

LabVIEW™ アップグレードノート

バージョン 7.0

このアップグレードノートでは、Windows、Mac OS、および UNIX 対応 LabVIEW をバージョン 7.0 にアップグレードするための手順や、その際に起こり得る問題、および LabVIEW 環境の機能強化などを含む新機能について説明します。

LabVIEW 環境で強化された各機能についての詳細や、LabVIEW 環境のカスタマイズに関する情報については、本書の「[LabVIEW 環境の機能向上](#)」のセクションを参照してください。アップグレードされる方は、LabVIEW 7.0 をご使用になる前に必ず「[LabVIEW 環境の機能向上](#)」のセクションをお読みになることをお勧めします。『LabVIEW 入門』マニュアルには、LabVIEW 7.0 の新機能や LabVIEW 環境の機能強化等について確認していただくための練習問題が含まれています。

詳細については

LabVIEW 7.0 の機能の詳細については、『LabVIEW ユーザマニュアル』および LabVIEW ヘルプを参照してください。『LabVIEW ヘルプ』にアクセスするには、[ヘルプ→オンラインヘルプリファレンス](#)を選択します。『LabVIEW ユーザマニュアル』やその他の LabVIEW マニュアルの PDF バージョンにアクセスするには、LabVIEW のメニューのヘルプ→[LabVIEW ドキュメントライブラリ](#)を表示を選択してください。LabVIEW ドキュメントライブラリでは、すべての LabVIEW マニュアルおよびアプリケーションノート (PDF) を表示できます。

PDF を表示するには、Adobe Acrobat Reader を使用してください。Acrobat Reader をダウンロードするには、アドビシステムズ社のホームページ www.adobe.co.jp にアクセスしてください。

目次

アップグレード時の問題点	3
VI を変換する	4
ツールセット、計測器ドライバ、およびアドオンをアップグレードする	4
旧バージョンの LabVIEW をアップグレードする	5

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

2003 年 06 月版
322784C-01

LabVIEW 6.x からアップグレードする.....	6
LabVIEW 5.x からアップグレードする.....	10
LabVIEW 4.x からアップグレードする.....	12
LabVIEW 3.x 以前のバージョンからアップグレードする.....	14
LabVIEW 7.0 の新機能.....	14
Express VI.....	14
ダイナミックデータタイプ.....	14
LabVIEW ダイアログボックスにおける機能強化.....	15
テンプレート VI を使用して新規 VI を作成する.....	15
制御器および関数パレットにおける機能強化.....	15
プロパティダイアログボックス.....	17
自動エラー処理.....	18
フロントパネル端子のアイコン.....	18
自動ワイヤルーティング.....	18
ストラクチャのサイズを自動的に変更する.....	19
フラットシーケンスストラクチャ.....	19
NI サンプルファインダ.....	19
制御器および表示器のデフォルトスタイルを変更する.....	20
LabVIEW データディレクトリ.....	20
NI アプリケーションプログラミングインタフェースのアップ デート.....	20
DAQ アシスタント.....	20
計測器 I/O アシスタント.....	21
I/O 制御器.....	21
計測器ドライバのアップデート.....	21
I/O におけるその他の機能向上.....	21
.NET 関数.....	22
Mac OS X のサポート.....	22
LabVIEW 環境の機能向上.....	23
新規ブロックダイアグラムオプション.....	23
新規フロントパネルオプション.....	24
新規その他のオプション.....	25
6.1 から変更されたデフォルト.....	25
オプションダイアログボックスに新たに追加されたページ.....	26
アライメントグリッドを使用してオブジェクトを整列する.....	26
自動ツール選択での機能向上.....	26
単位ラベルの編集とカスタマイズ.....	27
旧バージョンで使用できるように VI を保存する.....	28
印刷およびレポート生成の機能向上.....	28
その他の環境における機能強化.....	29
ツリー制御器を使用する.....	30
サブパネル制御器を使用する.....	30
リング制御器における機能強化.....	31
コンボボックス制御器を使用する.....	31
グラフおよびチャートにおける機能強化.....	32
VI をフローティングウィンドウで実行する.....	33
フロントパネルにおけるその他の機能向上.....	33

イベントを動的に登録して、ユーザイベントを処理する.....	35
サブ VI を拡張可能ノードとして表示する.....	35
不良ワイヤにおける機能強化.....	35
多形性 VI における機能強化.....	36
タイムスタンプ制御器およびデータタイプ.....	36
一般プローブとカスタムプローブ.....	37
数値をフォーマットする.....	38
波形の数値データの有効形式.....	38
ピクチャ制御器における機能強化.....	38
VI でのカーソルの外観を変更する.....	39
デジタル波形データタイプおよびデジタルデータタイプ.....	40
フィードバックノード.....	40
ループ内でシフトレジスタをトンネルに入れ替える.....	41
ブロックダイアグラムにおけるその他の機能強化.....	41
VI からデータを電子メールで送信する.....	42
DataSocket における機能強化.....	42
UDP における機能強化.....	42
ActiveX イベントを処理する.....	43
通信、VI サーバ、およびリモートフロントパネルにおけるその 他の機能強化.....	43
VI サーバの新規プロパティとメソッド.....	44
既存の VI サーバプロパティおよびメソッドへの変更点.....	45
ドキュメントにおける内容の強化.....	45
LabVIEW 7.0 でのその他の機能および変更点.....	46
アプリケーションビルダにおける機能強化.....	46
新規 VI および関数.....	46
既存の VI および関数における変更点.....	48
その他.....	49

アップグレード時の問題点

LabVIEW 6.x からアップグレードする場合は、本書の「[VI を変換する](#)」、「[ツールセット、計測器ドライバ、およびアドオンをアップグレードする](#)」、および「[LabVIEW 6.x からアップグレードする](#)」の各セクションをまずお読みください。

LabVIEW 5.x からアップグレードする場合は、本書の「[VI を変換する](#)」、「[ツールセット、計測器ドライバ、およびアドオンをアップグレードする](#)」、および「[LabVIEW 5.x からアップグレードする](#)」の各セクションをまずお読みください。

LabVIEW 4.x からアップグレードする場合は、本書の「[VI を変換する](#)」、「[ツールセット、計測器ドライバ、およびアドオンをアップグレードする](#)」、および「[LabVIEW 4.x からアップグレードする](#)」の各セクションをまずお読みください。

LabVIEW 3.x からアップグレードする場合は、本書の「ツールセット、計測器ドライバ、およびアドオンをアップグレードする」および「[LabVIEW 3.x 以前のバージョンからアップグレードする](#)」の各セクションをまずお読みください。

VI を変換する

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 7.0 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 7.0 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し、それに余分なメモリリソースが使用されます。



メモ LabVIEW 7.0 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。LabVIEW 6.1 で実行できるようにするためには、**ファイル→オプション付き保存**を選択し、**旧バージョン用に保存**をクリックして VI を保存します。LabVIEW 6.0 以前のバージョンで使用する予定のある VI は、LabVIEW 7.0 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。

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

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。**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 VI ライブラリ内の VI はアルファベット順に変換されます。変換プロセスで上位の VI が先に処理される場合、一括コンパイルでは、上位の VI を先に開いた場合とおよそ同じ量のメモリが必要になります。

メモリの使用状況を監視し、それまでに使用したメモリ量の合計を表示するには、**ヘルプ→LabVIEW について**を選択します。

ツールセット、計測器ドライバ、およびアドオンをアップグレードする

LabVIEW 7.0 をインストールしたら、アドオンもすべて最新バージョンであることを確認し、LabVIEW 7.0 ディレクトリにアドオンを再インストールします。

また、既存のツールセット VI も、LabVIEW 7.0 で使用するためには一括コンパイルする必要があります。VI の一括コンパイルの詳細については、本書の「VI を変換する」のセクションを参照してください。LabVIEW 7.0

は、原則として LabVIEW 4.0 以降対応のツールセット、計測器ドライバ、およびアドオンと互換性がありますが、いくつかの例外がありますので以下に示します。

- **(開発システム) LabVIEW アプリケーションビルダ** : LabVIEW アプリケーションビルダ 7.0 にアップグレードする必要があります。LabVIEW プロフェッショナル開発システムバージョン 7.0 には、アプリケーションビルダ 7.0 ライブラリが含まれています。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 アプリケーションビルダユーザガイド』を参照してください。
- **(開発システム) LabVIEW プロフェッショナル G 開発ツールキット** : プロフェッショナル G 開発ツールキット 5.0 以降のバージョンをご利用の場合は、LabVIEW プロフェッショナル開発システム 7.0 にアップグレードする必要があります。このアップグレードは、プロフェッショナル G 開発ツールキット 5.1 の既存のユーザについては無料です。プロフェッショナル開発システムバージョン 7.0 には、プロフェッショナル G 開発ツールキットの機能が含まれています。

旧バージョンの LabVIEW をアップグレードする

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 以前の LabVIEW は、labview x.x ディレクトリにインストールされます。x.x はバージョン番号です。

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。LabVIEW 7.0 をインストールしたら、旧バージョンの LabVIEW 環境設定ファイルを LabVIEW 7.0 ディレクトリにコピーします。

(Windows) 環境設定は labview.ini ファイルに格納されます。

(Mac OS) 環境設定は、システム : 初期設定フォルダ内の LabVIEW Preferences テキストファイルに格納されます。

(UNIX) 環境設定は、ユーザのホームディレクトリ内の .labviewrc ファイルに格納されます。

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 ディレクトリの内容をコピーします。LabVIEW 7.0 をインストールしたら、ファイルを LabVIEW 7.0 ディレクトリの user.lib ディレクトリにコピーします。

(Windows) また、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 バージョンを削除することによっても、

既存のバージョンを LabVIEW 7.0 に入れ替えることができます。アンインストールを実行しても、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。



注意

ユーザ独自の VI や制御器を vi.lib ディレクトリにある既存の .lib ファイルに保存した場合、LabVIEW は .lib ファイルにあるユーザが保存したすべての VI や制御器も含め、.lib ファイルをアンインストールします。

LabVIEW 7.0 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。

LabVIEW のバージョンによって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 6.x からアップグレードする

LabVIEW 6.x から LabVIEW 7.0 にアップグレードする際、以下のような問題が発生する可能性があります。

Windows 95 のサポート

LabVIEW 7.0 は Windows 95 をサポートしていません。

HiQ のサポート

Mac OS 対応の LabVIEW 7.0 では HiQ 機能はサポートされていません。また、LabVIEW 7.0 より後のバージョンでは、すべてのプラットフォームにおいて HiQ 機能はサポートされなくなります。アプリケーションで HiQ VI を使用する場合は、解析および数学 VI に置き換えることを検討してみてください。解析および数学 VI の使用方法の詳細については、『LabVIEW ヘルプ』を参照してください。

シリアル互換性 VI

シリアル互換性 VI は関数パレットに表示されません。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。

このバージョンの LabVIEW では、オペレーティングシステムのシリアルドライバとの通信に serpdrv ドライバは使用されなくなりました。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/ 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 7.0 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.lib にある「シリ

アルエイリアスポート設定 (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ポート設定」 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、`examples¥instr¥smp1ser1.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、ループが実行しない場合 For ループは未定義のデータを生成しました。LabVIEW 6.1 以降では、For ループのカウント端子に 0 を配線した場合や、自動指標付けが有効になった状態で入力として空の配列を For ループに配線した場合、For ループはデフォルトデータを生成します。ループは実行せず、自動指標付けが無効になった状態の出力トンネルにのみ、トンネルのデータタイプのデフォルト値が含まれます。

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。

(Windows 98) より多くのクライアントをサポートするように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ことができます。

(Windows 2000/NT/XP/Me、Mac OS、および UNIX) より多くのクライアントをサポートするように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ことができます。

マルチスレッド割り当て

LabVIEW 7.0 では、以前のバージョンに比べ VI の実行により多くのスレッドを割り当てます。このような変更のため、呼び出した DLL が実際には再入実行可能でない場合に「ライブラリ関数呼び出しノード (Call Library Function Node)」を誤って再入実行可能としてマークすると、マルチスレッドでエラーが発生することがあります。再入実行に関する詳細については、『Using External Code in LabVIEW』マニュアルの Chapter 2 「Shared Libraries (DLLs)」を参照してください。

LabVIEW におけるスレッドの割り当て方法を変更するには、`vi.lib\Utility\sysinfo.lib`にある「スレッド構成 (Thread Configuration)」VI を使用します。また、**VI プロパティ**ダイアログボックスの**実行**ページにある**再入実行**チェックボックスをオフにすることによって、VI の再入実行を無効にすることもできます。

計測器ドライバ

現在の LabVIEW パッケージには、計測器ドライバを収録した LabVIEW 計測器ドライブライブラリ 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

「複合演算 (Compound Arithmetic)」関数の使用後、「単位変換 (Convert Unit)」関数を使用して余分な単位を削除する必要はありません。

LabVIEW 7.0 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。これらの単位の省略形は以前と同じです。次の表は、LabVIEW 6.1 と 7.0 での単位変換ファクタの変更の詳細を示しています。

単位	6.1 の定義	7.0 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 vs. 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	16.5 ft	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

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。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。

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**に設定されていました。LabVIEW 7.0 では、それらの数値制御器はデータタイプのデフォルトデータ範囲値を使用するように変更されています。

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、強制ドットはランタイム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、ユーザが新しい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。それ以外の場合、ボタンラベルは**開く**になっています。LabVIEW 7.0 では、ユーザが変更しない限りボタンラベルは常に **OK** です。ボタンのラベルを変更するには、「ファイルダイアログ」関数の**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」関数を使用する場合は、VI の動作を再確認して、デフォルトラベルの **OK** が VI の機能の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 (Control Online Help)」関数

「オンラインヘルプを制御」関数の**ヘルプファイル**へのパス入力は必須入力です。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

することができます。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

「VI を実行」メソッドの**自動処理 Ref** パラメータを TRUE に設定すると、メソッドがエラーを返した場合でもリファレンスは破棄されます。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、これによって VI が壊れることがあります。

ロードされたときにフロントパネルを表示する

LabVIEW 7.0 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 サーバを使って 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、「FP を開く」メソッドを使用してフロントパネルをプログラムの的に表示する必要があります。

