に保存**ダイアログボックスを表示するには、**ファイル→旧バージョン用に保存**を選択します。

オプション付き保存ダイアログボックスは削除されました。**別名で保存**ダイアログボックスまたは**旧バージョン用に保存**ダイアログボックスを代用してください。**別名で保存**ダイアログボックスから**ソース配布プロパティ**ダイアログボックスにアクセスし、ソース配布を作成して指定した VI の

設定を構成してパスワードを追加し、ブロックダイアグラムを削除するか、または他の設定を適用することもできます。

ファイルサイズの向上

LabVIEW 8.0 で旧バージョンの LabVIEW から VI を保存すると、ファイルサイズが大幅に減少します。LabVIEW 8.0 で保存する VI のファイルサイズは、LabVIEW 7.1 で同じ VI を保存するファイルサイズよりも約 55% 少なくなります。LabVIEW 8.0 での LLB のファイルサイズは、LabVIEW 7.1 で同じ LLB を保存するファイルサイズよりも約 20% 少なくなります。

複数のバージョンの LabVIEW で共有ライブラリを使用する

LabVIEW 7.*x* 以前では、あるバージョンの LabVIEW で共有ライブラリ（DLL）を作成し、別のバージョンの LabVIEW でその共有ライブラリを使用すると、異なるバージョンの LabVIEW ランタイムエンジンが共有ライブラリを呼び出すため、エラーが返される可能性があります。LabVIEW 8.0 では、`labview.lib` の代わりに `labviewv.lib` を使用して共有ライブラリを作成すると、複数バージョンの LabVIEW やランタイムエンジンで共有ライブラリを使用できます。

ウィンドウランタイム位置を設定する

ランタイムフロントパネルウィンドウの位置とサイズをカスタマイズするには、**ファイル→ VI プロパティ**を選択し、**カテゴリ**プルダウンメニューから**ウィンドウランタイム位置**を選択します。また、フロントパネル : ランタイム位置メソッドを使用して、プログラムでランタイムフロントパネルウィンドウの位置とサイズをカスタマイズすることもできます。

旧バージョンの **VI プロパティ**ダイアログボックスの**ウィンドウサイズ**ページにある**フロントパネルを画面全体の幅と高さに合わせる**チェックボックスの代わりに、**位置**プルダウンメニューの**最大化**を使用します。

同様に、LabVIEW の旧バージョンの**ウィンドウの外観をカスタマイズ**ダイアログボックスにある**画面の中心に表示する**チェックボックスの代わりに、**位置**プルダウンメニューの**中央揃え**を使用します。

LabVIEW の旧バージョンでは、VI で**画面の中心に表示する**または**フロントパネルを画面全体の幅と高さに合わせる**を設定すると、その VI は停止しても編集モードの位置に戻りませんでした。VI の画面の中央位置とサイズは変わりませんでした。LabVIEW 8.0 では、VI は停止すると編集モードの位置に戻ります。

ランタイム言語の環境設定を行う

以下の方法でスタンドアロンアプリケーションの言語の環境設定ができます。最初にリストされているメソッドは、後にリストされているメソッドよりも優先度が高くなります。

- `-lang` *言語*コマンドラインオプション付きのアプリケーションを実行します。ここで、*言語*は使用したい言語の英語名です。
- アプリケーションの `.ini` ファイルに、`AppLanguage=` *言語*ラインを追加します。ここで、*言語*は使用したい言語の英語名です。
- **(Windows)** コマンドラインまたはアプリケーションの `.ini` ファイルで使用する言語を指定しない場合、アプリケーションはオペレーティングシステムと同じ言語を使用します。
- アプリケーションがオペレーティングシステムの言語をサポートしていない場合、アプリケーションはアプリケーションのデフォルトの言語を使用します。
- 上記の言語の環境設定のいずれも指定ない場合、アプリケーションはアルファベット順でサポートされている使用可能な最初の言語を使用します。アプリケーションでサポートされている言語を全く使用できない場合、LabVIEW でアプリケーションをロードすることはできません。

VI サーバの機能強化

VI サーバは LabVIEW 7.*x* 以前よりも詳しいエラーレポートを提供します。たとえば、「VI リファレンスを開く（Open VI Reference）」関数を使用したときにエラーが発生した場合、エラーレポートにエラーの原因となっている VI が表示されます。フロントパネルがない VI をロードすると、LabVIEW がフロントパネルをロードできないことを通知するエラーが発生します。またエラーレポートには、エラーの原因となっている特定のプロパティとメソッドの名前が表示されます。

VI サーバリファレンスを使用する

制御器リファレンス定数は VI サーバリファレンスに名前変更されました。ブロックダイアグラムにこのリファレンスを配置して、現在の VI、アプリケーション、または VI の制御器や表示器へリファレンスをリンクできます。このリファレンスを使用して、関連する VI、アプリケーション、あるいは制御器または表示器のプロパティおよびメソッドにアクセスすることができます。

操作ツールを使用してブロックダイアグラムで VI サーバリファレンスをクリックし、**このアプリケーション**または**この VI** を選択して現在のアプリケーションまたは VI にリファレンスをリンクします。また、VI サーバリファレンスを右クリックして、ショートカットメニューから**リンク先→このアプリケーション**または**リンク先→この VI** を選択します。VI 内で制御器または表示器にリファレンスをリンクするには、同じ方法でリファレンスをクリックまたは右クリックし、ショートカットメニューからリンクする制御器または表示器を選択します。

LabVIEW 標準の EXE ファイルは、多態性 VI に対して VI サーバリファレンスを開けません。EXE ファイルが作成されると、多態性 VI は削除され、

代わりに EXE ファイルに多態性インスタンスが作成されます。多態性インスタンスで VI サーバの操作を直接実行します。

VI サーバの NI セキュリティ

VI サーバには、クライアントが VI サーバ接続をできるようするなど、より多くの制御を行う NI セキュリティが含まれます。LabVIEW 7.*x* 以前では、VI サーバはクライアントのマシンアドレスを基準に接続の可否を判断します。LabVIEW 8.0 では、NI セキュリティの統合により、VI サーバはユーザおよびグループを基準に接続の可否を判断します。

デフォルトでは、VI サーバの NI セキュリティは無効です。NI セキュリティが無効の場合、マシンアドレスのみを使用して接続の可否を判断します。N セキュリティを有効にするには、ユーザまたはグループがアクセスできるか否かを指定する必要があります。**ツール→オプション**を選択して**オプション**ダイアログボックスを表示し、**VI サーバ : ユーザアクセス**を**カテゴリ**リストから選択して、**VI サーバ : ユーザアクセス**ページを表示します。この、VI サーバ : ユーザアクセスページは、アクセスの可否の判断に使用します。空のユーザ制御アクセスリストは、NI セキュリティが VI サーバに対して無効であることを意味します。

LabVIEW 8.0 では、クライアントは LabVIEW 7.*x* 以前と同じ方法で VI サーバ接続を開きます。サーバに VI サーバのユーザまたはグループ制限がある場合、クライアントは接続を行う前にログインする必要があることがあります。**NI セキュリティログイン**ダイアログボックスを使用するか、NI セキュリティ : ログインメソッドを使用してプログラムでログインします。ユーザが接続を許可されていない場合、サーバは接続を閉じてクライアントへ `kLVE_NISecurity_AuthenticationFailed` (1379) エラーを返します。

VI サーバ接続が開いている間にクライアントが変更すると、クライアントは自動的にサーバへメッセージを送信し、ユーザが変更されたことを通知します。サーバが新しいクライアントに接続を許可しない場合、サーバは VI サーバ接続を閉じて `kLVE_NISecurity_AuthenticationFailed` (1379) エラーを返します。

クラスブラウザウィンドウ

クラスブラウザウィンドウで、使用できるオブジェクトライブラリを選択して、選択したオブジェクトライブラリ内でクラス、プロパティ、メソッドを表示できます。選択したプロパティまたはメソッドとともにプロパティノードまたはインボーグノードを作成することができます。また、**クラスブラウザ**ウィンドウを使用して、クラス、プロパティ、メソッドを検索し、ドット付きのプロパティおよびメソッドを作成することもできます。**表示→クラスブラウザ**を選択して、**クラスブラウザ**ウィンドウを表示します。

