

LabVIEW™ アップグレードノート

バージョン 7.1

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 7.1 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、新機能について説明します。

LabVIEW 6.1 以前のバージョンを LabVIEW 7.1 にアップグレードする場合、LabVIEW 開発環境の強化やカスタマイズについては、『LabVIEW 7.0 アップグレードノート』を参照してください。ナショナルインスツルメンツでは、LabVIEW 6.1 以前のバージョンをアップグレードした場合には、それらのアップグレードノートの他に、『LabVIEW 7.0 アップグレードノート』もお読みになることを推奨します。National Instruments の Web サイト ni.com/info (英語) にアクセスして、info code の `rd1vuj` を入力すると、『LabVIEW 7.0 アップグレードノート』を表示することができます。日本語のマニュアルについては、ni.com/jp/manuals を参照してください。

『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 7.1 の機能についての詳細は、『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→オンラインヘルプリファレンス**を選択します。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。LabVIEW ドキュメントライブラリは、**ヘルプ→LabVIEW ドキュメントライブラリを表示**を選択すると表示することができます。

この PDF ドキュメントを表示、使用するには、検索およびアクセシビリティのプラグインを備えた Adobe Reader 5.0.5 以降のバージョンが必要です。

(Mac OS) PDF ファイルの表示や使用には、検索およびアクセシビリティプラグインを備えた Adobe Reader 6.x をご使用ください。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。

DataSocket™、DIAdem™、FieldPoint™、HiQ™、IVI™、LabVIEW™、MATRIX™、National Instruments™、NI™、ni.com™、NI-DAQ™、NI-VISA™、xmath™ は、National Instruments Corporation の商標です。本書に掲載されている製品および会社名は該当各社の商標または商号です。National Instruments 製品を保護する特許については、ソフトウェアに含まれている特許情報 (ヘルプ→特許情報)、CD に含まれている `patents.txt` ファイル、または ni.com/patents のうち、該当するリソースから参照してください。USI (Xerces C++, ICU、および HDF5) で使用されるコンポーネントについては、以下の著作権が適用されます。条件および免責事項については、`USICopyrights.chm` をご参照ください。

2004 年 5 月版
322784D-01

目次

アップグレード時の問題点	3
VI を変換する	3
ツールキット、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	4
LabVIEW 7.0 からアップグレードする	5
LabVIEW 6.x からアップグレードする	7
LabVIEW 5.x からアップグレードする	12
LabVIEW 4.x からアップグレードする	13
LabVIEW 3.x 以前のバージョンからアップグレードする	15
LabVIEW 7.1 の新機能	15
ラジオボタン制御器を使用する	15
解析 VI の機能強化	15
多項式 VI と有理多項式 VI	16
線形代数パレット	16
時間領域パレット	18
周波数領域パレット	18
ストレージ VI	18
信号を付加 (Append Signals) Express VI	18
Bluetooth VI および関数	19
グラフ、チャート、表、デジタルデータ制御器を画像として保存する	19
ナビゲーションウィンドウを使用する	19
バッファの割り当てを表示する	20
Xmath スクリプトノード	20
LabVIEW とハイパスレッド	21
サンプル VI を作成して NI サンプルファインダに表示する	21
NI サンプルファインダの強化機能	21
タイミングループ	22
VI サーバ、およびリモートフロントパネルにおける機能強化	22
新規プロパティとメソッド	22
LabVIEW 関連製品 CD	22
ドキュメントの強化と変更	23
改訂しない文書	23
旧バージョンで使用できるように VI を保存する	23
LabVIEW 7.1 でのその他の機能および変更点	24
アプリケーションビルダにおける機能強化	24
既存の VI および関数における変更点	24
新規作成、移動、名前変更されたサンプル VI	26
その他	26

アップグレード時の問題点

LabVIEW 7.0 からアップグレードする場合はまず、本書の「VI を変換する」、「ツールキット、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 7.0 からアップグレードする」の各セクションをお読みください。

LabVIEW 6.x からアップグレードする場合はまず、本書の「VI を変換する」、「ツールキット、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」の各セクションをお読みください。

LabVIEW 5.x からアップグレードする場合はまず、本書の「VI を変換する」、「ツールキット、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」の各セクションをお読みください。

LabVIEW 4.x からアップグレードする場合はまず、本書の「VI を変換する」、「ツールキット、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「ツールキット、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 7.1 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 7.1 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。このパフォーマンスの問題についての詳細は、『LabVIEW Performance and Memory Management』アプリケーションノートの「Memory Issues in Front Panels」のセクションを参照してください。