「VI リファレンスを開く (Open VI Reference)」関数

「VI リファレンスを開く」関数を使用してテンプレートへのリファレンスを作成したときにすでにテンプレートがメモリにある場合、関数はエラーを返します。

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、または仮想名をプログラムで使用する場合、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、大文字小文字も違ってはなりません。

テクニカルサポートフォーム

LabVIEW インストーラでは、`techsup.11b` はインストールされません。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については、ナショナルインスト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 からアップグレードする

LabVIEW 5.x から LabVIEW 7.0 にアップグレードする際、以下のような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 6.x からアップグレードする](#)」のセクションを参照してください。

各バージョンでの新機能および変更点については、LabVIEW のバージョン 5.x から 7.0 までの各バージョンの『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/manuals) を参照してください。

データログファイルを変換する

LabVIEW 7.0 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。データログファイルが LabVIEW 6.0 より前のバージョンであるか、波形データタイプを含んでいる場合、LabVIEW 7.0 はファイルを読み取ったり追加したりできるように変換します。他の場合はすべて、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たりしても、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 7.0 ではそのファイルを LabVIEW 7.0 形式に変換するように促すプロンプトが表示されます。変換するように選択すると、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き換えられます。ファイルを変換しないように選択すると、エラーが返されファイルは開きません。