プロパティノードでプロパティを右クリックしてショートカットメニューから**プロパティを選択**を選択するか、インボーグノードを右クリックしてショートカットメニューから**メソッドを選択**を選択すると、既存のプロパティまたはインボーグノードを編集できます。

.NET アセンブリを使用する

LabVIEW は、自動的に、インストールされている .NET 共通言語ランタイム（CLR）で最新バージョンをロードします。旧バージョンの .NET を使用して .NET アプリケーションを開発する場合は、最新バージョンの .NET をインストールしてそのアプリケーションを再配布します。旧バージョンの .NET を使用しているユーザはアプリケーションを実行することができません。特定の CLR バージョンをロードするには、構成ファイルを使用します。これにより、より最新バージョンの .NET をインストールし、.NET の開発を継続できます。また、他のユーザも旧バージョンの .NET でアプリケーションの実行を継続できます。

LabVIEW 7.*x* では、**ツール→上級→ .NET アセンブリリファレンス**を選択し、プライベート .NET アセンブリファイルを手動で登録しないと、それらのアセンブリを参照する VI を使用できません。LabVIEW 8.0 では、プライベート .NET アセンブリファイルを登録する必要がないので、**ツール→上級→ .NET アセンブリリファレンス**メニュー項目がありません。代わりに、「コンストラクタノード（Constructor Node）」を配置すると、指定するプライベート .NET アセンブリへのリファレンスを作成できます。LabVEW は、アセンブリから .NET オブジェクトを含む VI への相対パスを格納します。アセンブリを GAC（Global Assembly Cache）、LabVIEW のプロジェクトファイルと同じディレクトリ、または LabVIEW プロジェクトのサブディレクトリに格納すると、アセンブリをより効率的にロードします。

.NET CLR は、構成設定を使用してロードするアセンブリのバージョンを判断します。これらの設定は、リクエストしたアセンブリバージョンよりも優先されます。たとえば、システム管理者が、ユーザのコンピュータでバージョン 1.0.0.0 ではなく 1.0.0.1 がロードされるように設定したとします。ユーザがバージョン 1.0.0.0 をロードしようとすると、NET CLR によってアセンブリのバージョンが 1.0.0.1 に変更されます。さらに、バージョンが変更されたことを通知する LabVIEW のダイアログボックスを起動します。

プライベートアセンブリを使用する VI を他のフォルダまたはコンピュータに移動する場合、アセンブリを VI に対して同じ場所に維持する必要があります。LabVIEW が、GAC、プロジェクトを含むディレクトリ、または VI に格納された相対パスでアセンブリファイルを検索できない場合、LabVIEW はアセンブリファイルを検索することを要求するプロンプトを表示します。

プライベートアセンブリを共有ライブラリまたはスタンドアロンアプリケーションに VI を作成する場合、LabVIEW は関連プライベート .NET ア

センブリファイルを、ライブラリまたはアプリケーションと同じディレクトリの data サブディレクトリにコピーします。

LabVIEW 8.0 では、 .config ファイルが保存されたプロジェクト、共有ライブラリ、またはスタンドアロンアプリケーションに適用されます。

計測器ドライバファインダ

計測器ドライバファインダ（**ツール→計測→計測ドライバを検索**で選択可能）は、LabVIEW の開発環境を保持したまま LabVIEW プラグアンドプレイ計測器ドライバの簡単な検索およびインストール方法をユーザに提供します。また、計測器ドライバファインダは instr.lib で計測器ドライバを表示し、NI-VISA を使用して接続されている計測器を検索します。

メモ　計測器ドライバファインダは、Mac OS では使用できません。

計測器ドライバ VI ウィザード

計測器ドライバ VI ウィザードは、既存の計測器ドライバプロジェクトライブラリを拡張します。計測器ドライバ VI ウィザードを起動するには、計測器ドライバプロジェクトを作成または開いて、**プロジェクトエクスプローラ**ウィンドウのプロジェクトライブラリファイルを右クリックし、ショートカットメニューから**新規→計測器ドライバ VI** を選択します。このウィザードを使用して、計測器ドライバ VI を作成し、VI を計測器ドライバプロジェクトライブラリに挿入します。また、このウィザードを使用してフロントパネルの制御器に対応する計測器コマンド文字列のパラメータを構成し、計測器からの応答を解析する方法を定義します。

計測器ドライバプロジェクトウィザード

計測器ドライバプロジェクトウィザード（**ツール→計測→計測器ドライバプロジェクトを作成**で選択可能）は、GPIB、RS-232、イーサネット、USB 計測器などのプログラム可能な計測器を制御する VI を含む、計測器ドライバプロジェクトを自動化します。このウィザードは、**関数**パレットでドライバの VI を表示する計測器ドライバ VI およびパレットメニューを作成します。クラステンプレート、汎用のメッセージベーステンプレート、または類似した計測器の既存ドライバから、計測器ドライバを作成できます。

使用可能なクラステンプレートは、DC 電源装置、デジタルマルチメータ、関数発生器、およびオシロスコープです。任意のメッセージベースの計測器には、General Purpose テンプレートを使用できます。計測器が既存の計測器と類似している場合、既存の計測器ドライバを基準に新しいドライバを開始できます。

既存の計測器ドライバを基準に新しい計測器ドライバを作成する場合、計測器ドライバプロジェクトウィザードを起動する前に、既存の計測器ドライバプロジェクトを labview¥instr.lib ディレクトリへインストール

する必要があります。既存の計測器ドライバはプロジェクトスタイルドライバである必要があります。NI 計測器ドライバファインダを使用して、プロジェクトスタイルドライバをダウンロードおよびインストールできます。

計測器ドライバプロジェクトウィザードは、LabVIEW プラグアンドプレイ計測器ドライバの標準形式を使用する LabVIEW プロジェクトを使用します。プロジェクトは、計測器ドライバガイドラインに適合する VI およびパレットメニューファイルを含みます。ウィザードがプロジェクトを作成した後、計測器とともに動作するように各 VI をカスタマイズする必要があります。計測器ドライバ VI ウィザードを使用して、計測器ドライバ VI を作成して、VI を計測器ドライバプロジェクトライブラリに挿入できます。

再入可能 VI を使用およびデバッグする

LabVIEW を使用して、再入可能 VI で以下のタスクを実行します。

- 再入可能 VI をデバックできます。再入可能 VI のブロックダイアグラムのインスタンスをデバックするために表示できますが、ブロックダイアグラムインスタンスを編集できません。ブロックダイアグラム内でブレークポイントを設定して、プローブを使用し、実行のハイライト、実行のシングルステップを有効にすることができます。再入可能 VI のデバッグを許可するには、**ファイル→ VI プロパティ**を選択して **VI プロパティ**ダイアログボックスを表示し、プルダウンメニューから **実行**を選択して、**デバッギングを許可**チェックボックスをオンにします。
- **VI プロパティ**ダイアログボックスを使用して再入可能 VI を設定して、実行中にフロントパネルを開き、再入可能 VI の実行後にオプションで再び閉じます。
- VI サーバを使用して、プログラムで再入可能 VI でランタイム時にフロントパネルの制御器および表示器を制御できますが、ランタイム時に制御器と表示器を編集することはできません。リファレンスが再入可能インスタンスを指す場合に、制御器の値 : 取得［平坦化］、制御器の値 : 設定 (平坦化)、および VI を実行などの多く VI サーバのメソッドおよびプロパティを使用して、VI の再入可能インスタンスと通信できます。
- VI サーバを使用して、再入可能 VI のフロントパネルのコピーをランタイム時に作成することもできます。再入可能 VI のフロントパネルをコピーするには、「VI リファレンスを開く（Open VI Reference）」関数を使用して VI サーバリファレンスを開きます。タイプ指定 VI リファレンスを**タイプ識別子**入力に配線するか、**再入実行として開く**を**オプション**入力に配線します。リファレンスを開くと、LabVIEW は VI のコピーを作成します。VI サーバまたは **VI プロパティ**ダイアログボックスを使用して、再入可能 VI のフロントパネルを開くこともできます。
- また、イベントストラクチャケースを構成して、再入可能 VI のフロントパネルオブジェクトのイベントを処理します。