メモ

LabVIEW 7.1 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 7.0 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 7.0 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 20 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→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、およびアドオンをアップグレードする

LabVIEW 7.1 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 7.1 のディレクトリに再インストールします。

また、LabVIEW 7.1 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 の一括コンパイルについての詳細は、このドキュメントの「[VI を変換する](#)」のセクションを参照してください。LabVIEW 7.1 は、LabVIEW アプリケーションビルダを除き、LabVIEW 4.0 以降のバージョン用に設計されたツールキット、計測器ドライバ、アドオンに対応しています。

(開発システム) 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 7.1 にアップグレードしてください。LabVIEW プロフェッショナル開発システム 7.1 には、アプリケーションビルダ 7.1 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についての詳細は、『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。

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 7.1 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 7.1 ディレクトリにコピーします。LabVIEW 7.1 環境設定ファイルを旧バージョンのファイルと入れ替える場合は、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。

(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、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` ディレクトリのファイルをコピーします。LabVIEW 7.1 をインストールしたら、ファイルを LabVIEW 7.1 ディレクトリの `user.lib` ディレクトリにコピーします。

(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 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 7.1 に入れ替えることができます。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。



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、アンインストール時または再インストール時に、`.lib` ファイルがアンインストールされます。これには、ユーザが `.lib` ファイルに保存したすべての VI や制御器も含まれます。VI および制御器を **関数** および **制御器** パレットに追加するには、その VI および制御器を `user.lib` ディレクトリに保存します。

既存バージョンの LabVIEW を入れ替えるには、LabVIEW 7.1 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。

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 7.0 からアップグレードする

LabVIEW 7.0 から LabVIEW 7.1 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。

サポートされているプラットフォーム

LabVIEW 7.1 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 では、Windows Me/98 はサポートされていません。
- LabVIEW 7.1 では、Mac OS 9.x 以前のバージョンおよび PPC toolbox ルボックスはサポートされていません。
- LabVIEW 7.1 では、Sun Solaris 7 以降のバージョンがサポートされています。

FP を開くメソッドを使用する

LabVIEW 7.0 の「FP を開く」メソッドは「旧フロントパネルを開く」の名称が変更されたもので、この新しいメソッドは LabVIEW 7.1 に含まれています。LabVIEW 7.1 のこのメソッドは、フロントパネルが開いているとエラーを返しません。「旧フロントパネルを開く」メソッドが使用されている VI については、このメソッドを新しい「FP を開く」メソッドか、または `labview\vi.lib\oldvers\oldvers.lib` にある LabVIEW 7.1 OpenFP VI と入れ替えてください。

関数パレットから削除された VI

LabVIEW 7.1 では、以下の VI はインストールされません。

- **HP34401A Find Range.vi** : HP34401A Config Measurement.vi を代用してください。
- **「多項式実ゼロカウンタ (Polynomial Real Zero Counter)」 VI** : 「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。
- **PPC VIs** : TCP VI を代用してください。

内積 (Dot Product) VI についての変更

LabVIEW 7.0 では、「内積」VI によって以下のような公式の計算が行われます。

$$X*Y = \sum_{i=0}^{n-1} X_i Y_i$$

LabVIEW 7.1 では、「内積 (Dot Product)」VI によって以下の計算が行われます。

$$X*Y = \sum_{i=0}^{n-1} X_i Y_i^*$$

ここで、 y_i^* は y_i の複素共役です。

LabVIEW 6.x からアップグレードする

LabVIEW 6.x から LabVIEW 7.1 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 7.0 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 7.1 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。

Windows 95 のサポート

LabVIEW 7.1 は Windows 95 をサポートしていません。

波形データをファイルに保存する

LabVIEW 7.0 では、倍精度浮動小数点数ではなく t_0 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するため、波形のデータタイプがアップデートされました。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のデータをファイルに保存した場合、LabVIEW 7.1 でそのデータを検索しようとするエラーが発生する可能性があります。LabVIEW 6.x から LabVIEW 7.1 への波形データ移動については、National Instruments の Web サイト ni.com/info にアクセスして info code の `exd9zq` を入力すると詳細が表示されます。