メモ 今後も古いデータを LabVIEW 6.1 以前のバージョンで使うことがある場合は、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
```

(UNIX) 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変換されないようにするには、環境設定を `False` に設定します。

LabVIEW 5.x の VI サーバと LabVIEW 7.0 クライアント間の互換性の問題

LabVIEW 7.0 クライアントを LabVIEW 5.x アプリケーションの VI サーバに接続するとエラーになります。これは、LabVIEW 5.x のアプリケーションでは LabVIEW 7.0 VI サーバプロトコルの新しい機能を認識できないためです。

LabVIEW 5.x クライアントから LabVIEW 7.0 アプリケーションの VI サーバへの接続もできません。

指数表記

LabVIEW 6.0 以前では、 \wedge 演算子は「フォーミュラノード (Formula Node)」で指数を表していました。LabVIEW 7.0 では、指数を表す新しい演算子は $**$ で、 $x**y$ のように使います。 \wedge 演算子は、ビット単位の排他的論理和 (XOR) 演算を表します。

UDP 関数

標準の UDP 関数は、ネットワーク通信に使用します。UDP VI は、互換性 VI として `vi.lib¥_oldvers¥_oldvers.lib` に入っています。

LabVIEW 4.x からアップグレードする

LabVIEW 4.x から LabVIEW 7.0 にアップグレードする際、以下のような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 5.x からアップグレードする](#)」および「[LabVIEW 6.x からアップグレードする](#)」の各セクションを参照してください。

各バージョンでの新機能および変更点については、LabVIEW のバージョン 4.x から 7.0 までの各バージョンの『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/manuals) を参照してください。

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

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、データが配列でない場合はブールデータが 2 バイトに格納されます。データが配列の場合は、LabVIEW 4.x では各ブール要素が 1 ビットに格納されます。LabVIEW 7.0 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。新しいブール値のデータ形式によって、コードインタフェースノード (CIN) でのデータ操作に影響がありますが、LabVIEW 7.0 では既存の CIN との間に互換性があります。

1 つまたは複数のブール値を含んでいるバイナリデータを LabVIEW 4.x のファイルに書き込む場合、その形式は LabVIEW 7.0 で同じデータを書き込む場合とは異なります。LabVIEW 7.0 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。「ファイルに書き込み (Write File)」、「ファイル読み取り (Read File)」、「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 5 つの関数には、**4.x データの変換** ショートカットメニュー項目があります。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扱うことができます。LabVIEW 4.x 用にフォーマットされたデータを作成するには、「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、「ファイル読み取り」、「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 7.0 ではその関数上に赤い 4.x を描画して LabVIEW 4.x 形式に、あるいは LabVIEW 4.x 形式からデータを変換中であることを示します。データを変換しないようにするには、**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 7.0 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 7.0 でロードすると、「ファイルに書き込み」、「ファイル読み取り」、「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 で LabVIEW 7.0 のブールデータ形式を使用する必要がある場合は、各関数でショートカットメニューから **4.x データの変換**を選択します。原則として、LabVIEW 4.x 以前のバージョンで作成し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 7.0 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

VI 制御 VI は **関数**パレットには表示されず、`vi.lib\utility\viCtrl.11b` に互換性 VI として含まれています。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 (Call By Reference)」、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 7.0 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 が返すエラーコードは 7 および 1000 でした。LabVIEW 7.0 の VI 制御 VI は、1004 と 1003 のコードを返します。LabVIEW 4.x で作成した VI がエラーコード 7 および 1000 をチェックする場合は、その VI を LabVIEW 7.0 で実行するには VI に変更を加える必要があります。

DDE VI

(Windows) DDE VI は **関数**パレットには表示されず、`vi.lib\platform\dde.11b` に互換性 VI として含まれています。

LabVIEW 3.x 以前のバージョンからアップグレードする

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/jp) 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 からアップグレードする](#)」、[「LabVIEW 5.x からアップグレードする」](#)、および「[LabVIEW 6.x からアップグレードする](#)」の各セクションを参照してください。

各バージョンでの新機能および変更点については、LabVIEW のバージョン 3.x から 7.0 までの各バージョンの 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/manuals) を参照してください。

LabVIEW 7.0 の新機能

LabVIEW 7.0 の機能については、『[LabVIEW ユーザマニュアル](#)』および『[LabVIEW ヘルプ](#)』を参照してください。

Express VI

Express VI は、一般的な計測タスクに使用します。Express VI は、ダイアログボックスを使用して構成するため、最小限の配線しか必要としない関数ノードです。Express VI は関数パレットに含まれ、白い背景と青の枠で囲まれたアイコンで表示されます。

Express VI の使用方法の詳細については、『[LabVIEW 入門](#)』を参照してください。他の標準 VI および関数とともに Express VI を使用する例については、[examples\express](#) ディレクトリを参照してください。

ダイナミックデータタイプ



ダイナミックデータタイプは、左図のように、紺色の端子として表示されます。ほとんどの Express VI は、ダイナミックデータに対応しています。ダイナミックデータは、数値、波形またはブール値のデータを受け入れる任意の表示器または入力に配線できます。ダイナミックデータは、データを最良な形で表示する表示器に配線します。表示器には、グラフ、チャート、数値表示器などがあります。

ダイナミックデータタイプには、信号に関連付けられたデータのほかに、信号名やデータが集録された日付と時間など、信号についての情報を提供する属性が含まれます。属性はグラフまたはチャートに信号を表示する方法を指定します。たとえば、「DAQ アシスタント (DAQ Assistant)」 Express VI を使用して信号を集録しその信号をグラフにプロットすると、信号名がグラフのプロット凡例に表示され、x 軸が調整されて信号に関連するタイミング情報を表示します。

ダイナミックデータの属性を受け入れるグラフの例については、`examples\general\graphs` ディレクトリの「Graph Adapting to Attributes」VI を参照してください。

LabVIEW ダイアログボックスにおける機能強化

LabVIEW ダイアログボックスにあるメニューを使用して、LabVIEW の一般的な機能やユーティリティを利用することができます。

新規ボタンをクリックして、**新規**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には、標準の VI テンプレートのリストが表示されます。新規のブランク VI を開くには、**新規**プルダウンメニューから**ブランク VI** を選択するか、**ファイル**→**新規 VI** を選択します。

構成ボタンをクリックすると、Measurement & Automation Explorer (MAX) が起動します。新規の NI-DAQmx チャネルまたはタスクを構成するには、**構成**プルダウンメニューから適切な項目を選択します。

テンプレート VI を使用して新規 VI を作成する

ファイル→**新規**を選択すると**新規**ダイアログボックスが表示されます。新規ダイアログボックスには、一般的な計測アプリケーションを作成するのに使用できる標準テンプレート VI のリストで表示されます。また、**LabVIEW** ダイアログボックスの**新規**ボタンをクリックして、**新規**ダイアログボックスを表示することもできます。

制御器および関数パレットにおける機能強化

制御器および関数パレットでは、以下の機能が強化されました。

- 他のパレットセットまたは形式に変更するには、**制御器**または**関数**パレットのツールバーにある**オプション**ボタンをクリックします。表示される**制御器/関数パレット**ページで、次のオプションを設定することができます。
 - **パレットセット**: 上級および Express パレットセットは、LabVIEW 6.1 での基本、データ集録、デフォルト、およびテスト & 計測パレットセットに代わるものです。デフォルトパレットセットは Express で、**制御器**パレットと**関数**パレットの最上位にある、Express VI や一般的な計測アプリケーションを構築するのに必要なオブジェクトを含むサブパレットが含まれます。Express パレットセットの**すべての制御器**および**すべての関数**サブパレットには、標準の制御器、表示器、VI、および関数がすべて含まれます。上級パレットセットには、標準の制御器、表示器、VI、および関数をすべて含む最上位のサブパレットが含まれます。上級パレットセットの **Express** サブパレットには、Express VI および一般的な計測アプリケーションを構築するのに必要なオブジェクトが含まれます。



メモ Express パレットセットの場合、ツールセットやモジュールはサブパレットを**制御器パレット**や**関数パレット**の最上位にインストールしません。代わりに、サブパレットは**すべての制御器およびすべての関数サブパレット**にインストールされます。上級パレットセットの場合、ツールセットやモジュールはサブパレットを最上位にインストールします。

- **形式**: すべてのパレットの項目は、テキスト、アイコン、またはアイコンとテキストで表示することができます。**標準 (アイコンまたはテキスト)** 形式を選択して、個々のサブパレットの形式は**アイコンおよびテキスト**になっている場合、そのサブパレットの項目はアイコンで表示されます。**標準 (アイコンまたはテキスト)** 形式は、LabVIEW 6.1 以前のバージョンの**標準形式**と同じです。LabVIEW 7.0 の**標準形式**では、メインパレットを含むすべてのパレットが個々のパレットの形式で表示されます。
- **ナビゲーションボタン**: パレットのツールバーにある**検索ボタン**にはテキストラベルがあります。パレットのツールバー上にあるすべてのボタンにテキストラベルを表示したり、ボタンにはラベルを付けないように LabVIEW を構成することができます。
- **パレットのロード**: LabVIEW では、ユーザがマウスやキーボード操作を行っていないときにパレットに関する情報をバックグラウンドでロードしています。それにより、パレット上のオブジェクトを検索する際のパフォーマンスが向上します。ユーザがパレット内を移動する際にパレット情報をロードしたり、LabVIEW の起動時にパレット情報をロードするように LabVIEW を構成することができます。
- **関数パレット上で「ウィンドウタイトル」を使用**: このオプションは、**オプションダイアログボックスのブロックダイアグラム** ページから**制御器/関数パレット** ページに移動しました。
- **一時表示パレットに検索ボタンを表示**: フロントパネルまたはブロックダイアグラムを右クリックして一時的な**制御器**または**関数パレット**を表示すると、その一時表示パレット上に**検索ボタン**が表示されます。一時表示パレットに**検索ボタン**を表示しないようにするには、このチェックボックスをオフにします。
- 制御器や VI をそれぞれ**ユーザ制御器パレット**と**ユーザライブラリパレット**に追加するには、labview¥user.lib ディレクトリに保存します。制御器や VI をそれぞれ **Express ユーザ制御器パレット**と **Express ユーザライブラリパレット**に追加するには、labview¥user.lib¥_express ディレクトリに保存します。
- カスタムパレットセットを作成または編集するには、**ツール→上級→パレットセットを編集**を選択します。
- カスタムパレットセットでは、パレット形式が**アイコンおよびテキスト**のときに各アイコンの下に表示されるテキストラベルを構成することができます。テキストラベルを構成するには、**ツール→上級→パレットセットを編集**を選択し、**パレットセットプルダウンメニュー**か

らカスタムパレットセットを選択して、パレット上のアイコンを右クリックし、ショートカットメニューから**短い名前を編集**を選択します。

- **制御器**および**関数**パレットは、フロントパネルやブロックダイアグラムのウィンドウと同様にサイズ変更することができます。パレットをデフォルトサイズにするには、**パレットサイズを復元**ボタンをクリックします。
- LabVIEW では、パレットの検索において以下の機能が強化されました。
 - パレットが検索モードのときに、**パレットへ戻る**ボタンをクリックすると、検索モードが終了してパレットに戻ります。次に**検索**ボタンをクリックすると、テキストボックスには最後に入力した文字列が入力された状態になっているため、検索結果をダブルクリックして検索結果に戻りたい場合に便利です。
 - 検索結果で、Express VI には左側に水色のマークが表示されます。
 - 検索結果の 2 つ以上のオブジェクトが同じ名前を持つ場合、LabVIEW はオブジェクト名の右側にオブジェクトが括弧付きで示されたパレット名を表示します。たとえば、while ループを検索すると、検索結果として **while ループ << 実行制御 >>** および **while ループ << ストラクチャ >>** が表示されます。
 - パレットが検索モードのときに、パレットの下にある**オプション**ボタンをクリックすると、**名前とキーワード**と**名前のいずれかで**パレットを検索するように LabVIEW を構成することができます。これらのオプションは、LabVIEW 6.1 の**先頭と含む**の各オプションに代わるものです。**名前とキーワード**で検索すると、入力したテキストをパレット上のオブジェクト名とパレット上のオブジェクトに関連するキーワードから検索します。**名前**で検索すると、パレット上のオブジェクト名のみを検索します。
 - **制御器**パレットで**検索**ボタンをクリックすると、LabVIEW はデフォルトで制御器のみを検索します。**関数**パレットで**検索**ボタンをクリックすると、LabVIEW はデフォルトで VI と関数のみを検索します。**制御器と関数**ボタンをクリックすると、**制御器**および**関数**パレットのすべてのオブジェクトを検索することができます。
- パレットのツールバーにある**1 つ上のパレットへ戻る**ボタンをクリックしてマウスボタンを押し続けると、現在のサブパレットへのパスにある各サブパレットを含むショートカットメニューが表示されます。ショートカットメニューでサブパレット名を選択して、そのサブパレットに移動します。

プロパティダイアログボックス

プロパティダイアログボックスを使用して、制御器や表示器の外観と動作を構成します。フロントパネルで制御器または表示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**を選択すると、そのオブジェクトのプロパティダイアログボックスが表示されます。

自動エラー処理

デフォルトでは、実行時にエラーが起こると、VI は一時停止して、問題がおきているサブ VI もしくは関数をハイライトします。その後、エラーダイアログボックスが表示されます。

VI の自動エラー処理を無効にするには、**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**実行**を選択し、**自動エラー処理を有効**チェックボックスをオフにします。これから作成するすべての VI に対して自動エラー処理を無効にするには、**ツール→オプション**を選択し、上部にあるプルダウンメニューから**ブロックダイアグラム**を選択して、**新規 VI で自動エラー処理を有効にする**チェックボックスをオフにします。自動エラー処理が有効になっている既存の VI でエラーが発生したときに、実行を中断してエラーダイアログボックスを表示するようにしたくない場合は、**自動エラー処理を有効にする**チェックボックスをオフにします。

フロントパネル端子のアイコン



フロントパネルの制御器または表示器を構成して、アイコンまたはデータタイプ端子としてブロックダイアグラムに表示することができます。デフォルトで、フロントパネルオブジェクトはブロックダイアグラムではアイコン端子として表示されます。たとえば、左図に示すアイコン端子はフロントパネルではノブを表します。端子の下にある DBL は、倍精度浮動小数点数のデータタイプを表します。