- LabVIEW の標準ウェブサーバに接続することによって、LabVIEW 内部から、またはウェブブラウザから再入可能 VI のフロントパネルをリモートで表示したり制御したりできます。ウェブサーバを使用する場合、複数のユーザが VI のコントローラインタフェースにアクセスできます。各接続に、それぞれインスタンスがあります。
- XControl の外観 VI は、再入可能 VI である必要があります。
- 再入可能 VI のフロントパネルは、サブパネルにすることもできます。
- 旧バージョンの LabVIEW では、再入可能 VI の各インスタンスについて 1 つのフロントパネルのみが表示されたため、制御器および表示器の値が常に正しくはありません。LabVIEW 8.0 では、再入可能 VI の各インスタンスがそれぞれフロントパネルを持つため、制御器および表示器の正確な値を毎回確認できます。

ブロックダイアグラムから再入可能サブ VI を開くと、LabVIEW はソース VI の代わりに VI のクローンを開きます。VI のタイトルバーに（クローン）が含まれ、ソース VI のクローンであることが示されます。クローン値は編集できません。

VI 実行後にプローブを作成する

VI は、ワイヤ値が保持されるように構成できます。ブロックダイアグラムでプローブを作成すると、そのプローブによって VI 実行時にワイヤ上を通過したデータが表示されます。**ワイヤ値を保持**ボタンをクリックすると、実行フローの各ポイントでワイヤ値が保存されるので、ワイヤにプローブを配置すると、ワイヤを介して渡された最新のデータ値を即座に取得できます。プローブによってどのワイヤのデータを取得する場合でも、少なくとも 1 回は VI を正常に実行させる必要があります。VI 実行後のプローブの使用は、ブロックダイアグラムが複雑な VI のデバッグを行う場合に便利です。

ブロックダイアグラムのツールバーで**ワイヤ値を保持**ボタンをクリックすると、プローブで使用するワイヤ値を取得します。

ワイヤ値を保持しないボタンを任意にクリックすると、各実行でワイヤ値の保存を回避します。

メモ　**ワイヤ値を保持しない**オプションを選択すると、わずかに必要なメモリ容量が低減され、パフォーマンスが向上します。

複数ターゲットで VI をプロファイルする

パフォーマンスおよびメモリをプロファイルウィンドウを使用して、異なるターゲットにある VI を同時にプロファイルできます。**パフォーマンスおよびメモリをプロファイル**ウィンドウには、LabVIEW プロジェクトライブラリ（`*.lvlib`）およびターゲットを表示する新規の**プロジェクトライブラリ**列が含まれます。この列を使用して、同じ名前で異なるプロジェクトライブラリに含まれるか、または異なるターゲット上にロードされた

VI を差別化できます。**パフォーマンスおよびメモリをプロファイル**ウィンドウには新規の**アプリケーションインスタンスを選択**ボタンがあり、このボタンで**アプリケーションインスタンスを選択**ダイアログボックスを起動します。このダイアログボックスを使用して、**パフォーマンスおよびメモリをプロファイル**ウィンドウでデータを表示しているターゲットを選択します。**アプリケーションインスタンスを選択**ボタンの隣は、**パフォーマンスおよびメモリをプロファイル**ウィンドウおよび LabVIEW で表示に使用する色でアクセスする各アプリケーションインスタンスを示す凡例です。**パフォーマンスおよびメモリをプロファイル**ウィンドウを起動するには、以前は**ツール→上級→プロファイル VI** を選択していましたが、**ツール→プロファイル→パフォーマンスおよびメモリ**を選択します。

メモ LabVIEW プロジェクト（`*.lvproj`）は、VI やビルド仕様などの集合です。プロジェクトに関連するすべてのファイルが**プロジェクトエクスプローラ**ウィンドウに表示されます。一度に 1 つのプロジェクトのみプロファイルできます。すでに**パフォーマンスおよびメモリをプロファイル**ウィンドウを開いている場合は、**ツール→プロファイル→パフォーマンスおよびメモリ**オプションを無効にします。

64 ビット整数データタイプ

大きい整数を使用、表示、保存するには、64 ビット符号付き / 符号なし整数データタイプを使用します。

多態性の数値入力が使用されるすべての関数では、64 ビット整数データタイプがサポートされます。「循環（Rotate）」関数、「ビットシフト（Logical Shift）」関数、「2 の累乗でスケール（Scale By Power Of 2）」関数の循環量、シフト量、累乗の 64 ビット入力は、それぞれ 32 ビット整数に強制されます。関数が 64 ビット整数出力を返すようにしたい場合は、「10 進数文字列を数値に変換（Decimal String To Number）」、「16 進数文字列を数値に変換（Hexadecimal String To Number）」、「8 進数文字列を数値に変換（Octal String To Number）」、「小数 / 指数文字列を数値に変換（Fract/Exp String To Number）」関数の**文字列**入力に 64 ビット整数を配線する必要があります。

64 ビットの符号付きまたは符号なし整数を変換するには、それぞれ「4 倍長整数に変換（To Quad Integer）」関数と「符号なし 4 倍長整数に変換（To Unsigned Quad Integer）」関数を使用してください。

NI Spy

NI Spy は、アプリケーションの実行中に計測器 I/O 通信を監視するツールです。NI Spy を使用して、LabVIEW VI の実行中に計測器 I/O の呼び出しと結果をキャプチャできます。呼び出しと結果をキャプチャすると、計測器通信での問題をデバッグする上で役立ちます。**(Windows)** NI Spy を起動するには**スタート→プログラム→ National Instruments → NI Spy** を選択します。**(Mac OS)** NI Spy を起動するには **Applications → National Instruments → NI Spy** を選択して NI Spy をダブルクリックしま

す。**(Linux)** NI Spy を起動するには、コマンドラインで `nispy` と入力します。

検索 / 置換機能

In LabVIEW 7.*x* 以前では、ブロックダイアグラムのオブジェクトの 1 つを右クリックし、ショートカットメニューから**置換**を選択して他のオブジェクトに置換することができます。LabVIEW 8.0 では、複数のオブジェクトやテキストオブジェクトを一度に置換できます。**検索**ダイアログボックスを使用することで、VI、オブジェクト、およびテキストに基づいて VI を検索できます。その後、**検索結果**ウィンドウですべての項目または選択した項目を別のオブジェクトやテキストに置換できます。

カスタマイズ可能なキーボードショートカット

オプションダイアログボックスの**メニューショートカット**ページでは、VI メニュー項目のキーボードショートカットを設定します。<Ctrl> キーを修正キーとして使用せずに、<F1> ～ <F24> ファンクションキーを使用することができます。

印刷ダイアログボックスの機能強化

印刷ダイアログボックスには以下の機能強化が行われています。

- **VI を選択**ページは、現在の VI または複数の VI を印刷するかどうかを選択します。
- **内容を印刷**ページの**フロントパネルすべて**、**フロントパネル可視部**、**VI ドキュメント**オプションで、印刷内容を選択します。**パネルを使用**、**サブ VI として使用**、**ドキュメントのすべて**、**カスタム**の各ドキュメントオプションは、**内容を印刷**ページから **VI ドキュメント**オプションを選択してアクセスする **VI ドキュメント**ページへ移動しました。
- **カスタム詳細**ページは **VI ドキュメント**ページに名前が変更されました。**VI ドキュメントスタイル**オプションは **VI ドキュメント**ページでコンポーネントを設定します。**アイコンとコネクタペーン**および**説明**オプションは、**アイコンと説明**オプションおよびその構成要素に変更されました。制御器および表示器のラベルやキャプションを印刷するには、**制御器**オプションの下にある**ラベル**、**キャプション**、**キャプション (ラベル)** を使用します。
- このダイアログボックスでは、現在のアプリケーションインスタンス内にある VI のドキュメントを印刷できます。複数のアプリケーションインスタンス内にある VI のドキュメントを印刷する場合は、それぞれのアプリケーションインスタンスで印刷操作を繰り返さなければなりません。

VI アイコンを編集する

`labview¥resource¥plugins` ディレクトリにある lv_icon VI テンプレートを使用して、**アイコンエディタ**ダイアログボックスの外で VI アイコンを編集できる VI を作成します。

ドキュメントの機能強化と変更点

ドキュメントについては、以下のとおり機能強化と変更が行われました。

ドキュメントの再編成

『LabVIEW ヘルプ』に、LabVIEW 7.*x* 以前のマニュアルとアプリケーションノートの概念トピックが追加されました。したがって、PDF ライブラリや『LabVIEW Measurements Manual』、『Using External Code in LabVIEW』、『LabVIEW Development Guidelines』、『LabVIEW Analysis Concepts』、『アプリケーションノート』の PDF バージョンは添付されていません。