HiQ のサポート

Mac OS 対応の LabVIEW 7.1 では HiQ 機能はサポートされていません。また、LabVIEW 7.1 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、それを解析 VI および数学 VI と入れ替えることができないかどうかを検討してください。解析 VI および数学 VI の使用については、『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 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 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、`examples\instr\smplser1.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した。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 では、LabVIEW 7.0 より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。再入実行に関する詳細は、『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 によるスレッドの割り当てを変更するには、`vi.lib\Utility\sysinfo.llb`の`threadconfig.vi`を使用してください。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。

再入実行 VI の使用に関する詳細は、『Using LabVIEW to Create Multithreaded VIs for Maximum Performance and Reliability』アプリケーションノートを参照してください。

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) からダウンロードすることができます。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 7.1 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 7.1 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	7.1 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	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

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 7.1 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なかったため、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、ファイル入力フィールドの右側に**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 7.1 では、ユーザー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。プ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 7.1 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。

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で指数を表していました。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。 \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。

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、大文字小文字も同じでなくてはなりません。

技術サポートフォーム

LabVIEW のインストールプログラムでは、`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 からアップグレードする

LabVIEW 5.x から LabVIEW 7.1 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り得る他の問題点は、本書の「[LabVIEW 6.x からアップグレードする](#)」、[「LabVIEW 7.0 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 5.x から 7.1 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/jp/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 7.1 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、または波形データタイプを含んでいる場合、LabVIEW 7.1 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 7.1 ではそのファイルを LabVIEW 7.1 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き換えられます。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 7.1 クライアント間の互換性の問題

LabVIEW 7.1 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失敗となります。これは、LabVIEW 5.x のアプリケーションでは LabVIEW 7.1 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 7.1 アプリケーションの VI サーバへの接続もできません。

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.llb` にあります。

LabVIEW 4.x からアップグレードする

LabVIEW 4.x から LabVIEW 7.1 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り得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](#)」、[「LabVIEW 6.x からアップグレードする](#)」、および「[LabVIEW 7.0 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。

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 7.1 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、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 7.1 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 7.1 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 7.1 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 7.1 で同じデータを書き込む場合とは異なります。LabVIEW 7.1 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 で読み取ることで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)」、「ファイルから読み取り (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 7.1 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 7.1 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 7.1 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 で LabVIEW 7.1 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 7.1 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\vict1.lib 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ノード (Call By Reference 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 7.1 では異なるものがいくつかあります。以前のバージョンの 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 7.1 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 7.1 で動作するように修正する必要があります。

DDE VI

(Windows) DDE VI は関数パレットには表示されず、`vi.lib`¥`platform¥dde.lib` 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](#)」、「[LabVIEW 6.x からアップグレードする](#)」、および「[LabVIEW 7.0 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。

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 7.1 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 7.1 の新機能

LabVIEW 7.1 の機能についての詳細は、『LabVIEW ヘルプ』を参照してください。

ラジオボタン制御器を使用する

1 度に 1 つのみ選択できる項目リストを取得するには、ラジオボタン制御器を使用します。項目を 1 つ選ぶか何も選ばないかというオプションをユーザに与えるには、制御器を右クリックして、ショートカットメニューから**選択なしを許可**を選択し、メニュー項目の隣にチェックマークを付けます。ラジオボタン制御器は、データタイプが列挙体のため、ケーススタックのケースを選択することができます。

ラジオボタン制御器のサンプルについては、`labview`¥`examples`¥`general`¥`controls`¥`booleans.lib` の Radio Buttons Control VI および Radio Buttons with Event Structure VI を参照してください。

解析 VI の機能強化

解析 VI には、以下のような機能強化が行われています。

- 線形代数演算の演算時間を削減するための BLAS/LAPAK アルゴリズムの使用
- 多項式演算のための 2 つの新しいパレット
- 多態性線形代数および周波数領域 VI
- 新しい線形代数 VI と時間領域 VI

多項式 VI と有理多項式 VI

多項式 VI を使用して、多項式を使用した計算および評価を実行します。
有理多項式 VI を使用して、有理多項式を使用した計算および評価を実行します。

線形代数パレット

線形代数パレットには、上級線形代数、複素線形代数、上級複素線形代数の各サブパレットは存在しません。これらのパレットにあった VI は、線形代数パレット上の VI の多態性インスタンスとなりました。以下の表は、削除された VI とその新しい場所を示します。