チェックマークを外してデータタイプ端子を表示するには、端子を右クリックして、ショートカットメニューから**アイコンとして表示**を選択します。アイコン端子を表示するには、ショートカットメニューから再度**アイコンを表示**を選択します。アイコン端子はデータタイプ端子よりも大きいので、データタイプ端子をアイコン端子に変換すると、意図せず他のブロックダイアグラムオブジェクトが隠れてしまうことがあります。

新たに作成するフロントパネルオブジェクトの端子をデータタイプとして表示するように LabVIEW を構成するには、**ツール→オプション**を選択して一番上のプルダウンメニューから**ブロックダイアグラム**を選択し、**フロントパネル端子をアイコンとして配置**チェックボックスのチェックマークを外します。

自動ワイヤルーティング

ユーザが配線する際、LabVIEW は自動的にワイヤのルートを見つけます。ワイヤ上で右クリックしてショートカットメニューから**ワイヤを調整**を選択することで、自動的に既存のワイヤの経路を調整できます。一時的に自動ワイヤルーティングを無効にし、手動配線で経路を決定するには、配線を開始した後に <A> キーを押します。再び <A> キーを押すと、自動ワイヤルーティングを有効にすることができます。配線が終了すると、LabVIEW では再び自動ワイヤルーティングが有効になります。配線を開始した後でも、マウスボタンを押しながら他の端子に配線したりポイント

を設定してからマウスボタンを放すと、自動ワイヤルーティングを一時的に無効にできます。マウスボタンを放すと、自動ワイヤルーティングが再び有効になります。

すべての新しいワイヤに対して自動ワイヤルーティング機能を無効にするには、**ツール→オプション**を選択し、一番上のプルダウンメニューから**ブロックダイアグラム**を選択して、**自動ワイヤルーティング機能を有効にする**チェックボックスのチェックマークを外します。

ストラクチャのサイズを自動的に変更する

ストラクチャ内でストラクチャの枠の近くにオブジェクトを配置または移動すると、オブジェクトが入るようにストラクチャのサイズが変更されます。ストラクチャのサイズを手動で変更する場合、ストラクチャのサイズを内部のオブジェクトより小さくすることはできません。この機能を無効にするには、ストラクチャの枠を右クリックし、ショートカットメニューから**自動拡大**を選択してチェックマークを外します。この動作は、ブロックダイアグラム上に配置されるすべての新規ストラクチャに対し、デフォルトで有効になっています。すべての新しいストラクチャに対して自動サイズ変更機能を無効にするには、**ツール→オプション**を選択し、一番上のプルダウンメニューから**ブロックダイアグラム**を選択して、**自動拡大を有効にしてストラクチャを配置**チェックボックスのチェックマークを外します。

フラットシーケンスストラクチャ

シーケンスストラクチャには、フラットシーケンスストラクチャとスタックシーケンスストラクチャの2種類があります。フラットシーケンスストラクチャには、スタックシーケンスストラクチャと同様、順次実行される1つまたは複数のサブダイアグラム、つまりフレームが含まれています。フラットシーケンスストラクチャは、すべてのフレームを一度に表示し、最後のフレームが実行されるまでフレームを左から右に実行します。フレームの実行後データは各フレームから離れるため、各フレームからフラットシーケンスストラクチャの外にあるノードにデータを渡すことができます。シーケンスローカルを使用せずにブロックダイアグラムが一目でわかるようにするには、フラットシーケンスストラクチャを使用します。

スタックシーケンスストラクチャをフラットシーケンスストラクチャに変換するには、スタックシーケンスストラクチャを右クリックして、ショートカットメニューから**入れ替え→フラットシーケンス**と**置換**を選択します。

NI サンプルファインダ

ヘルプ→サンプルの検索を選択すると NI サンプルファインダが起動します。NI サンプルファインダを使用して、インストールされているサンプル VI や、NI Developer Zone (ni.com/zone) にあるサンプル VI を検索したり参照したりすることができます。

(Mac および UNIX) ウェブ上のサンプル VI を検索したり参照することはできません。

制御器および表示器のデフォルトスタイルを変更する

ユーザが端子を右クリックしてショートカットメニューから**作成→制御器**または**作成→表示器**を選択したときに LabVIEW が作成する制御器または表示器のスタイルを変更するには、**ファイル→VI プロパティ**を選択して上部の**カテゴリ**プルダウンメニューから**編集オプション**を選択します。端子を右クリックしてショートカットメニューから**作成→制御器**または**作成→表示器**を選択したときに LabVIEW が新規 VI で作成する制御器または表示器のスタイルを変更するには、**ツール→オプション**を選択して上部のプルダウンメニューから**フロントパネル**を選択します。

LabVIEW データディレクトリ

LabVIEW をインストールすると、LabVIEW が生成するデータファイルの整理や配置に使用できる LabVIEW Data サブディレクトリがインストーラによってオペレーティングシステムのデフォルトファイルディレクトリに作成されます。このディレクトリは、.lvmm ファイルなど、VI が生成するファイルの保存に使用します。別のデフォルトディレクトリを指定するには、**ツール→オプション**を選択し、プルダウンメニューから**パス**を選択します。

NI アプリケーションプログラミングインタフェースのアップデート

ナショナルインストルメンツでは、いくつかのハードウェアアプリケーションプログラムインタフェース (API) をアップデートしました。具体的には、新しい DAQ API として Windows 対応 NI-DAQmx が新たに加われました。それらのアップデートに関する詳細については、『LabVIEW Measurements Manual』および各 API のドキュメントを参照してください。

(Windows) NI-DAQmx VI の使用例については、`examples\DAQmx` ディレクトリを参照してください。

VISA VI および関数の使用例については、`examples\instr\visa.11b` および `examples\instr\smp1ser1.11b` を参照してください。

DAQ アシスタント

(Windows) DAQ アシスタントは、LabVIEW 7.0 以降で使用する NI-DAQmx の計測タスク、チャンネル、およびスケール構成用のグラフィックインタフェースです。DAQ アシスタントを使用して、NI-DAQmx でプログラミングする際にデバイスを構成します。DAQ アシスタントは、LabVIEW ダイアログボックス、DAQmx タスクおよび DAQmx グローバルチャンネル I/O 制御器、および「DAQ アシスタント」Express VI から起動することができます。

計測器 I/O アシスタント

(Windows) 計測器との通信は通常計測器ドライバを使って行います。ドライバがない場合は、計測器 I/O アシスタントを使って、シリアル、イーサネット、または GPIB インタフェースを使用する計測器と通信し、応答をグラフィックに構文解析することができます。計測器 I/O アシスタントは、計測器の通信を順序だったステップに構成します。計測器 I/O アシスタントを起動するには、ブロックダイアグラムに「I/O アシスタント」Express VI を配置するか、ブロックダイアグラム上で「I/O アシスタント」Express VI のアイコンをダブルクリックします。

I/O 制御器

(Windows) I/O パレットには、モーションデバイス、FieldPoint デバイス、および NI-DAQmx の I/O 制御器があります。FieldPoint I/O ポイント制御器を使用して、MAX で作成し構成した FieldPoint 項目にアクセスします。MAX を使用して構成したモーションデバイスにアクセスするには、モーションリソース制御器を使用します。MAX で構成したすべてのアイテムが I/O 制御器のプルダウンメニューにオプションとして表示されます。

(Windows) DAQmx 名前制御器パレットにある制御器を使用して、MAX および DAQ アシスタントで構成したタスク、グローバルチャネル、物理チャネル、端子、スケール、デバイス、およびスイッチリソースにアクセスします。NI-DAQmx VI の使用例については、[examples\DAQmx](#) ディレクトリを参照してください。

計測器ドライバのアップデート

ナショナルインスツルメンツでは頻繁に計測器ドライバをアップデートしていますので、Instrument Driver Network (ni.com/idnet) から新バージョンのドライバをダウンロードすることをお勧めします。アップデートした計測器ドライバは LabVIEW 7.0 でコンパイルされ、`7\labview` ディレクトリにインストールされます。IVI ドライバをアップデートすることによって、Initialize、Initialize with Options、Close などの関数が誤ってエラーを報告する問題が修正されます。

I/O におけるその他の機能向上

LabVIEW では、I/O において以下の機能が強化されました。

- 「VISA Wait on Event」関数は、同期または非同期で I/O 操作を実行します。デフォルトでは、非同期で実行するようになっています。指定したイベントのオカーレンスで同期的に待機するには、ノードを右クリックしてショートカットメニューから**同期 I/O** を選択します。LabVIEW 6.1 以前のバージョンでは、「VISA Wait on Event」は同期的にイベントを待機し、「VISA Wait on Event Async」VI が非同期

でイベントを待機するようになっていました。LabVIEW 7.0 の「VISA Wait on Event」関数は、LabVIEW 6.1 以前の「VISA Wait on Event Async」VI にとって代わるものです。

- VISA USB VI および関数は、USB デバイスの制御に使用します。
- 今バージョンの ReceiveSetup と SendSetup には、**バイトカウン**ト出力は含まれていません。
- GPIB CIC イベントの Recv CIC State プロパティのデータタイプは、数値からブールに変わりました。
- 「プロパティノード (Property Node)」を使用して DAQ プロパティにアクセスする場合、「プロパティノード」のショートカットメニューに**変更**サブメニューが表示されます。プロパティ情報を取得するには、「プロパティノード」を右クリックし、ショートカットメニューから**変更→読み取りに変更**を選択します。プロパティ情報を設定するには、「プロパティノード」を右クリックし、ショートカットメニューから**変更→書き込みに変更**を選択します。プロパティが読み取り専用の場合、ショートカットメニューの**書き込みに変更**はグレー表示になります。プロパティをデフォルト値にリセットするには、「プロパティノード」を右クリックして、ショートカットメニューから**変更→デフォルト値**を選択します。プロパティをデフォルト値に設定不可能な場合は、ショートカットメニューの**デフォルト値**がグレー表示になります。

.NET 関数

.NET 関数を使用して、.NET オブジェクトを作成したり .NET オブジェクトのプロパティやメソッドを設定します。.NET 関数を使用するためには、Microsoft .NET Framework がインストールされている必要があります。.NET およびそのフレームワークのインストールの詳細については、MSDN (Microsoft Developer Network) ウェブサイトを参照してください。LabVIEW では、delegate や Windows Forms を使用したイベントをサポートしていません。

Mac OS X のサポート

LabVIEW 7.0 では Mac OS X をサポートしています。Mac OS X で DAQ I/O を実行するには、LabVIEW Real-Time モジュールを使用する必要があります。Mac OS X での I/O 操作に関する詳細は、『LabVIEW RT Module for Mac OS X User Manual Addendum』を参照してください。Mac OS 9.x 以前のバージョンで実行する LabVIEW 用にコンパイルされた CIN は、Mac OS X 上の LabVIEW とは互換性がありません。Mac OS X 用に CIN を再構築するには、『Using External Code in LabVIEW』マニュアルの Chapter 3 「CINs」の『Compile on Mac OS X』のセクション、あるいは cintools¥Project Builder Files ディレクトリにある ReadMe(Project Builder).rtf ファイルを参照してください。

LabVIEW 環境の機能向上

LabVIEW 環境で向上された機能として、短縮メニュー、フロントパネル端子上のアイコン、自動ツール選択機能などがあります。

LabVIEW 環境のそれらの機能をカスタマイズするには、**ツール→オプション**を選択して、一番上のプルダウンメニューから **7.0 での新しい点と変更点**を選択します。以下のセクションでは、各オプションを **7.0 での新しい点と変更点**のページに記載されている順に説明しています。

新規ブロックダイアグラムオプション

- **フロントパネル端子をアイコンとして配置** : ユーザが作成した新しいフロントパネルオブジェクトの端子をアイコンで表示します。新しいフロントパネルオブジェクトの端子をデータタイプで表示する場合は、このチェックボックスをオフにします。既存の端子をアイコンまたはデータタイプで表示するように構成するには、本書の「**フロントパネル端子のアイコン**」のセクションを参照してください。
- **新規 VI で自動エラー処理を有効にする** : ユーザが作成する新規 VI で、VI の実行時にエラーが発生すると、実行を中断し、エラーが発生したサブ VI または関数をハイライトして、エラーダイアログボックスを表示することにより、発生したエラーを自動的に処理します。すべての新規のブランク VI でこの自動エラー処理機能を無効にするには、このチェックボックスをオフにします。既存の VI の自動エラー処理機能を無効にする方法の詳細については、本書の「**自動エラー処理**」のセクションを参照してください。
- **自動エラー処理を有効にする** : 自動エラー処理が有効になっている既存の VI で、VI の実行時にエラーが発生すると、実行を中断し、エラーが発生したサブ VI または関数をハイライトして、エラーダイアログボックスを表示することにより、発生したエラーを自動的に処理します。自動エラー処理が有効になっている既存の VI でエラーが発生したときに、実行を中断してエラーダイアログボックスを表示するようには、このチェックボックスをオフにします。既存の VI の自動エラー処理機能を無効にする方法の詳細については、本書の「**自動エラー処理**」のセクションを参照してください。
- **自動ワイヤルーティングを有効にする** : 配線時に自動的にワイヤの経路が検出されます。LabVIEW は、ループやストラクチャなど、ブロックダイアグラム上にある既存のオブジェクトを避けてワイヤを配線します。また、配線が複雑にならないように、できる限り直線的にワイヤを配線します。すべての新規ワイヤでこの自動ワイヤルーティング機能を無効にするには、このチェックボックスをオフにします。自動ワイヤルーティング機能を無効にした場合は、個々のワイヤを右クリックしてショートカットメニューから **ワイヤを調整**を選択すると、自動的に経路を調整することができます。自動ワイヤルーティング機能を一時的に無効にするには、ワイヤの最初のポイントをクリッ

くした後に <A> キーを押します。自動ワイヤルーティング機能を一時的に無効にする方法の詳細については、本書の「[自動ワイヤルーティング](#)」のセクションを参照してください。

- **帰還ワイヤ内にフィードバックノードを自動的に挿入** : サブ VI、関数、またはサブ VI や関数のグループの出力を同じ VI、関数、またはグループの入力に配線した場合に、For ループまたは While ループに「フィードバックノード」が挿入されます。自動的に「フィードバックノード」を挿入したくない場合は、このチェックボックスをオフにします。「フィードバックノード」の詳細については、本書の「[フィードバックノード](#)」のセクションを参照してください。
- **サブ VI を拡張可能アイコンとして配置** : ユーザがブロックダイアグラム上に配置したサブ VI を拡張可能ノードとして表示します。拡張可能ノードを使用することにより、配線やブロックダイアグラムの文書化が容易になります。このオプションは、Express VI には作用しません。拡張可能ノードとしてブロックダイアグラム上にすでに配置したサブ VI を表示する方法については、本書の「[サブ VI を拡張可能ノードとして表示する](#)」を参照してください。
- **ただちに Express VI を構成** : ブロックダイアグラムに Express VI を配置すると、ただちに構成ダイアログボックスが表示されます。ブロックダイアグラムに Express VI を配置したときただちに構成ダイアログボックスが表示されるようにしたくない場合は、このチェックボックスをオフにします。Express VI の詳細については、本書の「[Express VI](#)」のセクションを参照してください。
- **自動拡大機能を有効にしてストラクチャを配置** : ブロックダイアグラムに新たに配置するストラクチャにおいて、ストラクチャの境界付近にオブジェクトを配置したり移動したりするためのスペースを確保するため、ストラクチャのサイズが変更されます。すべての新規ストラクチャでこの自動サイズ調整機能を無効にするには、このチェックボックスをオフにします。既存のストラクチャの自動サイズ調整機能を無効にする方法の詳細については、本書の「[ストラクチャのサイズを自動的に変更する](#)」のセクションを参照してください。
- **不良ワイヤに赤い X 印を表示** : 不良ワイヤは、黒い破線の中央に赤の x 印が表示されます。不良ワイヤを赤の x のない黒の破線で表示するには、このチェックボックスをオフにします。不良ワイヤの表示方法の詳細については、本書の「[不良ワイヤにおける機能強化](#)」のセクションを参照してください。

新規フロントパネルオプション

- **新規 VI の制御器スタイル** : 端子を右クリックして、ショートカットメニューから**作成→制御器**または**作成→表示器**を選択したときに、新規 VI で作成される制御器や表示器のスタイルを変更します。既存の VI で作成される制御器や表示器のスタイルを変更する方法の詳細については、本書の「[制御器および表示器のデフォルトスタイルを変更する](#)」のセクションを参照してください。

新規その他のオプション