LabVIEW 7.*x* と LabVIEW 8.0 のドキュメントの比較は以下の表のとおりです。

表 5　LabVIEW 8.0 ドキュメントの再編成

ドキュメント	LabVIEW 7.*x*	LabVIEW 8.0
LabVIEW 入門	印刷版と PDF	印刷版と PDF
LabVIEW リリースノート	印刷版と PDF	印刷版と PDF
LabVIEW アップグレードノート	印刷版と PDF	印刷版と PDF
LabVIEW クイックリファレンスカード	印刷版と PDF	印刷版と PDF
LabVIEW ユーザマニュアル	印刷版と PDF	LabVIEW ヘルプ : **基本機能**（印刷版の一部）
LabVIEW Measurements Manual	印刷版と PDF	LabVIEW ヘルプ : **計測を実行する**、**計測器を制御する**
LabVIEW アプリケーションビルダ Readme	印刷版と PDF（*LabVIEW アプリケーションビルダ ユーザガイド*）	`labview¥readme` ディレクトリにあります。
LabVIEW Development Guidelines	印刷版と PDF	LabVIEW ヘルプ : **基本機能→開発ガイドライン**
LabVIEW Analysis Concepts	PDF	LabVIEW ヘルプ : **基本機能→信号処理および解析**

表 5　LabVIEW 8.0 ドキュメントの再編成（続き）

ドキュメント	LabVIEW 7.*x*	LabVIEW 8.0
Using External Code in LabVIEW	PDF	LabVIEW ヘルプ : **基本機能→テキストベースのプログラミング言語で書かれたコード呼び出し**
Integrating the Internet into Your Measurement System—DataSocket Technical Overview	PDF	—
LabVIEW とハイパースレッド	PDF	LabVIEW ヘルプ : **基本機能→LabVIEW とハイパースレッド**
LabVIEW カスタム制御器、表示器、およびタイプ定義を作成する	PDF	LabVIEW ヘルプ : **基本機能→フロントパネルを作成する→概要→カスタム制御器、表示器、およびタイプ定義を作成する**
LabVIEW データストレージ	PDF	LabVIEW ヘルプ : **LabVIEW でメモリにデータを保存する方法**
LabVIEW パフォーマンスとメモリ管理	PDF	LabVIEW ヘルプ : **基本機能→パフォーマンスおよびメモリを管理する→概要→ VI プロファイルウィンドウを使用して VI のパフォーマンスを監視する**
LabVIEW における多態性単位	PDF	LabVIEW ヘルプ : **基本機能→ VI およびサブ VI を作成する→概要→多態性単位を使用する**
LabVIEW VI の移植とローカライズ	PDF	LabVIEW ヘルプ : **基本機能→ VI の移植とローカライズ**
Using Apple Events and the PPC Toolbox to Communicate with LabVIEW Applications on the Macintosh	PDF	—
LabVIEW でパフォーマンスと信頼性強化のマルチスレッド VI を作成する	PDF	LabVIEW ヘルプ : **基本機能→マルチタスキング、マルチスレッディング、およびマルチプロセッシング**
TCP/IP および UDP で LabVIEW を使用する	PDF	LabVIEW ヘルプ : **TCP/IP および UDP で LabVIEW を使用する**

表 5　LabVIEW 8.0 ドキュメントの再編成（続き）

ドキュメント	LabVIEW 7.*x*	LabVIEW 8.0
ワイヤレスデバイスで LabVIEW を使用する	PDF	LabVIEW ヘルプ : **ワイヤレスデバイスで LabVIEW を使用する**
Using the Timed Loop to Write Multirate Applications in LabVIEW	PDF	LabVIEW ヘルプ : **基本機能→ループとストラクチャ→タイミングストラクチャ**

(Windows) 『LabVIEW ヘルプ』の**目次**タブでブックアイコンの内容全体を印刷するには、以下の手順を実行します。

1. ブックアイコンを右クリックします。
2. ショートカットメニューから**印刷**を選択し、**トピックの印刷**ダイアログボックスを表示します。
3. **選択された見出しおよびすべてのサブトピックを印刷**オプションを選択します。

メモ　**目次**タブで選択した単一のトピックを印刷する場合は、**選択されたトピックの印刷**を選択します。

4. **OK** ボタンをクリックします。

このヘルプファイルは PDF ドキュメントへのリンクを含む場合があります。PDF ドキュメントを印刷するには、Adobe Acrobat Viewer ツールバーにある印刷ボタンをクリックします。

『LabVIEW 基本機能』マニュアル、『LabVIEW リリースノート』、『LabVIEW アップグレードノート』、『LabVIEW 入門』マニュアル、『LabVIEW クリックリファレンスカード』の PDF 版を検索するには、`Adobe Acrobat Reader` を使用してください。LabVIEW に添付されている PDF ドキュメントを検索するには、以下の手順に従ってください。

1. **編集→検索**を選択して **PDF を検索**ウィンドウを表示します。
2. **検索する語句を指定してください。**テキストボックスに、検索する単語または語句を入力します。
3. **以下の場所にあるすべての PDF ドキュメント**ボタンをクリックして、ドロップダウンリストから `labview¥manuals` ディレクトリを選択します。
4. **検索**ボタンをクリックします。

単語や語句によるフォルダ内の PDF ドキュメントの検索の詳細については、Adobe Reader のヘルプを参照してください。

Readme ファイル（Windows のみ）

LabVIEW ファイルストラクチャの `readme` ディレクトリには、LabVIEW と、インストールされているモジュールやツールキットのの Readme ファイルが配置されています。Readme ファイルにアクセスするには、**スタート→すべてのプログラム→ National Instruments → LabVIEW → Readme** を選択します。

同期ボタン

『LabVIEW ヘルプ』では、**キーワード**タブ、**検索**タブ、**お気に入り**タブからトピックを表示した場合、**同期**ボタンをクリックすると目次内でそのトピックを特定できます。

メモ LabVIEW では、目次にダイアログボックスのリファレンスのトピックの一覧が表示されません。

サンプルにアクセスする

ブロックダイアグラムや配置したパレットで VI または関数を右クリックしてショートカットメニューから**サンプル**を選択し、VI または関数のサンプルへのリンクがあるヘルプトピックを開けられます。パレットの左上隅にある画鋲のアイコンをクリックして、パレットを配置します。

詳細ヘルプウィンドウ

詳細ヘルプウィンドウで、**ヘルプを表示するにはここをクリック**リンクと**詳細なヘルプ**ボタンはどちらも**詳細なヘルプ**に名前が変更されました。

NI サンプルファインダの機能強化

NI サンプルファインダの**参照**タブと**検索**タブの**必要条件**リストには、サンプルを使用するために必要な追加のハードウェアとソフトウェアが表示されます。NI サンプルファインダに**必要条件**リストを表示するには、**サンプルファインダを構成**ダイアログボックスの**一般**タブで、**必要条件リストの表示**チェックボックスをオンにします。

NI サンプルファインダから、`NI Developer Zone` のウェブサンプルに対するコメントや評価を表示、送信できます。NI サンプルファインダには、ウェブサンプルの関連ドキュメントへのリンクも表示されています。NI サンプルファインダに**リソース**リストを表示するには、**サンプルファインダを構成**ダイアログボックスの **Web** タブで、**リソースリストの表示**チェックボックスをオンにします。

NI サンプルファインダの**参照**タブを使用して、ハードウェアまたはソフトウェアで使用する特定のタスクに適用する NI Developer Zone のサンプルを参照できます。**(Windows および Linux)** ウェブサンプルを参照するには、**表示オプション**セクションで**タスク**オプションを選択し、**ni.com にあるサンプルを含む**チェックボックスをオンにします。表示す

るウェブサンプルの最大数は、**サンプルファインダを構成**ダイアログボックスの **Web** タブにある **Web クエリ結果の最大数**フィールドで指定できます。また、NI サンプルファインダの検索がタイムアウトになるまでの時間を**サンプルファインダを構成**ダイアログボックスの **Web** タブにある **Web クエリ時間制限（秒）**フィールドで指定することもできます。