LabVIEW 7.0 VI	LabVIEW 7.1 VI	多態性インスタンス
A x ベクトル	A x B	A x ベクトル
複素 A x B	A x B	複素 A x B
複素 A x ベクトル	A x B	複素 A x ベクトル
複素コレスキー分解	コレスキー分解	複素コレスキー分解
複素共役転置行列	転置行列	複素共役転置行列
複素行列式	行列式	複素行列式
複素内積	内積	複素内積
複素固有値とベクトル	固有値とベクトル	複素固有値とベクトル
複素逆行列	逆行列	複素逆行列
複素 LU 因数分解	LU 分解	複素 LU 因数分解
複素行列数	行列数	複素行列数
複素行列ノルム	行列ノルム	複素行列ノルム
複素行列階数	行列階数	複素行列階数
複素行列トレース	行列トレース	複素行列トレース
複素疑似逆行列	疑似逆行列	複素疑似逆行列
複素外積	外積	複素外積
複素 QR 分解	QR 分解	複素 QR 分解
特殊複素行列作成	特殊複素行列作成	特殊複素行列作成
複素線形方程式を解く	線形方程式を解く	複素線形方程式解 (単数右辺)、 複素線形方程式解 (複数右辺)
SVD 分解	SVD 分解	実 SVD 分解
複素正定値テスト	正定値テスト	複素正定値テスト

線形代数パレットには、以下の新しい VI が含まれています。

- 実均衡行列または複素均衡行列の固有ベクトルを元の行列の固有ベクトルに変換するには、「固有ベクトル逆変換 (Back Transform Eigenvectors)」VI を使用します。
- 固有値から実行列を生成するには、「固有値から実行列を作成 (Create Real Matrix From EigenValues)」VI を使用します。
- 実行列または複素行列のペアの右固有値および固有ベクトルを計算するには、「一般化固有値とベクトル (Generalized Eigenvalues and Vectors)」VI を使用します。
- 実行列または複素行列のヘッセンベルグ分解を行うには、「ヘッセンベルグ分解 (Hessenberg Decomposition)」VI を使用します。
- 演算される固有値および固有ベクトルの精度を向上するには、「行列均衡 (Matrix Balance)」VI を使用して実行列または複素一般行列の均衡をはかります。
- 実行列の特性多項式を計算するには、「行列特性多項式 (Matrix Characteristic Polynomial)」VI を使用します。
- 実行列または複素行列の指数を計算するには、「行列指数 (Matrix Exp)」VI を使用します。
- 実行列または複素行列の自然対数を計算するには、「行列対数 (Matrix Logarithm)」VI を使用します。
- 実行列または複素行列の乗数 μ を計算するには、「行列累乗 (Matrix Power)」VI を使用します。
- 実行列または複素行列の 2 乗を計算するには、「行列平方根 (Matrix Square Root)」VI を使用します。
- 実行列または複素行列のペアの QZ 分解を行うには、「QZ 分解 (QZ Decomposition)」VI を使用します。
- 実行列または複素行列のシューア分解を行うには、「シューア分解 (Schur Decomposition)」VI を使用します。
- $m \times n$ の実行列または複素行列の特異値分解を計算するには、「SVD 分解 (SVD Decomposition)」VI を使用します。
- シルベスター行列式を実行列または複素行列について解くには、「シルベスター行列式 (Sylvester Equations)」VI を使用します。
- 実行列または複素行列を転置するには、「転置行列 (Transpose Matrix)」VI を使用します。

時間領域パレット

時間領域パレットには、以下のような新しい VI があります。

- 「リサンプル (定数から定数) (Resample (constant to constant))」 VI を使用して、サンプリング間隔とタイムスタンプのために指定した値に従って、単一の入力信号または複数の入力信号をリサンプルします。
- 「リサンプル (定数から変数) (Resample (constant to variable))」 VI を使用して、指定した時間インスタンスに従って単一の入力信号または複数の入力信号をリサンプルします。

周波数領域パレット

LabVIEW 7.0 以前バージョンでは、**周波数領域**パレットに実数 FFT、実数逆 FFT、複素 FFT、および複素逆 FFT が含まれていました。これに対して LabVIEW 7.1 では、**周波数領域**パレットには多態性の FFT VI と逆 FFT VI が含まれています。FFT VI は実信号または複素信号を処理することができます。逆 FFT VI によって、実数 FFT または複素逆 FFT を計算することができます。

ストレージ VI

(Windows) ストレージ VI を使用して、NI テストデータ交換ファイル (.tdm) に対して、波形と波形プロパティの読み書きを行います。LabVIEW や DIAdem などの NI ソフトウェアの間でデータ交換を行うには、.tdm ファイルを使用してください。