- **短縮メニューを使用：(Windows および UNIX)** 最も一般的なメニュー項目および最近使用したメニュー項目のみをデフォルトで表示します。すべての項目を表示するには、メニューの下部の矢印をクリックします。デフォルトですべてのメニュー項目を表示するには、このチェックボックスをオフにします。
- **自動ツール選択をオンにロック：**<Tab> キーまたは <Shift-Tab> キーを押したときも自動ツール選択をオンのままにします。<Tab> キーを押したときに自動ツール選択を無効にし、<Shift-Tab> キーを押したときに自動ツール選択の切り替えを行うように LabVIEW を構成するには、このチェックボックスをオフにします。自動ツール選択機能を有効または無効にする方法の詳細については、本書の「[自動ツール選択での機能向上](#)」のセクションを参照してください。
- **ジャストインタイムアドバイスを有効にする：**LabVIEW で特定の操作を行った際にジャストインタイムアドバイスウィンドウを表示します。このダイアログボックスは、LabVIEW の旧バージョンと現バージョンの間で変更があったときにユーザに警告を発します。LabVIEW の使用中ジャストインタイムアドバイスウィンドウが表示されないようこの機能を無効にするには、このチェックボックスをオフにします。
- **LLB ファイルに Windows エクスプローラを有効にする：(Windows 2000/XP/Me/98)** .llb ファイルに対し Windows エクスプローラを有効にして、.llb ファイルをダブルクリックするとライブラリの内容が表示されるようにします。ライブラリ内のファイルは、開いたり、移動、コピー、名前の変更、削除を行うことができ、ライブラリ内の VI を右クリックしてショートカットメニューから**最上位**を選択すると、その VI を最上位 VI としてマークすることができます。このチェックボックスをオフにすると、.llb ファイルをダブルクリックしたときにライブラリ内のすべての最上位 VI が開きます。

6.1 から変更されたデフォルト

- **ダイアグラムからフロントパネル端子を削除／コピー：**ブロックダイアグラム上で端子を選択して <Delete> キーを押すことにより、フロントパネルオブジェクトを削除することができます。また、このオプションは、ブロックダイアグラムで端子を新しい位置にドラッグしている間に <Ctrl> キーを押すことにより、ブロックダイアグラムにあるフロントパネル端子をコピーすることもできます。**(Mac OS)** では <Option> キー、**(Sun)** では <Meta> キー、**(Linux)** では <Alt> キーを使用します。
ブロックダイアグラムでフロントパネル端子を削除またはコピーしたくない場合は、チェックボックスをオフにします。
- **ワイヤの接点でドットを表示：**ワイヤが枝分かれする部分にドットを表示して、ワイヤの接合点と接続されていないワイヤの交差部分を区別しやすくします。ドットなしでワイヤの接合点を表示するには、このチェックボックスをオフにします。

オプションダイアログボックスに新たに追加されたページ

- **アライメントグリッド**: このページで、グリッドのサイズ、コントラスト、およびスタイルなど、すべての新規 VI でのグリッドオプションを設定します。アライメントグリッドの使用の詳細については、本書の「アライメントグリッドを使用してオブジェクトを整列する」のセクションを参照してください。
- **制御器/関数/パレット**: このページで、パレットセットを選択したり、その形式を変更したりします。制御器および関数パレットのカスタマイズの詳細については、本書の「制御器および関数パレットにおける機能強化」のセクションを参照してください。

アライメントグリッドを使用してオブジェクトを整列する

フロントパネル上でアライメントグリッドを有効にし、オブジェクトを配置したときに整列するようにするには、**操作→パネル上のアライメントグリッドを有効**を選択します。アライメントグリッドを無効にし、グリッドを表示してオブジェクトを手動で整列させる場合は、**操作→パネル上のアライメントグリッドを無効**を選択します。また、<Ctrl-#> キーを押して、アライメントグリッドを有効または無効にすることもできます。フランス語のキーボードでは、<Ctrl-~> キーを押してアライメントグリッドを有効または無効にします。

(Mac OS) では <Command-*> キー、(Sun) では <Meta-#> キー、(Linux) では <Alt-#> キーを使用します。

また、ブロックダイアグラムでもアライメントグリッドを使用することができます。

すべての新規 VI でグリッドオプションを設定するには、**ツール→オプション**を選択して、上部のプルダウンメニューから**アライメントグリッド**を選択します。現在の VI でグリッドのサイズを設定するには、**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**編集オプション**を選択します。

自動ツール選択での機能向上

LabVIEW では、自動ツール選択機能において以下の機能が強化されました。

- <Tab> キーを使用して**ツールパレット**で最も頻繁に使用される 4 種類のツール間で切り替えるようにするには、**ツールパレットの自動ツール選択**ボタンをクリックして、自動ツール選択機能を無効にします。<Shift-Tab> キーを押すか、**自動ツール選択**ボタンをクリックして、自動ツール選択をもう一度有効にできます。また、**ツールパレット**で手動でツールを選択して、自動ツール選択機能を無効にすることもできます。自動ツール選択機能を再び有効にするには、<Tab> キーまたは <Shift-Tab> キーを押すか、あるいは**ツールパレットの自動**

ツール選択ボタンをクリックします。自動ツール選択機能が無効になっている場合、スペースバーを押すと、最も頻繁に使用する次のツールに切り替えることができます。

- <Tab> キーを押したときに自動ツール選択を無効にし、<Shift-Tab> キーを押したときに自動ツール選択の切り替えを行うように LabVIEW を構成することができます。そのように構成するには、**ツール→オプション**を選択し、上部のプルダウンメニューから**その他**を選択して、**自動ツール選択をオンにロック**チェックボックスをオフにします。
- 任意のマーカ間隔で数値オブジェクトにマーカを追加するには、<Ctrl> キーを押して、マーカを追加したい位置まで既存のマーカをドラッグします。スライド制御器と表示器で、終了マーカの1つをマーカを追加したい位置にドラッグできます。回転式制御器と表示器で、<Ctrl> キーを押して、終了マーカの1つをマーカを追加したい位置にドラッグします。**(Mac OS)** では<Option> キー、**(Sun)** では<Meta> キー、**(Linux)** では<Alt> キーを使用します。
- LabVIEW は階層ウィンドウで位置決めツールを選択します。スクロールツールに切り替えるには、カーソルがウィンドウ内のアイコン上にないときに<Ctrl-Shift> キーを押します。**(Mac OS)** では<Option-Shift> キー、**(Sun)** では<Meta-Shift> キー、**(Linux)** では<Alt-Shift> キーを使用します。
- フロントパネルまたはブロックダイアグラム上の任意の空白スペースをダブルクリックすると、フリーラベルが作成されます。
- 自動ツール選択機能は、VI の実行中ブロックダイアグラム上で使用可能です。

単位ラベルの編集とカスタマイズ

単位ラベルを編集するには、単位ラベルを右クリックし、ショートカットメニューから**単位文字列の作成**を選択して、**単位文字列の作成**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、単位に接頭辞や指数を適用したり、分母に単位を付けたり、単位文字列に単位を追加したり削除したりすることができます。このダイアログボックスには、LabVIEW で使用可能な単位のリストがアルファベット順あるいは単位タイプのグループごとに表示されます。

単位ラベルを右クリックして、他のフロントパネルのラベルをカスタマイズする場合と同様に、自動サイズ変更を切り替えたり、ラベルを回転することができます。「プロパティノード (Property Node)」を使用して、他のフロントパネルのラベルのプロパティを読み書きする場合と同様に、数値オブジェクトの単位ラベルプロパティ、グラフスケール、およびカラーグラフスケールを読み書きすることができます。

単位ラベルプロパティを書き込む例については、`examples¥general¥controls¥numerics.11b` の「Programmatic Units」VI を参照してください。

旧バージョンで使用できるように VI を保存する

LabVIEW 7.0 では、**ファイル→オプション付き保存**を選択して**旧バージョン用に保存**オプションを選択することにより、LabVIEW 6.1 で使用できるように VI を保存することができます。VI を LabVIEW 6.0 で使用できるように保存するには、その VI を LabVIEW 6.1 で開いて**ファイル→オプション付き保存**を選択し、**旧バージョン用に保存**オプションを選択して保存する必要があります。

印刷およびレポート生成の機能向上

LabVIEW では、印刷およびレポート生成において以下の機能が強化されました。



メモ このセクションで説明した VI は、LabVIEW 開発システムおよびプロフェッショナル開発システムのみでご利用いただけます。

- 「簡易 VI パネルプリントまたは文書化 (Easy Print VI Panel or Documentation)」VI を使用して、フロントパネルまたは VI ドキュメントを印刷したり、フロントパネルまたは VI ドキュメントをレポートに保存することができます。この VI は「パネル印刷 (Print Panel)」VI に代わるものです。「パネル印刷」VI は LabVIEW 7.0 では**関数**パレットに表示されません。
- VI 文書化 VI を使用して、レポートを印刷または保存する VI ドキュメントをカスタマイズします。
- **(Windows)** LabVIEW のデフォルトプリンタを含むコンピュータで使用可能なプリンタのリストを表示するには、「クエリ可能プリンタ (Query Available Printers)」VI を使用します。
- アプリケーションクラスと VI クラスには、印刷オプションをプログラマ的に設定できる新しい VI サーバプロパティがいくつか含まれています。
- **(Windows)** 「レポート印刷 (Print Report)」VI を使用して、デフォルトのプリンタで HTML レポートを印刷します。
- 「レポートフッタテキスト設定 (Set Report Footer Text)」VI の **HTML フッタサイズ**入力を使用して、レポートのフッタに使用する HTML の見出しタグを設定します。
- 「フロントパネル画像をレポートに追加 (Append Front Panel Image to Report)」VI は多形性です。VI にパス、VI リファレンス、または文字列を配線して、レポートに付加するフロントパネルを指定することができます。「フロントパネル画像をレポートに追加」VI の旧文字列のインスタンスは、LabVIEW 6.1 以前の「Append Front Panel Image to Report」VI に似ています。
- 「表をレポートに追加 (Append Table to Report)」VI は多形性で、そのインスタンスは LabVIEW 6.1 以前の「Append Text Table to Report」VI と「Append Numeric Table to Report」VI に代わるものです。

- **VI プロパティ** ダイアログボックスの**印刷オプション**ページには、**印刷するブロックダイアグラムをページに合うようスケール**チェックボックスがあります。印刷のページに合うようにブロックダイアグラムをスケールする場合は、このチェックボックスをオンにします。
- 「**VI をプリンタに印刷**」メソッドをスタンドアロンアプリケーションや共有ライブラリで使用すると、LabVIEW はフロントパネルのみを印刷します。
- **印刷**ダイアログボックスの**パネルを枠で囲む**チェックボックスは、**カスタム詳細**ページから**プリンタ**ページへ移動しました。
- **印刷**ダイアログボックスで、**高レベルのフレームを繰り返し**チェックボックスが**順序**（**ネストの場合は高いレベルから繰り返し**）チェックボックスに変わりました。

その他の環境における機能強化

LabVIEW 環境において以下の機能が強化されました。

- **ウィンドウ**→**パネルを表示**および**ダイアグラムを表示**の各メニュー項目が、それぞれ**ウィンドウ**→**フロントパネルを表示**および**ブロックダイアグラムを表示**に変わりました。
- フロントパネルのウィンドウタイトルは **This.vi フロントパネル**、ブロックダイアグラムのウィンドウタイトルは **This.vi ブロックダイアグラム**になります。ここで **This.vi** は VI の名前です。
- ディレクトリ内からファイル名で VI を検索するには、**ツール**→**デスク上の VI を検索**を選択します。
- `panel.cpp`, line 2978 のような内部 LabVIEW エラーが発生すると、デフォルトデータディレクトリの `lvfailurelog` サブディレクトリにあるログファイルにエラー情報が保存されます。次回 LabVIEW を起動したときに、エラーを調べるようプロンプトするダイアログボックスが表示されます。その時点でエラーを調べることもできますし、**ヘルプ**→**内部エラーを調査**を選択して後ほど調べることができます。エラーを調査すると、LabVIEW は選択したログファイルからエラー情報をナショナルインストルメンツのウェブサイトに送信し、そのウェブサイトをデフォルトのウェブブラウザで表示します。ブラウザに表示されるフォームにはエラー情報が含まれ、考えられるエラーの原因を検索することができます。
- **(Windows XP)** LabVIEW では、現在の Windows XP テーマを使用して、ダイアログ制御器、スクロールバー、メニューなどを表示します。LabVIEW で作成したアプリケーションを Windows XP 上で配布した場合も、そのアプリケーションでは現在の Windows XP テーマを使ってオブジェクトを表示します。
- 位置決めツールをサイズ変更可能なオブジェクト上に移動すると、オブジェクトをサイズ変更できる部分にサイズ変更ハンドルが表示されます。サイズの拡大を垂直または水平方向に限定するには、それぞれオブジェクトの上か下、あるいは右か左のサイズ変更ハンドルを使用します。

- フロントパネルおよびブロックダイアグラムオブジェクトのテキスト入力を終了するには、<Ctrl-Enter> キーを押します。

ツリー制御器を使用する

選択項目の階層リストを表示するには、ツリー制御器を使用します。ツリー制御器に入力する項目は、ノードという項目のグループに分けます。ノードの隣にある展開記号をクリックしてノードを展開し、そのノード内にあるすべての項目を表示します。また、ノードの隣にある記号をクリックして、ノードを格納することもできます。



メモ

ツリー制御器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。VI にツリー制御器が含まれている場合、VI の実行はすべての LabVIEW パッケージで可能ですが、制御器の構成は基本パッケージでは行えません。

ツリー制御器内の項目は、リストボックスと同じ方法で構成します。また、各ノードの隣に表示される記号の種類を変更でき、ツリー制御器内でユーザが項目をドラッグできるかどうかを設定することもできます。

また、ツリー制御器プロパティおよびメソッドを使用して、ツリー制御器をプログラマ的に構成することもできます。ツリー制御器上でユーザが行うイベントの処理には、ツリー制御器イベントを使用します。

ツリー制御器のサンプルについては、`examples\general\controls\Directory Tree Control.llb` の「Directory Hierarchy in Tree Control」VI を参照してください。

サブパネル制御器を使用する

現在の VI のフロントパネル上に他の VI のフロントパネルを表示するには、サブパネル制御器を使用します。たとえば、サブパネル制御器を使用して、ウィザードのような役割を果たすユーザインタフェースを設計することができます。最上位 VI のフロントパネルに**戻る**および**次へ**ボタンを配置し、サブパネル制御器を使用して、ウィザードの各ステップに対して異なるフロントパネルをロードします。



メモ

サブパネル制御器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。VI にサブパネル制御器が含まれている場合、その VI はすべての LabVIEW パッケージで実行することはできますが、ベースパッケージで制御器を構成することはできません。

フロントパネル上にサブパネル制御器を配置した場合、LabVIEW はブロックダイアグラム上の制御器に対するフロントパネル端子を作成しません。代わりに、ブロックダイアグラム上に「インボークノード (Invoke Node)」が作成され、「VI を挿入」メソッドが選択されます。

サブパネル制御器をプログラマ的に変更するには、サブパネルプロパティおよびメソッドを使用します。

サブパネル制御器の使用例については、`examples\general\controls\subpanel.llb`を参照してください。

リング制御器における機能強化

リング制御器を右クリックしてショートカットメニューから**項目を編集**を選択すると、制御器で選択することができるリストに項目を追加することができます。また、リング制御器を右クリックしてショートカットメニューから**未定義の値を許可**を選択することによって、ユーザが制御器に対して定義された項目リストにエントリに関連付けられていない数値を入力できるようにリング制御器を構成することもできます。

実行時にリング制御器に未定義の数値を入力するには、制御器をクリックしてショートカットメニューから**<その他>**を選択し、表示されたデジタル表示に数値を入力して、<Enter> キーを押します。

リング制御器の項目リストを構成する際、各項目に特定の数値を割り当てることができます。項目に特定の数値を割り当てないと、LabVIEW はリスト内の項目の順序に従って、最初の項目に値 0 を割り当て、以降連続する値を割り当てます。特定の数値を割り当てるには、リング制御器を右クリックしてショートカットメニューから**項目を編集**を選択し、**リングプロパティ**ダイアログボックスの**項目を編集**ページの**連続的な値**チェックボックスをオフにします。このダイアログボックスにある表の**値**セクションで、制御器の各項目に対応する数値を変更します。リング制御器の各項目には、固有の数値がついている必要があります。



メモ ユーザが未定義の値を列挙型制御器に入力できるように設定することはできません。また、列挙型制御器内の項目に特定の数値を割り当てることもできません。