タスクでサンプルを参照している場合は、**サンプルを開くにはダブルクリックしてください**リストの**インストールされていないツールキットとモジュール**フォルダに、現在インストールされていないツールキットとモジュールのサンプルが含まれます。**インストールされていないツールキットとモジュール**フォルダでサンプルを選択する場合は、**必要条件**リストに、必要なツールキットまたはモジュールに関する詳細情報のリンクが含まれます。

コンピュータにインストールされたサンプルをキーワードまたは説明に基づいて検索するには、NI サンプルファインダの**検索**タブを使用します。キーワードで検索するには、**検索**ボタンの横にあるプルダウンメニューで**キーワード**を選択します。説明で検索するには、**検索**ボタンの横にあるプルダウンメニューで**説明**を選択します。説明でサンプルを検索するには、**ni.com にあるサンプルを含む**チェックボックスをオフにする必要があります。説明でサンプルを検索した場合、検索したテキストは**情報**テキストボックスでハイライト表示されます。

VI ベースまたはプロジェクトベースの LabVIEW サンプルを検索または参照するには、NI サンプルファインダを使用します。プロジェクトベースのサンプルを開くと、LabVIEW プロジェクトファイル（`.lvproj`）が**プロジェクトエクスプローラ**ウィンドウ内に開きます。サンプルを開くには、**プロジェクトエクスプローラ**ウィンドウの**マイ コンピュータ**の下にある VI をダブルクリックします。VI ベースのサンプルの動作には変更はありません。

LabVIEW 7.*x* 以前では、NI サンプルファインダはインストールしている DAQ ボードのみを検索できます。LabVIEW 8.0 では、NI サンプルファインダでインストールしているハードウェアのほとんどを検索できます。**設定**ボタンをクリックして**サンプルファインダを構成**ダイアログボックスを表示すると、検索にインストールされていないハードウェアを追加できます。

LabVIEW 8.0 でのその他の機能および変更点

LabVIEW 8.0 では、その他に以下の機能強化および変更が行われています。

既存の VI および関数の変更点

LabVIEW 8.0 では、既存の VI および関数において以下の点が変更されました。

- 「VISA シリアルポート構成（VISA Configure Serial Port）」および「VISA シリアルブレーク（VISA Serial Break）」VI の**複製 VISA リ**

ソース名出力は、**VISA リソース名出力**に名前変更されました。「VISA USB 割り込みデータ取得（VISA Get USB Interrupt Data）」VI の**複製 VISA USB 割り込みイベント**出力は、**VISA USB 割り込みイベント出力**に名前変更されました。その他すべての VISA VI および関数の**複製 VISA リソース名**出力は、**VISA リソース名出力**に名前変更されました。

- 「VISA トリガアサート（VISA Assert Trigger）」関数の**プロトコル**入力には、6 (PXI: 予約) および 7 (PXI: 予約なし) の 2 つの新しい値があります。
- 「VISA 非同期ロック（VISA Lock Async）」には、**ロックタイプ**と**リクエストされたキー**、および新規出力の**アクセスキー**があります。これらのパラメータを使用して、現在のセッションの排他的なロックまたは共有ロックを要求します。
- 「VISA シリアルブレーク（VISA Serial Break）」VI の**継続時間**入力は、ブレークの長さをミリ秒単位で指定します。
- 「VISA イベント待機（VISA Wait on Event）」関数の**イベントリソースクラス**入力は、**イベントクラス**と名前が変更され、必須入力からオプション入力に変更されました。
- 「波形要素取得（Get Waveform Components）」関数は、デフォルトで **Y** データ配列を取得します。
- **数値**パレットで「複合演算（Compound Arithmetic）」関数を選択してドロップした場合、デフォルトのモードは「和」です。「複合演算」関数を**ブール**パレットで選択した場合、デフォルトのモードは「OR」です。
- 「バリアントへ変換（To Variant）」、「バリアントからデータに変換（Variant To Data）」、「バリアントから平坦化文字列に変換（Variant To Flattened String）」、「バリアント属性設定（Set Variant Attribute）」の各関数のパフォーマンスが向上しました。これらの関数を呼び出す VI のほとんどが、これまでより高速に実行されます。特に大きなデータセットを持つ VI で効果が顕著です。
- 「文字列の検索と置換（Search and Replace String）」関数の**文字列**パラメータが、**入力文字列**に名前変更されました。また、「文字列の検索と置換」関数には 5 つの新しいパラメータが追加されました。**大小文字の区別を無視？**パラメータは、文字列検索で大文字と小文字を区別するかどうかを指定します。**マルチライン？**パラメータは、^ および $ の記号をそれぞれ、行の最初と最後に一致させるか、ストリング全体の最初と最後に一致させるかどうかを指定します。**置換数**パラメータは、LabVIEW が **検索文字列**を置換した回数を返します。この関数は、**エラー入力**パラメータ、**エラー出力**パラメータとも呼ばれます。
- 「PNG ファイル読み取り（Read PNG File）」VI の**透明度しきい**入力は、生成する**画像データ**の**マスク**に 32 ビット PNG 画像の α 情報を取り込みます。LabVIEW は、**透明度しきい**以上の α 値を不透明とみなします。それ以外のすべての α 値は、完全に透明であるとみな

されます。**透明度しきい**の値は、0 ～ 255 の間でなければなりません。

- 「デジタルパターン検索（Search for Digital Pattern）」VI の**比較モード**入力は、X の値を検索時にどのように扱うかを指定します。この関数の DWDT デジタルパターン検索インスタンスの**モード**および**開始指標 / 秒**入力は、それぞれ**開始値形式**および**開始**に名前変更されました。**指標**モードは、**サンプル**に名前変更されました。
- 「Y 値取得（Get Y Value）」VI の波形インスタンスの**モード**および**指標 / 秒**入力は、それぞれ **Y 位置形式**および **Y 位置**に名前変更されました。これらのインスタンスの**実際の指標 / 時間**出力は、**実際の Y 位置**に名前変更されました。
- 「Y 値取得（Get Y Value）」VI の DWDT Y 値取得インスタンスの**波形データ値**出力は、波形データの値を返します。
- 「Y 値取得（Get Y Value）」VI の DTbl Y 値取得インスタンスの**デジタルデータ値**出力は、デジタルデータの値を返します。このインスタンスの**デジタル入力**入力は、**デジタルデータ入力**に名前変更されました。このインスタンスの**デジタル出力**および**デジタル値**出力は、それぞれ**デジタルデータ出力**および**デジタル値**に名前変更されました。
- 「部分波形取得（Get Waveform Subset）」VI の WDT 部分波形取得および DWDT 部分波形取得インスタンスの**モード**、**波形入力**、**開始**、および**持続時間**は、それぞれ**開始 / 遅延形式**、**波形入力**、**開始**、および**持続時間**に名前変更されました。**指標**モードは、**サンプル**に名前変更されました。また、WDT 部分波形取得インスタンスには、新規で**開始 / 遅延形式**および**絶対時間**があります。「部分波形取得（Get Waveform Subset）」VI の WDT 部分波形取得および DWDT 部分波形取得インスタンスの**波形出力**、**実際の開始データ**、および**実際の持続時間**出力は、**波形出力**、**実際の開始**、**実際の持続時間**に名前変更されました。
- 「部分波形取得」VI の DTbl 部分デジタルインスタンスの**デジタル**、**開始**、**サンプル数**入力は、**デジタルデータ**、**開始**、**サンプル数**に名前変更されました。このインスタンスの**部分デジタル**出力は、**部分デジタルデータ**に名前変更されました。
- 「Y 値取得（Get Y Value）」、「波形から XY ペア（Waveform to XY Pairs）」、「指標波形配列（Index Waveform Array）」、「波形タイムスタンプアライメント（Align Waveform Timestamps）」、「波形連結（Append Waveforms）」、「波形 dt コピー（Copy Waveform dt）」、「スケール Δt（Scale Delta t）」の各 VI に、64 ビットの符号付き整数のインスタンスがあります。「波形から XY ペア」VI には、64 ビットの符号なし整数のインスタンスもあります。
- 「ブール配列からデジタルに変換（Boolean Array to Digital）」には、2 つの新規入力があります。**データ圧縮**はデジタル出力を圧縮するかどうかを指定します。**サンプルレート**は出力デジタル波形の秒単位のサンプル周波数を指定します。