ストレージ VI によって、波形と波形のプロパティが組み合わせられ、チャンネルが形成されます。チャンネルグループはチャンネルで構成されます。ファイルには、チャンネルグループのセットがあります。数値の他、文字列の配列、タイムスタンプの配列がストレージ VI によってサポートされます。リファレンス番号は、ブロックダイアグラム上のチャンネル、チャンネルグループ、およびファイルを指します。

また、ストレージ VI を使用してファイルの問い合わせをし、指定条件を満たすチャンネルグループやチャンネルを取得することもできます。

ストレージ VI を使用したサンプルについては、`labview\examples\file\storage.11b` のサンプルを参照してください。

信号を付加 (Append Signals) Express VI

信号を相互に追加するには、「信号を付加 (Append Signals)」Express VI を使用します。「信号を付加 (Append Signals)」Express VI のサンプルについては、`examples\express` ディレクトリにある Append Signals VI を参照してください。

Bluetooth VI および関数

(Windows) Bluetooth 通信プロトコルを使用したデバイスと通信するには、Bluetooth VI および関数を使用します。LabVIEW では、Windows XP サービスパック 1 以降で Microsoft Bluetooth ドライバを使用した Bluetooth デバイスのみがサポートされています。Microsoft Bluetooth ドライバがサポートされている Bluetooth デバイスについての詳細は、www.microsoft.com の Microsoft Windows カタログの Web サイトを参照してください。ほとんどの Bluetooth デバイスには、デフォルトで専用の Bluetooth ドライバが使用されています。このデバイスを LabVIEW で使用するには、Microsoft Bluetooth ドライバに切り替えなければなりません。ドライバを切り替えるには、デバイスのメーカーから Microsoft Bluetooth ドライバ CD を取得する必要があります。

LabVIEW での Bluetooth 通信についての詳細は、『Using LabVIEW with Wireless Devices』アプリケーションノートを参照してください。Bluetooth 技術についての詳細は、www.bluetooth.com の Bluetooth Web サイトを参照してください。Bluetooth VI および関数のサンプルについては、`labview\examples\comm\Bluetooth.llb` を参照してください。

グラフ、チャート、表、デジタルデータ制御器を画像として保存する

制御器や表示器の画像をクリップボードに、または .emf や .bmp ファイルとして保存するには、グラフ、チャート、表、またはデジタルデータの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 **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。

ナビゲーションウィンドウを使用する

ナビゲーションウィンドウには、アクティブな編集モードのフロントパネルまたはブロックダイアグラムの概要が表示されます。大きなフロントパネルやブロックダイアグラムをナビゲートするには、**ナビゲーションウィンドウ**を使用します。フロントパネルやブロックダイアグラムウィンドウの特定の領域を表示するには、**ナビゲーションウィンドウ**の画像領域をクリックします。また、フロントパネルやブロックダイアグラムをスクロールするには、**ナビゲーションウィンドウ**で画像をクリックしてドラッグします。フロントパネルやブロックダイアグラムに表示されない部分は、**ナビゲーションウィンドウ**ではグレー表示されます。

ナビゲーションウィンドウを表示するには、**ウィンドウ→ナビゲーションウィンドウを表示**を選択するか、または <Ctrl-Shift-N> キーを押します。**(Mac OS)** では <Command-Shift-N> キー、**(Sun)** では <Meta-Shift-N> キー、**(Linux)** では <Alt-Shift-N> キーを使用します。



メモ **ナビゲーションウィンドウ**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。

バッファの割り当てを表示する

バッファ割り当てを表示ウィンドウを表示するには、**ツール→上級→バッファ割り当てを表示**を選択します。バッファを確認するデータタイプの横にチェックマークを付けて、**リフレッシュ**ボタンを押します。ブロックダイアグラム上には、LabVIEW がバッファを作成してブロックダイアグラム上のデータを保持する場所を示す黒い四角が表示されます。



メモ バッファ割り当てを表示ウィンドウは、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 がバッファを作成する場所がわかれば、VI を編集して、LabVIEW による VI の実行に必要なメモリ容量を削減できる可能性があります。メモリ割り当て方法と効率的なメモリ使用については、『LabVIEW Performance and Memory Management』アプリケーションノートの「VI Memory Usage」セクションを参照してください。