リング制御器とコンボボックスの使用例については、`examples\general\controls\ring.llb`の「Ring and Combo Box」VIを参照してください。

コンボボックス制御器を使用する

フロントパネルで繰り返し表示できる文字列のリストを作成するには、コンボボックス制御器を使用します。コンボボックス制御器は、テキスト制御器やメニューリング制御器に類似しています。ただし、コンボボックス制御器の値およびデータタイプは、リング制御器のような数値ではなく、文字列です。

コンボボックス制御器を右クリックしてショートカットメニューから**項目を編集**を選択すると、制御器で選択することができるリストに文字列を追加することができます。デフォルトでは、コンボボックス制御器を使用すると、ユーザはその制御器に定義された文字列のリストにない文字列の値

を入力することができます。コンボボックス制御器を右クリックしてショートカットメニューから**項目を編集**を選択すると、チェックマークが外され、ユーザが未定義の文字列値を制御器に入力できないようになります。

ランタイム時にコンボボックス制御器に文字列を入力すると、LabVIEWは入力された文字で始まる最初の最短の文字列を選択します。

コンボボックス制御器の文字列のリストを構成する場合、各文字列にカスタムの値を指定することができます。カスタムの値を指定すると、フロントパネルのコンボボックス制御器に表示される文字列を、ブロックダイアグラムのコンボボックス端子が返す文字列値とは異なるようにする場合に便利です。コンボボックス制御器を右クリックしてショートカットメニューから**項目を編集**を選択し、**コンボボックスプロパティ**ダイアログボックスの**項目を編集**ページの**値一致ラベル**チェックボックスをオフにします。このダイアログボックスにある表の**値**の列で、制御器の各文字列に対応する値を変更します。

リング制御器とコンボボックスの使用例については、`examples\general\controls\ring.llb`の「Ring and Combo Box」VIを参照してください。

グラフおよびチャートにおける機能強化

LabVIEWでは、グラフとチャートにおいて以下の機能が強化されました。

- グラフ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→Xスクロールバー**を選択すると、グラフまたはチャートの下にスクロールバーを表示することができます。操作ツールを使用してスクロールバーをクリックし、x軸方向にスクロールします。「Xスクロールバー表示」プロパティを使用すると、グラフやチャートのx軸スクロールバーをプログラムの表示することができます。このプロパティは、強度チャートクラスと波形チャートクラスの「スクロールバー表示」プロパティに代わるものです。
- グラフまたはチャートを右クリックして**上級→スケールのレイアウトをリセット**を選択すると、yスケールがプロット領域の左に、xスケールがプロットの下に戻って、スケールマークがリセットされます。
- チャートのxスケールを複製することはできません。
- グラフパレットのフィットズームツールはUndo Zoomツールに代わるものです。フィットズームツールを使用して、グラフやチャートのすべてのxスケールまたはyスケールを自動スケールします。
- デフォルトで、グラフとチャートのプロット凡例は、凡例に表示されている最も長いプロット名の幅に合わせて自動的にサイズ調整されません。この動作を無効にするには、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**凡例を自動サイズ調整**を選択してチェックマークを外します。また、波形グラフクラスと波形チャート

クラスの「凡例を自動サイズ調整」プロパティを使用して、この動作をプログラムの制御することもできます。

- グラフおよびチャートで、x軸は浮動小数点表記、y軸は自動形式を使用するように構成されています。グラフまたはチャート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、**プロパティ**ダイアログボックスを表示して軸の形式を構成します。

VI をフローティングウィンドウで実行する

- フロントパネルをフローティングウィンドウで構成して、常に他のフローティングでないすべての LabVIEW ウィンドウの上に表示されるようにすることができます。そのように構成するには、**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**ウィンドウの外観**を選択して、**カスタマイズ**ボタンをクリックします。**ウィンドウ動作**セクションで**フローティングオプション**を選択します。また、「フロントパネルウィンドウ：動作」プロパティを使用して、フロントパネルをプログラムのフローティングウィンドウとして構成することも可能です。このプロパティは、「Front Panel Window: Is Dialog」プロパティに代わるものです。
- フロントパネルを開いたり閉じたりするには、それぞれ「FP を開く」メソッドと「FP を閉じる」メソッドを使用します。フロントパネルウィンドウを開いたときの状態、たとえばウィンドウを最小化と最大化のどちらにするかなどを設定する場合は、「フロントパネルウィンドウ：開く」プロパティの代わりに「FP を開く」メソッドを使用します。ナショナルインスツルメンツでは、「フロントパネルウィンドウ：開く」プロパティではなく「FP を開く」メソッドを使用することをお勧めします。また、「フロントパネルウィンドウ：状態」プロパティを使用して、すでに開いているフロントパネルウィンドウの状態を設定することもできます。これらのプロパティおよびメソッドの使用例については、`examples\viserver`にある「FrontPanelStates」VI を参照してください。

フロントパネルにおけるその他の機能向上

フロントパネルではさらに以下の機能が強化されています。

- 増分／減分ボタンを表示／非表示にするには、数値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**→**増分／減分**を選択します。ボタンの表示／非表示をプログラムの切り替えるには、「増分／減分 表示？」プロパティを使用することもできます。
- **(Windows, Mac OS X, および UNIX)** 多項目選択リストボックスで <Ctrl> キーを押し、別の項目をクリックして複数の項目を選択します。<Shift> キーを押しながら現在の項目の上下の項目をクリックすると、現在の項目と選択した項目の間のすべての項目が選択されます。**(Mac OS 9.x 以前)** 多項目選択リストボックスで <Shift> キーを押し、別の項目をクリックして複数の項目を選択します。

- **(Windows, Mac OS X, および UNIX)** 0 項目を選択できる選択モードのリストボックスで、<Ctrl> キーを押して現在の項目をクリックすると、その項目の選択を解除できます。**(Mac OS 9.x 以前)** <Shift> キーを押しながら現在の項目をクリックします。
- 複数のフロントパネルオブジェクトを同じサイズに変更するには、フロントパネルのツールバー上にある**オブジェクトをサイズ変更する**プルダウンメニューを使用します。選択されたすべてのオブジェクトのサイズを最大または最小のオブジェクトの幅 / 高さに合わせてたり、選択したすべてのオブジェクトをピクセル単位で特定のサイズに変更することができます。
- ActiveX コンテナを右クリックし、ショートカットメニューから**上級→設計モード**を選択して、VI の編集集中にコンテナを設計モードで表示します。設計モードでは、イベントは生成されずイベント手順は実行されません。デフォルトモードは実行モードで、ユーザと同様にオブジェクトと対話することを意味します。
- **ダイアログ制御器**パレットには、スピン制御器、垂直および水平のスライダ制御器や進行状況バーなど、ダイアログボックスでのみ使用するための新しい制御器や表示器が含まれています。それらの制御器や表示器は、カスタマイズをサポートしていないプラットフォーム特定の描画関数を用いているため、右クリックして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、外観をカスタマイズすることはできません。また、「プロパティノード (Property Node)」や「インボークノード (Invoke Node)」を使ってそれらの制御器や表示器の外観をカスタマイズすることもできません。
ダイアログ制御器パレットにはまた、パス制御器、表、ツリー制御器、および自動的に親オブジェクトの背景色と同じ色で表示される不透明ラベルなども含まれています。
- 回転式制御器または表示器のスケールを回転させるには、いずれかの終了マーカをドラッグします。
- **制御器**パレットの **ActiveX** サブパレットは LabVIEW 7.0 にはありません。ActiveX オブジェクトはすべて別のサブパレットに含まれます。
- **(UNIX)** テキストラベルなど、テキストを含む制御器を編集する場合は、コピーや切り取りの操作を行わなくても、中央のマウスボタンを使用して最後にハイライトされたテキストをそのまま貼り付けることができます。
- 実行時に LabVIEW にテキストを貼り付けた場合、貼り付けたテキストのフォント特性は認識されません。たとえば、黒のテキストを含む文字列制御器に赤のテキストを貼り付けると、文字列制御器に貼り付けられたテキストは黒で表示されます。

イベントを動的に登録して、ユーザイベントを処理する

旧バージョンの LabVIEW では、すべてのイベントの登録は静的に処理されていました。LabVIEW 7.0 では、イベントを動的に登録して、カスタムイベントを作成し名前を付けることができます。

最上位 VI の実行全体を通してではなく、アプリケーションの一部のみでイベント生成が行われるようにしたい場合や、アプリケーションの実行中にイベントを生成する VI や制御器を変更したい場合に、動的登録を使用します。また、動的登録を使用すると、イベントが発生した VI 内のみでなく、サブ VI でもイベントを処理することができます。

イベントを作成して名前を付けることができます。それをユーザイベントといい、ユーザ定義のデータを含んでいます。キューやノーティファイアのように、ユーザイベントを使用すると、アプリケーションの異なる部分が非同期で通信できるようになります。ただし、ユーザイベントはイベントストラクチャとも統合されているため、ユーザインタフェースとプログラマ的に生成したイベントの両方を同じイベントストラクチャで処理することができます。オブジェクトの値を更新してイベントを生成するには、「値 (信号)」プロパティを使用します。

イベントを動的に登録し、ユーザイベントを処理するには、`examples\general\dynamicevents.llb` を参照してください。

サブ VI を拡張可能ノードとして表示する

拡張可能ノードを使用することにより、配線やブロックダイアグラムの文書化が容易になります。拡張可能なノードとしてサブ VI を表示するには、サブ VI を右クリックし、ショートカットメニューから**アイコンとして表示**を選択してチェックマークを外します。

サブ VI をデフォルトで拡張可能ノードとしてブロックダイアグラム上に配置するには、**ツール→オプション**を選択して上部のプルダウンメニューから**ブロックダイアグラム**を選択し、**サブ VI を拡張可能として配置**チェックボックスをオンにします。

不良ワイヤにおける機能強化

LabVIEW では、不良ワイヤにおいて以下の機能が強化されました。

- 不良ワイヤは、中央に赤い x がある黒い破線として表示されます。赤い x の左右いずれかに付いた矢印はデータフローの方向を示し、矢印の色はワイヤを流れるデータのデータタイプを示します。不良ワイヤ上に配線ツールを移動すると、不良の原因に関する情報が**ヘルプ**ウィンドウに表示されます。不良ワイヤを修正するには、ワイヤを右クリックして、ショートカットメニューから**分岐ワイヤを削除**、**分岐ワイヤを作成**、**未接続の配線を削除**、**ワイヤを消去**、**制御器に変更**、**表示器に変更**、**ソースで指標付け使用**、および**ソースで指標付け不使用**などのオプションを選択します。これらのオプションは、ワイヤの不良の原因によって変わります。中央に赤の x がいない黒の破線で不良

ワイヤを表示するには、**ツール→オプション**を選択し、上部プルダウンメニューから**ブロックダイアグラム**を選択して、**不良ワイヤに赤い X 印を表示**チェックボックスをオフにします。

- 端子に配線するとワイヤが壊れる場合は、カーソルが警告マーク付きの糸巻きに変わります。不良ワイヤを配線することもできますが、不良ワイヤを修正しなければ VI は実行できません。

多形性 VI における機能強化

多形性 VI セレクタを使用して、多形性 VI のインスタンスを手動で選択することができます。ブロックダイアグラム上の多形性 VI を右クリックし、ショートカットメニューから**表示項目→多形性 VI セレクタ**を選択して、セレクタを表示します。操作ツールを使用して、多形性 VI セレクタをクリックし、ショートカットメニューからインスタンスを選択します。標準の多形性 VI の場合、多形性 VI またはそのセレクタのショートカットメニュー上にカーソルを移動すると、**ヘルプ**ウィンドウに各インスタンスの説明が表示されます。また、セレクタを右クリックしショートカットメニューから**多形性 VI を検索**を選択することによって、多形性 VI セレクタに関連した多形性 VI ノードを検索することもできます。

作成する多形性 VI の**多形性 VI** ウィンドウを表示すると、多形性 VI またはインスタンス VI のアイコンをブロックダイアグラムに表示するかどうか、および多形性 VI にカーソルをあてたときに多形性 VI やインスタンス VI のコネクタペーンと説明を**ヘルプ**ウィンドウに表示するかどうかを構成することができます。また、ブロックダイアグラムに多形性 VI を配置したときに多形性 VI セレクタをデフォルトで表示するかどうかを構成したり、多形性 VI とそのセレクタのショートカットメニューに**自動**の項目を表示するかどうかも構成することができます。多形性 VI とそのセレクタのショートカットメニューのインスタンスの名前を編集するには、**名前**を編集ボタンをクリックします。

タイムスタンプ制御器およびデータタイプ



タイムスタンプ制御器を使用して、絶対時間を高精度で使用、表示、および保存することができます。左図に示すタイムスタンプデータタイプは、秒の整数部分で 15 桁の精度、秒の小数部分でも 15 桁の精度を正確に保つことが可能です。タイムスタンプデータタイプは、64 ビットの基数を持つ 128 ビット符号付き固定小数点数です。最上位の 64 ビットを 64 ビット整数として解釈することができます。この整数は、世界標準時の 1904 年 1 月 1 日午前 12:00 から経過した秒数を表します。この整数は、1904 年からの経過時間として 2^{63} までの秒数を表すことができます。最下位の 64 ビットは、64 ビット符号なし整数として解釈することができます。この整数は、最上位の 64 ビットで指定した全秒数の後の 2^{-64} 秒の数値を表します。つまり、最上位の 64 ビットは秒数を、最下位の 64 ビットは 1 秒以下の数を表します。この 128 ビットデータタイプは、2 つの 64 ビット成分からなる絶対時間を表しています。タイムスタンプ値は数値制御器で表示することができます。ただし、数値制御器は相対量を表示するのにに対し、タイムスタンプ制御器は絶対量を表示します。

「日付 / 時間を秒で取得 (Get Date/Time In Seconds)」関数または「日付 / 時間を秒に変換 (Date/Time To Seconds)」関数を使用する LabVIEW 6.1 の VI を LabVIEW 7.0 でロードした場合、LabVIEW はブロックダイアグラム上で適切な変換関数を挿入して、タイムスタンプデータを LabVIEW 6.1 の VI で使用できるデータタイプに変換します。



メモ



LabVIEW 7.0 では、**オプション**ダイアログボックスに**時間と日付**ページはありません。LabVIEW は、時間と日付のグローバル設定をオペレーティングシステムから取得します。タイムスタンプ制御器用に時間と日付を設定する場合は、左に示すような、制御器の右にある時間/日付参照ボタンをクリックします。

一般プローブとカスタムプローブ

一般プローブを使用して新しいプローブを作成し、アプリケーションをデバッグします。一般プローブは、ワイヤ上を通過して渡されるデータに関する総合的な情報を表示する VI です。たとえば、VI Refnum プローブは、リファレンスの VI 名、VI パス、および 16 進数値に関する情報を返します。また、一般プローブを使用すると、ワイヤを通過するデータに基づいて応答することができます。たとえば、エラークラスタにはエラープローブを使用して、エラーのステータス、コード、および説明を受け取ったり、エラーや警告が発生したときに VI を一時停止するかどうか、あるいは特定のエラーコードを受け取ったときに VI を一時停止するかかななどを指定することができます。ワイヤを右クリックして、ショートカットメニューから**カスタムプローブ**を選択し、一般プローブを選択します。右クリックしたワイヤのデータタイプと一致するプローブのみがショートカットメニューに表示されます。使用する一般プローブを選択すると、そのプローブがそのデータタイプのデフォルトプローブとなり、ユーザがワイヤを右クリックしてショートカットメニューから**プローブ**を選択すると、そのプローブがロードされます。

一般プローブのサンプルについては、`examples¥general¥probes.llb`にある「Using Supplied Probes」VI を参照してください。

ワイヤ上を流れるデータをプローブする際により自由に制御できるようにするには、**カスタムプローブ**を作成します。既存のプローブに基づいてカスタムプローブを作成するか、新規にプローブを作成します。ワイヤを右クリックしてショートカットメニューから**カスタムプローブ**→**新規**を選択すると、**新規プローブを作成**ウィンドウが表示されます。「新規プローブを作成」ウィンドウでは、右クリックしたワイヤのデータタイプに基づき、順を追ってカスタムプローブを作成することができます。

数値をフォーマットする

デフォルトで、LabVIEW は数字を計算機のように表示し保存します。数値制御器または表示器は、自動的に指数表記に切り替わる前に、最大 6 桁を表示します。指数表記に切り替わる前に LabVIEW が表示する桁数は、**プロパティダイアログボックスの形式と精度** ページで構成できます。

また、パーセントコード形式を使用して、LabVIEW における数字の表示方法を指定することもできます。

波形の数値データの有効形式

波形のデータ (Y) 成分には、整数など、任意の数値データタイプの配列を使用できます。波形データタイプ端子の内部は、データが浮動小数点数の場合はオレンジ色、整数の場合は青になります。

ピクチャ制御器における機能強化

LabVIEW では画像の修正機能において以下の点が強化されました。