- 「デジタル反転（Invert Digital）」VI の**エラー応答**入力は、**反転エラー応答**に名前変更されました。「デジタル反転」VI には、新しく 2 つの入力が追加されました。**信号指標**は、データの反転を開始する信号を指定します。**信号数**は、**信号指標**からカウントされる反転する信号の数を指定します。
- 「ビープ音（Beep）」VI の**システムアラートを使用？**入力は、LabVIEW でデフォルトのシステムアラートを使用して**周波数 (Hz)** と**持続時間 (ミリ秒)** を無視するかどうかを指定します。LabVIEW *7.x* 以前では、この VI に**強度**入力または**エラー**出力はありません。
- 「レポートをファイルに保存（Save Report to File）」VI の**置換のプロンプト？**入力が TRUE で、**レポートファイルパス**が既存のファイルへのパスを示している場合、既存ファイルを上書きするかどうかを確認するダイアログボックスが開きます。それ以外の場合は、既存ファイルは警告なしに上書きされます。
- 「表をレポートに追加（Append Table to Report）」VI の**列幅の単位（US)** は、**列幅の単位（デフォルト)** に変更されました。HTML レポートにこの VI を使用して表を作成する場合、**列幅の単位**入力は無視されます。代わりに、**列幅**入力に 100 を乗算した値が、ピクセル単位の最大列幅となります。
- 「VI のサブ VI リストをレポートに追加（Append VI List of SubVIs to Report）」VI の **Express VI 構成情報を含む**入力は、レポートにブロックダイアグラムにあるすべての **Express VI** の構成情報を含めるかどうかを指定します。
- スペース定数は**文字列**パレットに追加されました。このスペース定数は、ブロックダイアグラムに 1 文字スペースの文字列を提供します。
- 「文字列に平坦化（Flatten To String）」関数には 2 つの新規入力があります。**配列または文字列サイズを先頭に追加？**は、**任意のデータ**が配列または文字列の場合に、データサイズ情報を**データ文字列**の先頭に含むかどうかを示します。また、**バイト順序**は結果として得られた平坦化文字列のエンディアン形式を設定します。これらの新規入力のデフォルト値を使用する場合、この関数は以前のとおり動作します。この関数には、新規の**エラー入力**入力と新規の**エラー出力**出力もあります。
- 「文字列に平坦化（Flatten To String）」関数には **7.x データの変換**という新規のショートカットメニュー項目があります。これは、**タイプ文字列（7.x のみ）**出力を表示し、その出力に赤い **7.x** が付いたこの関数用のアイコンを表示します。**タイプ文字列 (7.x のみ)** を配線する場合にのみ、**タイプ定義を表示**ショートカットメニュー項目が表示されます。**タイプ文字列**出力は**タイプ文字列（7.x のみ）**に名前が変更されました。これは、端子がすでに旧バージョンの LabVIEW から配線された状態で、関数を右クリックしてショートカットメニューから **7.x データの変換**を選択している場合にのみ表示されます。
- 「文字列から非平坦化（Unflatten From String）」関数には 2 つの新規入力があります。**データは配列または文字列サイズを含む？**は、配列または文字列の入力の先頭からデータサイズ情報を読み取るかど

うかを示します。また、**バイト順序**は平坦化文字列の入力での、エンディアン形式の順序を示します。これらの新規入力のデフォルト値を使用する場合、この関数は以前のとおり動作します。この関数には新規出力として**バイナリ文字列の残り**もあります。この出力、この関数が変換しなかった残りのバイトを含みます。**エラー**出力は、新規の**エラー入力**入力と新規の**エラー出力**出力に変更されました。

- 「日付 / 時間を秒に変換（Date/Time to Seconds）」関数には **UTC?** という新規入力があります。これは、**日時**がグリニッジ標準時またはコンピュータに設定されたタイムゾーンであるかどうかを指定します。LabVIEW 7.*x* では、関数がグリニッジ標準時を基準に計算しているため、「日付 / 時間を秒に変換（Date/Time to Seconds）」関数は **(DST)** パラメータを無視していました。**UTC?** が FALSE（デフォルト）の場合、LabVIEW 8.0 では **(DST)** を基準に時間を変換します。LabVIEW 7.*x* または旧バージョンの LabVIEW で保存した VI を開く場合は、この変換を回避するために `-1` の値を **(DST)** に割り当てます。
- 「日付 / 時間を秒に変換」および「秒を日付 / 時間に変換（Seconds to Date/Time）」関数は、**日時**クラスタに**わずかな時間**という新規要素があります。この新規要素は、秒を基準に経過した小数点秒数です。
- 「秒を日付 / 時間に変換（Seconds to Date/Time）」関数の **UTC へ**入力は、**日時**がグリニッジ標準時またはコンピュータに設定されたタイムゾーンであるかどうかを指定します。
- 「日付 / 時間文字列をフォーマット（Format Date/Time String）」関数には **UTC 形式**という新規入力があります。これは、出力文字列がグリニッジ標準時またはコンピュータに設定されたタイムゾーンであるかどうかを指定します。
- 「文字列にフォーマット（Format Into String）」および「文字列からスキャン（Scan From String）」関数には、タイムスタンプデータ用のサポートが含まれます。**形式文字列の編集**ダイアログボックスには、**相対時間形式**および**タイムスタンプ形式**の変換オプションがあります。また、**スキャン文字列を編集**ダイアログボックスには、**相対時間をスキャン**および**タイムスタンプをスキャン**変換オプションがあります。
- 形式文字列で変数の順序を指定するには、「文字列にフォーマット（Format Into String）」関数で `$` 形式指定子の構文要素を使用します。
- 「プロパティを設定（Set Property）」VI の文字列、DBL、SGL、I32、I16、U8、およびタイムスタンプインスタンスの**見つからなければ作成**入力は、**プロパティ名**で指定したプロパティが存在しない場合にカスタムプロパティを作成するかどうかを指定します。

新規サンプル VI

LabVIEW 8.0 に追加されたサンプル VI について、説明を表示したり VI を起動したりするには、NI サンプルファインダの**参照**タブの **LabVIEW 8.0 の新しいサンプル**フォルダを参照してください。

外部コード関数の変更点

次の新しい関数は、DLL または CIN から呼び出せます。

- `CToLStr` は C 文字列を LabVIEW 文字列に変換します。
- `LToCStr` は LabVIEW 文字列を C 文字列に変換します。
- `DSSetHandleFromPtrNULLMeansEmpty` は、ポインタからのハンドルを割り当てて、ポインタからのデータをハンドルをハンドルにコピーします。`DSSetHandleFromPtrNULLMeansEmpty` 関数は `DSSetHandleFromPtr` 関数に似ていますが、ポインタにデータがないときにハンドルを 0 バイトに割り当てるのではなく Null にする点で異なります。
- `NIGetOneErrorCode` は数値エラーコードを適切なテキストの説明に変換します。

LabVIEW 7.*x* 以前では、CIN は `AZHPurge`、`AZHNoPurge`、`AZHLock`、`AZHUnlock` をサポートしません。

名前が変更されたプロパティとメソッド

LabVIEW 8.0 には、新規の VI サーバのクラス、プロパティ、メソッド、イベントが含まれます。LabVIEW 8.0 ででは、以下のプロパティおよびメソッドも名前変更されました。

プロパティ

以下のプロパティが名前変更されました。

表 6 LabVIEW 8.0 で名前が変更されたプロパティ

クラス	LabVIEW 7.1 での名前	LabVIEW 8.0 での名前
アプリケーション	アプリケーション : リアルタイムホスト接続	アプリケーション : ユーザインタフェース使用可
デジタル表	セル背景色	アクティブ セル : セル背景色
デジタル表	セルの前景色	アクティブ セル : セル前景色
デジタル表	上左表示セル	左上のセル
リストボックス	スクロールバー表示	表示項目 : 垂直スクロールバー表示
リストボックス	記号表示	表示項目 : 記号表示
複数列リストボックス	セルの背景色	アクティブ セル : セル背景色
複数列リストボックス	セルサイズ	アクティブ セル : セルサイズ
複数列リストボックス	列ヘッダ表示	表示項目 : 列ヘッダ表示
複数列リストボックス	水平スクロールバー表示	表示項目 : 水平スクロールバー表示
複数列リストボックス	記号表示	表示項目 : 記号表示