VI を変更したために LabVIEW でその VI を再コンパイルする必要がある場合には、バッファ情報が正しくない可能性があるため黒い四角は消えます。VI を再コンパイルして黒い四角を表示するには、**バッファ割り当てを表示**ウィンドウ上の**リフレッシュ**ボタンをクリックしてください。**バッファ割り当てを表示**ウィンドウを閉じると、黒い四角は消えます。

Xmath スクリプトノード

外部 Xmath スクリプトを実行するには、この Xmath スクリプトノードを使用します。Xmath スクリプトノードを使用するには、Xmath をコンピュータにインストールしてください。

数学的解析、表示、スクリプトのアプリケーションを開発するには、NI MATRIXx 製品シリーズの一部である Xmath を使用します。高度なデータ解析には Xmath をスクリプト開発のための動作環境として、また、シミュレーションデータ用の表示ツールとして使用してください。Xmath には、システムモデリング、解析関数、数学のライブラリが含まれています。Xmath および NI MATRIXx についての詳細は、ni.com/matrixx を参照してください。

Xmath スクリプトノードのサンプルについては、`labview\examples\scriptnode\Differential Equation.llb`にある Xmath Script - Lorenz Diff Eq VI および `labview\examples\scriptnode\Fractal.llb`にある Xmath Script - Fractal VI を参照してください。

LabVIEW とハイパースレッド

ハイパースレッドは、高性能な Intel Pentium 4 以降のバージョンの高度な機能です。ハイパースレッド対応のコンピュータには単一のプロセッサしか装備されていませんが、複数のプロセッサが装備されているコンピュータと同様に動作します。テキストベースの言語では、アプリケーションのマルチスレッド処理を行うため、これらのスレッド内で通信するための複数のスレッドおよびコードを作成する必要があります。これに対し、LabVIEW では VI 内のマルチスレッド処理のタイミングを認識できるため、実行システムによりマルチスレッド通信が処理されます。ハイパースレッド対応コンピュータで LabVIEW がどのようにコードを実行するかについては、『LabVIEW and Hyper-Threading』アプリケーションノートを参照してください。

サンプル VI を作成して NI サンプルファインダに表示する

作成したサンプル VI を NI サンプルファインダに保存するには、**ツール→NI サンプルファインダ用にサンプル VI を準備**を選択してください。サンプル VI を保存したら、そのサンプル VI のタスクベース参照パス、キーワードなどの情報を保管するデータ (.bin3) ファイルを作成します。 .bin3 ファイルを使用して、NI サンプルファインダにサンプル VI を表示されます。



メモ 作成したサンプル VI と .bin3 ファイルは examples、instr.lib、または user.lib ディレクトリに常駐させなければなりません。サンプル VI も、ユーザのコンピュータに常駐するのと同じディレクトリ構成に常駐させる必要があります。 .bin3 ファイルのディレクトリが examples、instr.lib、user.lib のディレクトリより 2 レベル以上下位に常駐する場合には、NI サンプルファインダでは、.bin3 ファイルを検索することができません。

NI サンプルファインダの強化機能

頻繁に使用するサンプル VI を整理して簡単にアクセスできるようにするには、NI サンプルファインダの**参照タブのお気に入りフォルダおよび最新フォルダ**を使用してください。また、ハードウェア要件に基づいて、表示するサンプル VI を指定することもできます。ハードウェアデバイスごとのサンプル VI の表示については、『NI サンプルファインダヘルプ』（英語）を参照してください。NI サンプルファインダを最初に起動したときと、デバイスを**ハードウェアプルダウンメニュー**に追加、あるいはメニューから削除したときには常に、**ハードウェアプルダウンメニューからハードウェアリストを更新**を選択してください。

タイミングループ

(Windows 2000/XP) タイミングループは、指定した実行時間でループの繰り返しを実行します。マルチレートタイミング、正確なタイミング、ループ実行時のフィードバック、ダイナミックに変化するタイミング特性、または実行優先順位のいくつかのレベルがある VI を開発する場合には、タイミングループを使用します。タイミングループの使用についての詳細は、『Using the Timed Loop to Write Multirate Applications in LabVIEW』アプリケーションノートを参照してください。タイミングソースを制御する VI を開発するには、タイミングループを備えたタイミングループと DAQmx の VI および関数を使用してください。