- 次の VI で、いくつかのパラメータが**画像データ**クラスタパラメータに変更されました。「平坦化ピクスマップ描画 (Draw Flattened Pixmap)」、「ピクスマップを平坦化 (Flatten Pixmap)」、「BMP ファイル読み取り (Read BMP File)」、「JPEG ファイル読み取り (Read JPEG File)」、「PNG ファイル読み取り (Read PNG File)」、「非平坦化ピクスマップ (Unflatten Pixmap)」、「BMP ファイル書き込み (Write BMP File)」、「JPEG ファイル書き込み (Write JPEG File)」、「PNG ファイル書き込み (Write PNG File)」
- 以下の VI は新規 VI です。
 - 画像にマスクを適用するには、「マスク作成 (Create Mask)」VI を使用します。この VI は、画像をピクチャ表示器に書き込む前に画像の特定の色を透明にしたい場合に役に立ちます。
 - ファイルに保存するなど、画像に対して特定のタスクを実行できるようにするため、ピクチャデータタイプを**画像データ**クラスタに変換するには、「ピクチャからピクスマップに変換 (Picture to Pixmap)」VI を使用します。
 - 画像全体ではなくソース画像のサブセットを取得するには、「画像サブセット取得 (Get Image Subset)」VI を使用します。
- 「ピクスマップを平坦化」VI には、ピクスマップにマスクを適用するための**マスク**入力と、ピクスマップのカラーテーブルを定義するための**カラー**入力があります。
- 「ピクスマップを非平坦化」VI には、各ピクセルのマスク情報を記述した**マスク**出力と、ピクスマップのカラーテーブルを記述した**カラー**出力があります。
- 「平坦化ピクスマップ描画 (Draw Unflattened Pixmap)」VI は、LabVIEW 6.1 の「1 ビットピクスマップ描画 (Draw 1-bit Pixmap)」、「4 ビットピクスマップ描画 (Draw 4-bit Pixmap)」、「8 ビット

ピクスマップ描画 (Draw 8-bit Pixmap)」、および「True-Color
ピクスマップ描画 (Draw True-Color Pixmap)」VI に代わるもの
です。「非平坦化ピクスマップ描画 (Draw Unflattened
Pixmap)」VI には、ピクチャにマスクを適用するための**マスク**入力
があります。

- 以下の VI では、**黒描画?** 入力が**ペン**入力に変わっています。「弧描
画 (Draw Arc)」、「円を半径で描画 (Draw Circle by Radius)」、
「ライン描画 (Draw Line)」、「複数ライン描画 (Draw Multiple
Lines)」、「楕円描画 (Draw Oval)」、「ポイント描画 (Draw
Point)」、「四角形描画 (Draw Rect)」、「丸四角形描画 (Draw
Round Rect)」**ペン**入力を使用して、ピクチャを描画するのに使用する
ペンの幅とスタイルを設定します。
- 「四角形にテキスト描画 (Draw Text in Rect)」VI および「ポイント
にテキスト描画 (Draw Text at Point)」VI では、**黒描画?** 入力が**テ
キスト方位**入力に変わりました。**テキスト方位**を使用して、入力する
テキストの方向を設定します。
- 「テキスト境界取得 (Get Text Rect)」VI には**テキスト方位**入力があり、
それを使用して VI がテキストの境界四角形を取得する前にテキ
ストの方向を指定します。
- 以下の VI では、VI へのパスを配線しないと、LabVIEW はファイル
ダイアログボックスを表示してファイルを選択できるようにします。
「BMP ファイル書き込み (Write BMP File)」、「JPEG ファイル書き込
み (Write JPEG File)」、「PNG ファイル書き込み (Write PNG File)」
- 以下の VI サーバメソッドでは**画像データ**クラスタを使用します。
「パネル画像取得」(VI)、「VI アイコンを画像データとして取得」
(VI)、「VI アイコンを画像データから取得」(VI)、「画像を取得」
(制御器)、「端子画像を取得」(制御器)
- 「Set VI Icon」メソッドは「VI アイコンをファイルから取得」に名前
が変更されました。
- ピクチャクラスには、画像をプログラマ的に修正するための新しい
VI サーバプロパティがいくつか含まれています。
- 「JPEG ファイル読み取り (Read JPEG File)」VI は、JPEG ファイ
ルを 24 ビット画像として読み取ります。

これらの強化された機能の例については、examples¥picture¥
pictct1.llb を参照してください。

VI でのカーソルの外観を変更する

カーソル VI を使用して、VI のフロントパネル上でのカーソルの外観を変
更します。例えば、VI がデータの集録や解析を行っていてユーザ入力を受
け入れられない場合は、カーソルを砂時計に変更することができます。
VI でのデータの集録や解析が終了してユーザ入力を受け付けるようにな
ると、砂時計に変更したカーソルをデフォルトのカーソルへ戻すことがで

きます。カーソルは、システムカーソルや LabVIEW カーソルに変更したり、ファイルから新たに作成することもできます。



メモ カーソル 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で利用可能です。

カーソルの外観を変更する例については、`examples¥cursors¥CursorUtilities.llb` を参照してください。

デジタル波形データタイプおよびデジタルデータタイプ

デジタル波形を表すには、デジタル波形データタイプを使用します。デジタル波形でデジタルデータを表すには、デジタルデータタイプを使用します。デジタル波形をデジタルデータ表示器に配線して、デジタル波形のサンプルと信号を表示します。

デジタル波形とデジタルデータに対して操作を実行するには、デジタル波形 VI および関数を使用します。デジタルデータ表示器には、行と列に配列されたデジタルデータが表示されます。デジタルデータ表示器を使用して、デジタル波形を作成したり、デジタル波形からデジタルデータを抽出したりします。デジタル波形データタイプには、デジタル波形のデータ、開始時刻、 Δx 、および属性が含まれます。**デジタル波形**パレットにある「波形作成 (Build Waveform)」関数を使用して、デジタル波形を作成します。デジタル波形をデジタル波形グラフに配線すると、グラフはタイミング情報とデジタル波形データに基づいて自動的に波形をプロットします。

フィードバックノード

サブ VI、関数、またはサブ VI および関数のグループの出力を同じ VI、関数、またはグループの入力に配線した場合、For ループまたは While ループに「フィードバックノード (Feedback Node)」が挿入されます。「フィードバックノード」は、シフトレジスタと同様に、ループが繰り返しを完了するとデータを保存し、その値をループの次の繰り返しに渡して、データタイプを転送します。「フィードバックノード」を使用すると、ループで必要以上に長いワイヤを使用しなくて済みます。「フィードバックノード」の矢印は、ワイヤのデータフローの方向を示します。

フィードバックノードは、**関数**パレットから選択して For ループまたは While ループに配置することもできます。

フィードバックノードが自動的に挿入されるようにしたくない場合は、**ツール**→**オプション**を選択し、上部のプルダウンメニューから**ブロックダイアグラム**を選択して、**帰還ワイヤ内にフィードバックノードを自動的に挿入**チェックボックスをオフにします。

「フィードバックノード」の使用例については、`examples¥general¥structs.llb` の「Feedback Node Build Array」VI を参照してください。

ループ内でシフトレジスタをトンネルに入れ替える

ループの繰り返しから次の繰り返しに値を転送する必要がなくなった場合には、シフトレジスタを右クリックしてショートカットメニューから**トンネルと置換**を選択し、シフトレジスタをトンネルに入れ替えます。ループの繰り返しから次の繰り返しに値を転送する場合は、トンネルを右クリックしてショートカットメニューから**シフトレジスタと置換**を選択し、トンネルをシフトレジスタに入れ替えます。

ブロックダイアグラムにおけるその他の機能強化

ブロックダイアグラムではさらに以下の機能が強化されています。

- 厳密に類別化された制御器 refnum は、同じ種類のデータの制御器 refnum のみを受け入れます。
- ダイアグラム上の任意のオブジェクトを右クリックし、ショートカットメニューから**ソースパレット**を選択して、サブパレットから同様のオブジェクトにアクセスすることができます。**ソース**はブロックダイアグラムオブジェクトを含むサブパレット名です。ノードの端子を右クリックすると、2つのサブパレットメニューが表示されることがあります。1つはそのノードのパレット、もう1つはよく使用されるデータタイプのソースパレットです。
- サブVIの推奨端子を配線しないと、LabVIEWはそのサブVIを含むVIに対し一切の警告を発生しません。
- ケースストラクチャ、フラットシーケンスストラクチャ、スタックシーケンスストラクチャ、Forループ、またはWhileループを右クリックして、ショートカットメニューから**入れ替え**を選択し、ストラクチャを別の同様のストラクチャに入れ替えます。
- ピクチャまたは refnum 定数を作成するには、フロントパネルのピクチャまたは refnum 制御器をブロックダイアグラムにドラッグします。フロントパネル制御器をブロックダイアグラムにドラッグしたときの制御器の値を持つ定数が作成されます。refnum 定数の値は Not A Refnum です。また、ピクチャまたは refnum 定数を作成するには、ブロックダイアグラム端子を右クリックしてショートカットメニューから**作成→定数**を選択する方法もあります。制御器 refnum 制御器または端子から定数を作成する場合、LabVIEWはクラス指定子定数を作成します。
- ワイヤのブランチを右クリックしショートカットメニューから**分岐ワイヤを削除**を選択して、ワイヤのブランチを削除します。
- 文字列定数では、自動サイズ調整が有効になっている場合は <Shift-Enter> キーを押して無効にします。自動サイズ調整が無効になっている場合は、<Shift-Enter> キーを押して定数にスクロールバーを表示することができます。自動サイズ調整が無効になっていてスクロールバーが表示されていない場合、ユーザが定数にテキストを入力すると、定数は縦方向にサイズを調整します。文字列定数その内容より小さくサイズ変更すると、スクロールバーが表示されます。

スクロールバーが表示されている場合に定数をその内容よりも大きくサイズ変更すると、スクロールバーは表示されなくなります。

- **関数**パレットには、**装飾体**サブパレットが含まれます。
- 「プロパティノード (Property Node)」または「インボークノード (Invoke Node)」を右クリックすると、**VI サークラスを選択**、**VISA クラスを選択**、**ActiveX クラスを選択**などのショートカットメニュー項目が最上位の**クラスを選択**メニューの下に表示されます。

VI からデータを電子メールで送信する

SMTP 電子メール VI を使用して、データやファイルの添付を含む電子メールを送信ができます。この電子メール送信には、簡易メール転送プロトコル (SMTP) が使用されます。

SMTP 電子メール VI の使用例については、`examples\comm\smtpex.llb` を参照してください。

DataSocket における機能強化

LabVIEW では、DataSocket において以下の機能が強化されました。

- 最新の値だけでなく、DataSocket サーバにパブリッシュされたすべての値を読み取るには、クライアントでデータをバッファ処理する必要があります。また、DataSocket サーバマネージャを使用して、サーバ側のバッファ処理を構成する必要があります。サーバ側のバッファ処理の詳細については、「DataSocket ヘルプ」を参照してください。
- DataSocket 接続をプログラムの的に開閉するには、「DataSocket を開く (DataSocket Open)」および「DataSocket を閉じる (DataSocket Close)」関数を使用します。「バッファ最大バイト」、「バッファ最大パケット」、「バッファ利用率 (バイト)」、「バッファ利用率 (パケット)」、「接続状況」、および「URL」などの DataSocket プロパティを使用して、DataSocket のバッファ処理の制約条件を構成したり、DataSocket 接続ステータスを検証したり、DataSocket URL を読み取ることができます。
「DataSocket 書き込み (DataSocket Write)」および「DataSocket を閉じる (DataSocket Close)」関数の **ms タイムアウト (0)** 入力を使用して、関数が保留している操作を完了させるまでに待機する時間 (ミリ秒) を指定します。

UDP における機能強化

「UDP マルチキャストオープン (UDP Multicast Open)」VI を「UDP オープン (UDP Open)」関数の代わりに使用して、読み取り、書き込み、もしくは UDP マルチキャストデータの読み取りと書き込みが可能な接続を開きます。マルチキャスト IP アドレスは、マルチキャストグループを定義します。マルチキャスト IP アドレスは、224.0.0.0 ~ 239.255.255.255 の範囲になります。クライアントがマルチキャストグ

ループに参加する場合、クライアントはグループのマルチキャスト IP アドレスにサブスクライブします。マルチキャストグループにサブスクライブすると、クライアントはマルチキャスト IP アドレスに送信されたデータを受け取ります。

UDP マルチキャストの使用方法については、『Using LabVIEW with TCP/IP and UDP』アプリケーションノートを参照してください。UDP マルチキャストのサンプルについては、`examples\comm\UDP.11b`にある「UDP Multicast Receiver」VI および「UDP Multicast Sender」VI を参照してください。

ActiveX イベントを処理する

ActiveX イベント VI は関数パレットに表示されません。ActiveX イベントをアプリケーションで使用するには、「レジスタイベントコールバックノード (Register Event Callback node)」を使用して ActiveX イベントを登録し、処理する必要があります。

「レジスタイベントコールバックノード」の使用例については、`examples\comm\axevent.11b` を参照してください。

通信、VI サーバ、およびリモートフロントパネルにおけるその他の機能強化

LabVIEW では、VI サーバとリモートフロントパネルにおいて以下の機能が強化されました。

- 「接続情報」と「接続を確認」のアプリケーションメソッドを使用して、VI サーバ接続が応答しているかどうかをチェックします。これらのメソッドを使用しないと、LabVIEW は接続をチェックしないため、動作は旧バージョンの LabVIEW と同じになります。

「接続情報」および「接続を確認」メソッドの使用例については、`examples\viserver\connpolling.11b` を参照してください。

- LabVIEW 6.1 と 7.0 のブラウザプラグインは同時に実行することができますが、通信するリモートフロントパネルサーバは異なります。LabVIEW 6.1 のブラウザプラグインでは LabVIEW 6.1 の VI のみが表示され、LabVIEW 7.0 のブラウザプラグインでは LabVIEW 7.0 の VI のみが表示されます。HTML の OBJECT/EMBED タグによって、ブラウザがロードするプラグインが決まります。LabVIEW の ActiveX コントロールでは、OBJECT タグで指定した CLASSID によってロードするプラグインが決まります。LabVIEW 6.1 のブラウザプラグインの CLASSID と LabVIEW 7.0 のブラウザプラグインの CLASSID は異なります。Netscape ブラウザプラグインの場合は、EMBED タグで指定した MIME タイプによりロードするプラグインが決まります。LabVIEW 6.1 で使用する構文に関する詳細については、LabVIEW 6.1 の『LabVIEW ヘルプ』を参照してください。

- フロントパネルにいくつものチャートを含む VI など、データ量の多い VI をエクスポートして、リモートで表示したり制御したりすることができます。
- **ウェブパブリッシュツール**ダイアログボックスで、**制御をリクエスト**チェックボックスをオンにすると、ブラウザに埋め込まれた VI の制御を即時にリクエストすることができます。
- すべてのアドレスをリスンせずに、リスンするネットワークアドレスを指定するには、「TCP リスナを作成 (TCP Create Listener)」関数の**ネットアドレス**パラメータを使用します。0 を**ポート**入力に配線すると、オペレーティングシステムが有効であると判断した使用可能な TCP ポートを動的に選択できます。
- 「リファレンスを閉じる (Close Reference)」関数は、「Close LV Object Reference」および「Automation Close」関数に代わるものです。
- 「VI リファレンスを開く (Open VI Reference)」の**オプション**入力の 0x10 フラグは、不明のサブ VI を検索するようユーザにプロンプトします。

VI サーバの新規プロパティとメソッド

新しいプロパティおよびメソッドのリストは、『LabVIEW ヘルプ』の LabVIEW 7.0 の機能リストをご覧ください。LabVIEW では、以下の VI サーバプロパティおよびメソッドが新たに追加されました。

- アプリケーションクラスと VI クラスには、リモートフロントパネルを開いたり閉じたり制御を要求するのに使用する新しい VI サーバプロパティおよびメソッドがいくつか追加されています。リモートフロントパネルのプロパティの使用例については、`examples¥remotepanel` ディレクトリにある「Programmatic Remote Panel Control-Client」VI と「Programmatic Remote Panel Control-Server」VI を参照してください。
- 「制御器の値を取得 (バリエーション)」、「制御器の値を設定 (バリエーション)」、および「すべての制御器の値を取得 (バリエーション)」メソッドを使用して、LabVIEW バリエーションデータタイプに制御器の値を読み書きします。これらのメソッドを使用する場合、データを平坦化したり、タイプデスクリプタを指定する必要はありません。
- 「値 (信号)」プロパティは、制御器の値を設定して「値変更」イベントを生成します。このプロパティは、「値」プロパティと同様に、オブジェクトの値を更新します。ただし、「値 (信号)」プロパティはさらに、ユーザがオブジェクトの値を対話式に変更したかのように、イベントを生成させます。ナショナルインストルメンツでは、プログラマ的な値の変更に応答して LabVIEW がイベントを生成するようにしたい場合のみ、このプロパティを使用することをお勧めします。