表 6　LabVIEW 8.0 で名前が変更されたプロパティ（続き）

クラス	LabVIEW 7.1 での名前	LabVIEW 8.0 での名前
複数列リストボックス	垂直スクロールバー表示	表示項目 : 垂直スクロールバー表示
スケール	マイナー目盛りカラー	目盛りカラー : マイナー目盛りカラー
文字列	スクロールバー表示	垂直スクロールバー表示
表	セル背景色	アクティブ セル : セル背景色
表	セルサイズ	アクティブ セル : セルサイズ
表	列ヘッダ表示	表示項目 : 列ヘッダ表示
表	水平スクロールバー表示	表示項目 : 水平スクロールバー表示
表	指標表示	表示項目 : 指標表示
表	行ヘッダ表示	表示項目 : 行ヘッダ表示
表	垂直スクロールバー表示	表示項目 : 垂直スクロールバー表示
ツリー制御器	ドラッグを許可	ドラッグ / ドロップ : 項目のドラッグを許可
ツリー制御器	項目間のドラッグを許可	ドラッグ / ドロップ : 項目間のドロップを許可
ツリー制御器	親項目のドラッグを許可	ドラッグ / ドロップ : 親項目のドラッグを許可
波形チャート	凡例を自動サイズ調整	凡例 : 自動サイズ調整
波形グラフ	凡例を自動サイズ調整	凡例 : 自動サイズ調整

メソッド

以下のメソッドが名前変更されました。

表 7　LabVIEW 8.0 で名前が変更されたメソッド

クラス	LabVIEW 7.1 での名前	LabVIEW 8.0 での名前
制御器	Dflt に初期化	デフォルト設定に戻す
リストボックス	Get DblClk Row	ダブルクリックされた行を取得
複数列リストボックス	Get DblClk Row	ダブルクリックされた行を取得
テキスト	MoveToDefLoc	デフォルト位置へ移動
ツリー制御器	カスタム項目記号 : 記号を復元	カスタム項目記号 : ビルトイン記号に復元
ツリー制御器	カスタム項目記号 : 記号を復元	カスタム項目記号 : 全てをビルトイン記号に復元

表 7　LabVIEW 8.0 で名前が変更されたメソッド（続き）

クラス	LabVIEW 7.1 での名前	LabVIEW 8.0 での名前
ツリー制御器	カスタム項目記号 : 記号を設定	カスタム項目記号 : カスタム記号に設定
ツリー制御器	カスタム項目記号 : 記号配列を設定	カスタム項目記号 : カスタム記号配列に設定
VI	FP を閉じる	フロントパネル : 閉じる
VI	VI 文字列をエクスポート	VI 文字列 : エクスポート
VI	全ての制御器の値を取得	制御器の値 : すべてを取得 [平坦化]
VI	全ての制御器の値を取得 [バリアント]	制御器の値 : すべてを取得 [バリアント]
VI	制御器の値を取得	制御器の値 : 取得 [平坦化]
VI	制御器の値を取得 [バリアント]	制御器の値 : 取得 [バリアント]
VI	スケールされたダイアグラム画像を取得	ブロックダイアグラム : スケールされた画像を取得
VI	ロック状態を取得	ロック状態 : 取得
VI	パネル画像を取得	フロントパネル : パネル画像を取得
VI	スケールされたパネル画像を取得	フロントパネル : スケールされたパネル画像を取得
VI	VI アイコンを画像データとして取得	VI アイコン : 画像データとして取得
VI	VI 文字列をインポート	VI 文字列 : インポート
VI	リモートパネル制御をロック	リモートパネル : 制御ロックを設定
VI	現在のすべての値をデフォルト設定にする	デフォルト値 : 現在の設定をデフォルト設定にする
VI	FP を開く	フロントパネル : 開く
VI	すべてをデフォルト設定に戻す	デフォルト値 : 全てをデフォルト設定に戻す
VI	VI アイコンをファイルへ保存	VI アイコン : ファイルへ保存
VI	制御器の値を設定	制御器の値 : 設定 [平坦化]
VI	制御器の値を設定 [バリアント]	制御器の値 : 設定 [バリアント]
VI	ロック状態を設定	ロック状態 : 設定

表 7　LabVIEW 8.0 で名前が変更されたメソッド（続き）

クラス	LabVIEW 7.1 での名前	LabVIEW 8.0 での名前
VI	VI アイコンをファイルから取得	VI アイコン : ファイルから設定
VI	VI アイコンを画像データから設定	VI アイコン : 画像データから設定
VI	リモートパネル制御ロックを解除	リモートパネル : 制御ロックを解除

エラーコードの機能強化

エラーコード –8999 ～ –8000 を使用して、カスタムのエラーメッセージを定義できます。

LabVIEW 8.0 では、エラーコードに以下の変更が行われています。

表 8　LabVIEW 8.0 エラーコードの変更点

LabVIEW 7.1 エラーコード	LabVIEW 8.0 エラーコード
20003	20012
20101	20111
20102	20112
20103	20113
20104	20114

その他

上記の他、LabVIEW 8.0 では、以下の点が変更されました。

- LabVIEW 8.0 では**登録情報**ダイアログボックスが表示されません。以前は、登録情報ダイアログボックスが未登録バージョンの LabVIEW を起動すると必ず表示されていました。LabVIEW 8.0 はインストール時に氏名や企業名などの情報を収集します。
- **(Windows)** LabVIEW 7.*x* 以前では、<Ctrl-Tab> キーを押すと、ウィンドウを開いた順番で繰り返し LabVIEW ウィンドウを開きます。LabVIEW 8.0 では、<Ctrl-Tab> キーを押すと、ウィンドウが画面上に表示されている順番で繰り返し LabVIEW ウィンドウを開きます。この順番は、<Alt-Tab> キーを押して確認できる順番と同じです。**(Linux)** ウィンドウの順番は、使用するウィンドウマネージャにより異なります。
- ツリー制御器で子ノード項目の下に項目をドラッグすると、その項目は子ノードと同じ階層レベルで子ノードの下に配置されます。

- VI 名、VI パス、VI フロントパネルの画像、VI リファレンスの 16 進数値を表示するには、VI Refnum(パネルイメージ) プローブを使用します。値が無効な Refnum の場合に、このプローブを使用してブレークポイントを設定できます。
- リストボックスと複数列リストボックスの無効項目プロパティを、リストボックスにない項目に対して設定すると、エラーが返されます。
- **(Mac OS)** LabVIEW でマウススクロールホイールをサポートします。ケースストラクチャ、イベントストラクチャ、スタックシーケンスで、セレクタラベル上にカーソルを移動し、<Command> キー押しながらマウスホイールを動かすと、これらのストラクチャのサブダイアグラム間をスクロールできます。
- **(Mac OS)** LabVIEW ではポップアップの <Control>- クリックのみを受け入れますが、<Command>- クリックのキーの組み合わせは受け入れません。
- **制御器エディタ**ウィンドウのカスタマイズモードでは、ショートカットメニューから**ファイルからピクチャをインポート**項目を選んで、カスタム制御器に使われているピクチャをファイルからインポートできます。また、**編集→貼り付け**を選択して、装飾体をクリップボードの画像に置換することもできます。装飾体をファイルの画像に置換するには、装飾体を右クリックしてショートカットメニューから**ファイルからインポート**を選択します。
- **オプション**ダイアログボックスの**パスオプション**ページにあるカスタム**メニューディレクトリ**は指定できません。
- **パスオプション**ページのカスタム **VI 検索パス**を指定する場合は、`<osdatadir>` シンボリックパスを使用できます。
- タイムスタンプをバリアントに変換すると、バリアント表示器が配線先のタイムスタンプの現在の値を表示します。
- CVI 関数パネルコンバータダイアログボックスと **VXIPNP ドライバを更新**ダイアログボックスは、LabVIEW 8.0 では廃止されました。この機能を有効にするには、`ni.com/idnet`（英語）から LabVIEW Interface Generator for LabWindows™/CVI™ Instrument Drivers ツールをダウンロードしてください。
- **Enter キーでテキスト入力を終了**オプションは、**オプション**ダイアログボックスの**環境**ページにあります。
- **変更の説明**ダイアログボックスには、選択された VI のディスクのパスを一覧表示する**パス**テキストボックスが含まれます。**パス**テキストボックスは、同名の VI を区別するために使用することもできます。
- VI のデバッグを無効にする処理のパフォーマンスが多少向上しました。
- **VI ライブラリマネージャ**ウィンドウは、**LLB マネージャ**ウィンドウに名前変更されました。LLB 内の項目をトップレベルとして示すには、**ファイル**リストにある項目を右クリックしてショートカットメニューから**トップレベル ?** を選択します。**LLB マネージャ**ウィンドウでの変更をキャンセルすることはできません。