タイミングループ VI および関数のサンプルについては、`labview¥examples¥general¥timedloop.11b` を参照してください。

VI サーバ、およびリモートフロントパネルにおける機能強化

さまざまなバージョンの LabVIEW 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。各ブラウザのプラグインは、LabVIEW の同じバージョンで開発された VI を表示することができます。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。CLASSID は、プラグインの各バージョンで異なります。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 7.1 で使用する構文についての詳細は、『LabVIEW ヘルプ』を参照してください。これらのバージョンで使用する構文についての詳細は、以前のバージョンの LabVIEW の『LabVIEW ヘルプ』を参照してください。

新規プロパティとメソッド

LabVIEW 7.1 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。

- 選択なしを許可プロパティ
- スケールされたダイアグラム画像を取得メソッド
- スケールされたパネル画像を取得メソッド
- Is Probe プロパティ

LabVIEW 関連製品 CD

特殊なアプリケーションを開発するためのアドオンソフトウェアツールキットを購入することができます。すべてのツールキットは、シームレスに LabVIEW に統合されます。こうしたツールキットについての詳細は、LabVIEW に付属の LabVIEW 関連製品 CD、ナショナルインスツルメンツのウェブサイト (ni.com/toolkits) を参照してください。

ドキュメントの強化と変更

『LabVIEW ヘルプ』の特定の VI と関数のリファレンス項目には、**サンプルを開く** ボタンと **関連サンプルを参照** ボタンについて記載されています。その項目のサンプル VI を開くには、**サンプルを開く** ボタンをクリックします。NI サンプルファインダを開いて関連するサンプル VI を表示するには、**関連サンプルを参照** ボタンをクリックします。

ナショナルインストルメンツでは、以下のウェブブラウザを使用して『LabVIEW ヘルプ』を表示することをお勧めします。

- **(Mac OS)** Safari 1.0 以降のバージョン
- **(UNIX)** Netscape 6.0 以降のバージョン、Mozilla 1.2 以降のバージョン

改訂しない文書

ナショナルインストルメンツでは、LabVIEW 7.1 のリリースにあたり以下の文書の改訂はいたしません。

- LabVIEW Development Guidelines
- LabVIEW Measurements Manual
- LabVIEW クイックリファレンスカード
- Using External Code in LabVIEW

ナショナルインストルメンツでは、LabVIEW 7.1 のリリースにあたり以下のアプリケーションノートおよびホワイトペーパーの改訂はいたしません。

- Integrating the Internet into Your Measurement System—DataSocket Technical Overview
- LabVIEW 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
- Using LabVIEW to Create Multithreaded VIs for Maximum Performance and Reliability

旧バージョンで使用できるように VI を保存する

LabVIEW 7.1 では、**ファイル→オプション付保存** 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 7.0 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 6.1 で使用できるように保存するには、その VI を LabVIEW 7.0 で開いて **ファイル→オプション付保存** 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。

LabVIEW 7.1 でのその他の機能および変更点

その他、LabVIEW 7.1 では、以下のような諸機能の追加と変更があります。

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。

既存の VI および関数における変更点

LabVIEW 7.1 では、既存の VI および関数において以下の点に変更されました。

- イベント登録解除関数は、**ActiveX** パレットにあります。
- **エラーメッセージ**入力は、「エラーコードからエラークラスター (Error Cluster from Error Code)」VI のオプション入力です。ユーザ定義のエラーコードのエラーの説明を指定するには、**エラーメッセージ**を使用してください。
- 「ノーティフィケーションを待機 (Wait on Notification)」関数、「複数からのノーティフィケーションを待機 (Wait on Notification from Multiple)」関数の**以前を無視**入力の動作により、LabVIEW 6.1 以降のバージョンの動作が変更されました。ループの最初の繰り返しの**以前を無視**入力に TRUE が渡され、その後の繰り返しの FALSE 値が渡された場合には、関数はループの最初の繰り返しの送信されたメッセージを無視します。
- フォーミュラノードでは、atan2 関数がサポートされています。
- 多態性の「デジタル比較 (Digital Comparison)」VI には、波形から波形、表から表の 2 つの新しいインスタンスがあります。「デジタル比較 (Digital Comparison)」VI を使用して、デジタル波形と別のデジタル波形、デジタルデータセットと別のデジタルデータセット、デジタル波形と指定値、また、デジタルデータセットと指定値を比較します。
- 「LabVIEW 計測ファイル書き込み (Write LabVIEW Measurement File)」Express VI には、**リセット**入力があり、これを使用すると LabVIEW 計測データファイル (.lvmm) への書き込みを再度開始することができます。
- 「ダイナミックデータへ変換 (Convert to Dynamic Data)」Express VI には、入力信号の**開始時**を**ゼロ**または現在のシステム時間である**現在**に設定するオプションがあります。
- 「リファレンス呼び出しノード (Call by Reference Node)」の **VI リファレンス**入力は、タイプ指定 VI リファレンスとスタティック VI リファレンスに配線することができます。また、この「リファレンス呼び出しノード」を使用して、スタティック VI リファレンス内の VI を実行することもできます。