既存の VI サーバプロパティおよびメソッドへの変更点

LabVIEW では、既存の VI サーバプロパティおよびメソッドで以下の点
が変更されました。

- LabVIEW の制御器プロパティには、「キャプション」プロパティの
ように、LabVIEW 制御器へのリファレンスを返すものがあります。
旧バージョンの LabVIEW では、これらのプロパティは LabVIEW が
そのプロパティを呼び出すたびに新しいリファレンスを返していまし
た。「Close LV Object Reference」関数を使用して明示的にリファ
レンスを閉じなかった場合、そのリファレンスは VI が停止するまで
メモリ内に残ったままでした。

LabVIEW 7.0 では、制御器プロパティは LabVIEW が呼び出すた
びに新しいリファレンスを作成せず、毎回同じリファレンスを返しま
すので、制御器プロパティのリファレンスを格納するために使用され
るメモリの量を低減できます。また、制御器プロパティのリファレン
スを明示的に閉じる必要もありません。

- クラスタで「値」プロパティを使用する場合は、値を「名前でバンド
ル解除 (Unbundle By Name)」関数に配線してクラスタ内の要素
の値を取得します。

ドキュメントにおける内容の強化

LabVIEW では、ドキュメントリソースにおいて以下の点が強化されま
した。

- キーボードショートカットの最新リストについては、『クイックリ
ファレンスカード』および『LabVIEW ヘルプ』を参照してくださ
い。
- スタンドアロンアプリケーションまたは共有ライブラリを作成する場
合の『注意事項とヒント』については、『LabVIEW アプリケーショ
ンビルダユーザガイド』を参照してください。
- ツールバーの**ヘルプを表示**ボタンをクリックして、**ヘルプ**ウィンドウ
を表示することもできます。
- **ヘルプ**ウィンドウのサイズを変更して、最大サイズに設定すること
もできます。
- **ヘルプ**ウィンドウには、「プロパティノード (Property Node)」ま
たは「インボークノード (Invoke Node)」のショートカットメ
ニュー上にカーソルを移動すると、各プロパティまたはメソッドの説
明が表示されます。
- ほとんどの LabVIEW サブパレットでは、アイコン上やサブパレット
内の空白部分上にカーソルを移動することによって、**ヘルプ**ウィンド
ウにサブパレットの説明とリンクが表示されます。カスタムパレット
セット内で、サブパレットの説明を作成したり、サブパレットを
HTML ファイルやコンパイル済みのヘルプファイルにリンクしたりす

るには、**ツール→上級→パレットセットを編集**を選択し、サブパレットを右クリックして、ショートカットメニューから**サブメニューードキュメントを編集**を選択します。

- 『LabVIEW ヘルプ』のVI および関数リファレンスのトピックには、**ブロックダイアグラムに配置する**および**関数パレットを表示する**ボタンがあります。**ブロックダイアグラムに配置する**ボタンをクリックすると、オブジェクトをカーソルでつかんでブロックダイアグラムに配置することができます。**関数パレットを表示する**ボタンをクリックすると、**関数パレット**上でそのオブジェクトがハイライトされます。『LabVIEW ヘルプ』でステップごとの手順説明を含むトピックでは、オブジェクトをフロントパネルやブロックダイアグラムに配置するための各ステップで、これらのボタンが表示されます。
- VI および関数リファレンストピックの上部にあるコネクタペーンの画像で、入力または出力名をクリックすると、その入力または出力のトピックの説明までスクロールすることができます。
- **VI プロパティ**ダイアログボックスまたは「オンラインヘルプを制御 (Control Online Help)」関数を使用して、Windows 上で .chm ファイルにリンクするか、Mac OS と UNIX でコンパイルされていない HTML ファイルにリンクする場合、HTML ファイルは help¥html¥help.chm ディレクトリに入っている必要があります。ここで help.chm は、Windows 上でリンクする .chm ファイル名です。
- 『LabVIEW Analysis Concepts』マニュアルは、ポイントバイポイント VI に関する情報を収録しています。LabVIEW で使用する解析の概念については、このマニュアルを参照してください。LabVIEW 6.1 以前では、ポイントバイポイント VI は『Getting Started with Point-By-Point VIs』マニュアルで説明されていました。

LabVIEW 7.0 でのその他の機能および変更点

アプリケーションビルダにおける機能強化

アプリケーションビルダの旧バージョンと現行バージョンでの変更点については、『LabVIEW アプリケーションビルダユーザガイド』を参照してください。

新規 VI および関数

LabVIEW には以下の VI および関数が含まれています。

- **(Windows, Mac 9.x 以前, Linux)** 入力デバイス制御 VI を使用して、コンピュータに接続されているジョイスティック、キーボード、またはマウスに関する情報を得ることができます。入力デバイス制御 VI は、Mac OS 9.x 以前でのみ利用できます。入力デバイス制御 VI のサンプルについては、examples¥input¥InputDemo.llb を参照してください。

- アナログ波形 VI および関数を使用して、波形に対し算術関数と比較関数を実行します。特殊関数および数値関数 VI を使用して、一般的な数学関数を実行します。
- 解析を実行するには、以下の VI を使用します。「波形アライメント (連続) (Align Waveforms (continuous))」、「波形アライメント (単発) (Align Waveforms (single shot))」、「アナログ - デジタル波形変換 (Analog to Digital Waveform)」、「ベルヌーイノイズ (Bernoulli Noise)」、「ベルヌーイノイズ波形 (Bernoulli Noise Waveform)」、「バイナリ M 系列 (Binary MLS)」、「2 項ノイズ (Binomial Noise)」、「2 項ノイズ波形 (Binomial Noise Waveform)」、「デジタル - アナログ波形変換 (Digital to Analog Waveform)」、「ガンマノイズ (Gamma Noise)」、「ガンマノイズ波形 (Gamma Noise Waveform)」、「逆 f フィルタ係数 (Inverse f Filter Coefficients)」、「逆 f フィルタ (Inverse f Filter)」、「逆 f ノイズ波形 (Inverse f Noise Waveform)」、「M 系列シーケンス波形 (MLS Sequence Waveform)」、「ポアソンノイズ (Poisson Noise)」、「ポアソンノイズ波形 (Poisson Noise Waveform)」、「波形のリサンプル (連続) (Resample Waveforms (continuous))」、「複数波形のリサンプル (単発) (Resample Waveforms (single shot))」、「トーンとノイズ (Tones and Noise)」、「トーンとノイズ波形 (Tones and Noise Waveform)」

これらの VI のサンプルについては、`examples\measure\resample_align_xmpl.llb` を参照してください。

- 「デフォルトブラウザ内 URL を開く (Open URL in Default Browser)」VI を使用して、デフォルトのウェブブラウザに URL または HTML ファイルを表示することができます。デフォルトのウェブブラウザに HTML レポートを表示するには、「ブラウザ内 HTML レポートを開く (Open HTML Report in Browser)」VI を使用します。
ブラウザで HTML レポートを表示するサンプルについては、`examples\reports\withHTML.llb` にある「Example HTML Report」VI を参照してください。
- 「3 ボタンダイアログ (Three Button Dialog)」VI を使用すると、メッセージと 3 つのボタンを含むダイアログボックスを表示できます。
- 「前後の空白文字を削除 (Trim Whitespace)」VI を使用して、文字列の開始部分、終了部分、または両端の空白スペース (スペース、タブ、復帰、および改行) をすべて削除します。
- 「SO バッファ数設定 (SO Set Num Buffers)」VI を使用して、音響操作に関連する出力バッファの数を設定します。
- **(UNIX)** 「Open System Command Pipe」VI を使用して、システムシェルコマンドへのパイプを開き、後続のパイプ VI に渡すためのファイルデスクリプタを返します。
- 「パターンの検索と置換 (Search and Replace Pattern)」VI を使用して、文字列内の正規表現を検索し、一致するサブ文字列をすべて別の文字列に置き換えます。

- 「エラーコードからエラークラスタ (Error Cluster From Error Code)」VI を使用して、エラーまたは警告コードをエラークラスタに変換します。
- 「メモリ解放要求 (Request Deallocation)」関数を使用して、その関数が含まれる VI の実行後に使用していないメモリを割り当て解除します。
- 「XML ファイル読み取り (Read From XML File)」VI を使用して、LabVIEW XML ファイルからタグを読み取って構文解析します。XML スキーマにおける更新は、タイムスタンプ、デジタル波形、デジタルデータ、ダイナミックデータタイプなどの新しい計測データタイプを反映したものです。またそのスキーマには、さらにわかりやすいファレンスの説明が含まれています。XML スキーマ LVXMLSchema.xsd は、labview¥help から labview¥vi.lib¥utility に移動しました。
- 「セマフォステータス取得 (Get Semaphore Status)」VI の**取得可能数**出力は新規出力で、現時点でセマフォを集録できるタスクの数が含まれています。**サイズ**出力にはセマフォを同時に取得できるタスクの最大数が含まれています。
- 「キー読み取り (Read Key)」VI の文字列インスタンスの**マルチバイトエンコード**入力は、新たに追加された入力です。

既存の VI および関数における変更点

LabVIEW では、既存の VI および関数において以下の点を変更されています。

- 「ノーティフィケーションを待機 (Wait on Notification)」、「キューに要素を追加 (Enqueue Element)」や「キューから要素を削除 (Dequeue Element)」などの待機するキュー関数やノーティファイア関数は、ある状況においてエラーコード 1 の代わりにエラーコード 1122 を返すようになりました。新しいエラーコードは、関数が待機しているキューまたはノーティファイア refnum が無効になったことを指定します。
- 「構成データを開く (Open Config Data)」VI の**構成ファイルパス**入力は、必須入力です。構成データオブジェクトへのリファレンスを開いた場合でも、**構成ファイルパス**を指定する必要があります。
- 「構成データを閉じる (Close Config Data)」VI の**構成ファイルに書き込み?**入力のデフォルトは TRUE です。LabVIEW 6.1 以前では、デフォルトは FALSE でした。
- MATLAB スクリプトノードは、文字列とパスをサポートしています。MATLAB スクリプトサーバを呼び出しても、MATLAB ウィンドウは起動しません。MATLAB スクリプトの内容を消去するには、スクリプトノードを右クリックしてショートカットメニューから**スクリプトを消去**を選択します。LabVIEW で MATLAB スクリプトノードを使用して、MATLAB でと同じ MATLAB スクリプトを実行することができます。ただし、現在の作業ディレクトリは各アプリケーションで異なります。

ます。たとえば、LabVIEW MATLAB スクリプトノードから MATLAB の save コマンドを呼び出したときに、ファイルの保存場所を指定していない場合、MATLAB スクリプトから save コマンドを呼び出したときとは異なる場所にファイルが保存されます。LabVIEW と MATLAB のどちらから save を呼び出しても、常に同じディレクトリにファイルを保存したい場合は、スクリプトの頭に `cd <dir>` を追加します。ここで `<dir>` はファイルを保存するディレクトリです。

- 「2 の累乗でスケール (Scale By Power Of 2)」関数は、入力データ (x) が符号なしの際にビットシフトを実行します。
- **(Windows)** 「In Port」VI と 「Out Port」VI は、システムメモリから 32 ビット値を読み取ることができます。Windows 2000/NT/XP で 「In Port」VI および 「Out Port」VI を使用する場合は、Accesshw.zip ファイルをダウンロードする必要はありません。
- 「秒を日付 / 時間に変換 (Seconds To Date/Time)」関数および 「日付 / 時間文字列を取得 (Get Date/Time String)」関数は、LabVIEW 7.0 では最も近い秒数に丸めません。これらの関数は、完了した秒数を表示します。たとえば、3.6 秒の場合、3 は完了した秒であるため、3 と表記されます。この動作は、国際標準時に適合しています。
- 配列やクラスタの要素または基礎群を比較するため 2 つ以上の入力を持つ比較関数を設定するには、関数を右クリックしてショートカットメニューから **比較モード** → **要素の比較** または **比較モード** → **基礎群の比較** を選択します。
- 数学 VI は **解析** バレットにあります。

その他

LabVIEW では、そのほかに以下のような変更があります。

- マルチシートライセンスをサポートしています。マルチシートライセンスの詳細については、ni.com/license を参照してください。
- **(Windows)** 現在の LabVIEW のインストールを変更したり、LabVIEW をアンインストールするには、コントロールパネルの 「アプリケーションの追加と削除」 で **National Instruments ソフトウェア** を選択してください。インストールを変更すると、ナショナルインスツルメンツのソフトウェアのリストが表示されます。各コンポーネントを追加または削除するが、製品をアンインストールするには、リストから製品を選択します。複数の製品を選択して **アンインストール** ボタンをクリックすると、選択したすべての製品が削除されます。
- パス制御器を右クリックしてショートカットメニューから **参照オプション** を選択した場合は、**パターンラベル** テキストボックスを使用して、カスタムフィルパターンのラベルを **一致パターン** テキストボックスに入力します。**パターンラベル** テキストボックスに入力した文字列は、カスタムパターンの隣のファイルダイアログボックスに表示されます。「ダイアログオプションを参照: パターンラベル」プロパティを使用して、このラベルをプログラムで設定することもできます。

- 「ファイルダイアログ (File Dialog)」関数の**パターンラベル**入力を使用して、**パターン**入力に配線したカスタムファイルパターンのラベルを指定します。**パターンラベル**入力に配線した文字列は、カスタムパターンの隣のファイルダイアログボックスに表示されます。**ボタンラベル**入力を使用して、ファイルダイアログボックスの **OK** ボタンのラベルを指定します。**選択モード**でユーザがディレクトリを選択できるようにした場合は、**ボタンラベル**入力を使用してファイルダイアログボックスの**選択**ボタンのラベルを指定します。
- パス制御器を右クリックしてショートカットメニューから**参照オプション**を選択した場合は、**選択モード**プルダウンメニューから**新規または既存ファイル/ディレクトリ (LLB 使用)**を選択することができます。このオプションを選択すると、.11b ファイルの内容を含む新規または既存のファイルまたはディレクトリをファイルダイアログボックスから選択できるようになります。また、「ファイルダイアログ」関数の**選択モード**入力から制御器や定数を作成して、**新規または既存ファイル/ディレクトリ (LLB 使用)**モードを選択することもできます。
- パスを検索して選択するには、パス定数を右クリックして、ショートカットメニューから**パスを参照**を選択します。
- LabVIEW 7.0 では、FFT 演算は大まかに $N\log N$ に比例して増え続けます。また FFT 演算の速度はサイズの因数分解可能性によって異なることはなくなりました。
- Perforce をソースコード管理プロバイダとして設定している場合、SCC ユーザインタフェースは Perforce ユーザインタフェースのように変わります。
- LabVIEW やスタンドアロンアプリケーションをコマンドラインから起動すると、ユーザ定義の引数を渡すことができます。コマンドラインでは、ユーザ定義の引数のセットの前に、スペースで囲んだ 2 つのハイフン (--) を入力します。LabVIEW は、labview.exe を起動する際にこれらの 2 つのハイフンの後の引数を使用しません。LabVIEW は、2 つのハイフンの後の引数を、起動した VI のブロックダイアグラムに渡します。「アプリケーション: コマンドライン 引数」プロパティを使用して、実行可能ファイルの起動時に渡されるユーザ定義のコマンドライン引数を読み取ります。

このプロパティをスタンドアロンアプリケーションで使用すると、すべての引数をユーザ定義の引数として渡すことができるため、コマンドラインのユーザ定義引数の前に 2 つのハイフンを入力する必要はありません。そのためには、**アプリケーションまたは共有ライブラリ (DLL) を作成**ダイアログボックスの**アプリケーション設定**ページにある**すべてのコマンドライン引数をアプリケーションに渡す**チェックボックスをオンにします。

「アプリケーション: コマンドライン 引数」プロパティを使用してユーザ定義のコマンドライン引数を処理するサンプルについては、examples¥viserver¥cmdline.11b にある「Handling Command Line」VI を参照してください。

- **(UNIX) Edit** メニューの **Import Clipboard** と **Export Clipboard** の各項目は削除されました。コピーおよび貼り付け関数は、Windows および Mac OS と同様に動作します。
- **メモリをなるべく早く解放する** チェックボックスと **デフォルトのタイマを使用する** チェックボックスは、**オプション** ダイアログボックスの **パフォーマンス&ディスク** ページには表示されなくなりました。