- **VI ライブラリを編集**ダイアログボックスが**ツール**メニューから削除されました。ライブラリから VI を削除する、または VI をライブラリ内でトップレベルの VI として表示するには、**LLB マネージャ**ウィンドウを使用します。
- LabVIEW 7.*x* 以前では、エラークラスタをケースストラクチャに配線すると `True` ケースがエラーケースになり、`False` ケースがエラーなしケースになっていました。LabVIEW 8.0 では、エラークラスタをケースストラクチャに配線すると `True` ケースがエラーなしケースになり、`False` ケースがエラーケースになります。
- フロントパネルで制御器や表示器を右クリックし、**作成→プロパティノード**または**作成→インボークノード**を選択して、ショートカットメニューからプロパティやメソッドを選択すると、ブロックダイアグラムで特定のプロパティまたはメソッドを作成できます。
- **(Windows)** NI-CAN と **CAN 制御器**パレットにある CAN チャンネル制御器を使用して、NI-CAN チャンネル API にアクセスします。**CAN 制御器**パレットにある制御器の詳細については、NI-CAN のドキュメントを参照してください。
- **プロパティ**ダイアログボックスの**形式と精度**ページで、**精度のタイプ**を**有効数字**に設定する場合は、単精度浮動小数点数の場合は**桁**フィールドに 1 ～ 6 の数値を、倍精度拡張精度浮動小数点数の場合は**桁**フィールドに 1 ～ 13 の値を使用することをお勧めします。
- スケールとして X 軸を選択し、t0 で指定された値の代わりに 0 に対する x スケールの開始を設定する場合、チャート、強度チャート、および強度グラフ**プロパティ**ダイアログボックスの**スケール**ページで、**X 軸上の波形時間スタンプを無視**チェックボックスをオンにします。スケールとして Y 軸を選択し、個別データラインとしてデジタル波形データを表示する場合は、**デジタルバスを展開**チェックボックスをオンにします。
- 複数のファイルを同時に開けます。**(Windows) ファイル→開く**を選択して標準のファイルダイアログボックスを表示し、<Shift> キーまたは <Ctrl> キーを押しながら複数のファイルを選択できます。**(Mac OS)** <Shift> キーまたは <Option> キーを押します。**(Linux)** <Shift> キーまたは <Alt> キーを押します。
- すべてのブラウザで、**ウェブパブリッシュツール**ダイアログボックスの**モニタ**オプションがサポートされています。
- LabVIEW 7.*x* 以前では、**ツール→比較→ VI の比較**を選択して**選択**ボタンをクリックすると、**名前で VI を選択**ダイアログボックスが表示されていました。LabVIEW 8.0 では、**ツール→比較→ VI の比較**を選択して**選択**ボタンをクリックすると、**VI を選択**ダイアログボックスが表示されます。
- LabVIEW 8.0 では、**ツール→比較→ VI 階層を比較**を選択すると、メモリ内のファイルシステムまたは VI リストから VI を選択できます。
- LabVIEW 7.*x* では、ダイアグラムのオブジェクトを右クリックして、ショートカットメニューから***ソースパレット***を選択すると、サブパ

レットから同様のオブジェクトにアクセスできます。ここで、***ソース***はブロックダイアグラムのオブジェクトを含むサブパレット名です。LabVIEW 8.0 でも、ブロックダイアグラムのオブジェクトを右クリックして、ショートカットメニューから**置換→*ソースパレット***を選択すると、ブロックダイアグラムのオブジェクトを、そのブロックダイアグラムのオブジェクトを含むサブパレットから類似するオブジェクトに置換できます。また、ワイヤを右クリックしてショートカットメニューから **挿入→*ソースパレット***を選択して、最も頻繁に使用するソースパレットからオブジェクトを挿入することもできます。これにより、そのワイヤが接続する 2 つのオブジェクトの間に選択したオブジェクトが挿入されます。

- LabVIEW 7.*x* 以前では、タイプ定義のデータタイプおよびタイプ定義更新のインスタンスを変更すると、インスタンスは、現在のデータ、デフォルトデータ、プライベートデータ、ラベル、キャプションのみを保存します。LabVIEW 8.0 では、タイプ定義のインスタンスはより多くの属性を保存することがあります。
- LabVIEW 8.0 では、タイプ定義を別の制御器に置換すると、この制御器は LabVIEW 7.*x* 以前で保存するタイプ定義の属性よりも少ない数の属性を保存することがあります。
- LabVIEW では、必要に応じてタイプ定義が伝達されます。
- **幅と高さを設定**ダイアログボックスは、**オブジェクトをサイズ変更**に名前変更されました。このダイアログボックスの**現在のサイズ値**リストボックスでは、**幅**や**高さ**を変更してもサイズ変更できないオブジェクトにアスタリスクが付けられます。
- LabVIEW 7.*x* では、VI が見つからない場合はそれを同じ名前の VI で上書きする必要がありました。LabVIEW 8.0 では、見つからない VI を別の名前の VI で上書きできるようになりました。見つからない VI のすべてのインスタンスは、選択した VI に置き換えられます。
- LabVIEW 7.*x* 以前のバージョンでは、<Ctrl-A> キーを押して、フロントパネルまたはブロックダイアグラムのオブジェクトの調整操作を繰り返すことができました。LabVIEW 8.0 では、<Ctrl-Shift-A> キーを押して調整操作を繰り返します。<Ctrl-A> キーを押して、フロントパネルまたはブロックダイアグラムのすべてのオブジェクトを選択します。
- LabVIEW がサブ VI をロードするタイミングを構成できます。サブ VI を右クリックしてショートカットメニューから**設定呼び出し**を選択し、**VI 呼び出し構成**ダイアログボックスを表示します。このショートカットメニュー項目は、VI サーバ呼び出しをサポートするアプリケーションインスタンスで開いている VI のみに使用できます。PDA および FPGA ターゲットなど、VI サーバ呼び出しをサポートしないアプリケーションインスタンスで開かれている関数、Express VI、多態性 VI、VI には使用できません。発呼者 VI が大きい場合は、ダイアログボックスで**最初の呼び出しでロードして保持**オプションを選択することによって、ロードする時間とメモリを節約できます。このオプションを選択すると、サブ VI は発呼者 VI によって必要とされ

るまでサブ VI はロードされず、操作完了後にメモリからサブ VI を解放することができます。

- LabVIEW 8.0 では、ブロックダイアグラムからフロントパネルの制御器をコピーして、新規 VI のブロックダイアグラムへ貼り付けると、フロントパネルの配置が元の VI の状態と異なります。非表示領域で重複や表示を回避するには、左上隅にフロントパネルのオブジェクトを配置します。
- 西暦として表示される範囲が 1904 ～ 2038 年から 1600 ～ 3000 年に拡張されました。
- **詳細ヘルプ**ウィンドウ内で復帰文字を表示するには、各パラグラフを 2 つの復帰文字で区切ります。
- 押されたキーが VI メニューのキーボードショートカット（<Ctrl-C> や <Ctrl-V> など）に一致する場合、メニュー項目が有効か無効かにかかわらず、LabVIEW はキーダウンイベントを生成しません。
- アプリケーション : 言語プロパティは **ko** を返して、LabVIEW 環境の言語が韓国語であることを示すことができます。
- 複数の VI をタグ付きテキストファイルにエクスポートしたり、タグ付きテキストファイルから複数の VI にインポートするには、それぞれ VI 文字列 : エクスポートおよび VI 文字列 : インポートメソッドを使用します。

National Instruments、NI、ni.com、および LabVIEW は National Instruments Corporation（米国ナショナルインスツルメンツ社）の商標です。National Instruments の商標の詳細については、ni.com/legal の「Term of Use」セクションを参照してください。本文書中に記載されたその他の製品名および企業名は、それぞれの企業の商標または商号です。National Instruments の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）にて使用されるコンポーネントに関する著作権、条件および免責条項の一覧は、USICopyrights.chm を参照してください。

© 1998–2005 National Instruments Corporation. All rights reserved.

Printed in Ireland. 371780A-0112 1005