- 「Savitzky-Golai フィルタ PtByPt (Savitzky Golai Filter PtByPt)」 VI は、「Savitzky-Golay フィルタ PtByPt (Savitzky Golay Filter PtByPt)」 VI という名前に変更されました。
- LabVIEW 7.0 では、「(不完全) ベータ関数 (Incomplete) Beta Function)」 VI は、 $\alpha = 1$ の場合にベータ関数を完了しました。これに対し、LabVIEW 7.1 では、上限のすべての非正数の実数値が $\alpha \leq 1$ で、 x および y のすべての非負数実数値について、関数が定義されます。
- 以下の VI は `vi.lib¥analysis¥numeric.llb` から `vi.lib¥analysis¥baseonly.llb` へ移動しました。
 - 「正規化ベクトル (Normalize Vector)」 VI
 - 「正規化行列 (Normalize Matrix)」 VI
 - 「クイックスケール 1D (Quick Scale 1D)」 VI
 - 「クイックスケール 2D (Quick Scale 2D)」 VI
 - 「スケール 1D (Scale 1D)」 VI
 - 「スケール 2D (Scale 2D)」 VI
 - 「単位ベクトル (Unit Vector)」 VI
- 「FIR ウィンドウ係数 (FIR Windowed Coefficients)」 VI には、**FIR ウィンドウ係数** をスケールするかどうかを指定する **オプション** という新しい入力があります。デフォルト値は、**not scaled** です。
- LabVIEW 7.1 では、MSE VI の動作が変更されました。**Y 値** と **X 値** 入力配列の長さが異なる場合、MSE VI は要素数が最少のシーケンスに基づいて **mse** を計算し、警告を返します。
- HP34401A Initialize VI には、「VISA オープン (VISA Open)」 VI の高度なアクセスオプションを設定する **Access Mode** という新しいパラメータが追加されました。デフォルト値は、0 です。**Instr Descriptor** パラメータは、**VISA Resource** という名前に変更されました。
- HP34401A App. Example VI、HP34401A Config Trigger VI、HP34401A Getting Started VI では、**Source** パラメータは、**Trigger Source** という名前に変更されました。
- HP34401A Find Meter VI の **Instr Descriptor** パラメータは、**VISA Resource** という名前に変更されました。
- 「指数関数フィット (Exponential Fit)」 VI には **標準偏差** という新しいパラメータが追加されました。このパラメータは、データポイント $(x(i), y(i))$ の標準偏差、シグマ (i) です。
- **名前** 入力は、「セマフォ破棄 (Destroy Semaphore)」 VI から削除されました。

新規作成、移動、名前変更されたサンプル VI

以下のサンプル VI は、LabVIEW 7.1 では移動、または名前が変更されました。

- examples¥sound ディレクトリの sndExample.llb は sound.llb に、sndExample adv.llb は、sound adv.llb に名前が変更されました。
- TreeView VI は、examples¥comm¥axevent.llb にあります。

LabVIEW 7.1 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 7.1 の新しいサンプル** フォルダを参照してください。

その他

上記の他、LabVIEW 7.1 では、以下の点に変更されました。

- オプションダイアログボックスの**その他**ページにある**起動時にナビゲーションダイアログをスキップ**チェックボックスは、**起動時に LabVIEW ダイアログボックスをスキップ**という名前に変更されました。
- デジタルステート X および Z には、デジタル波形グラフ制御器に新しいラインスタイルがあります。
- 32 ビット符号なし整数および符号付き整数または倍精度浮動小数点値をプローブする場合には、プローブの**条件**ページにある基数を選択してください。この変更が適用されるのは、条件付き倍精度プローブ、条件付き倍精度配列プローブ、条件付き符号付 32 プローブ、条件付き符号付倍長整数配列プローブ、条件付き符号なし 32 プローブ、条件付き符号なし倍長整数配列プローブです。
- FieldPoint I/O ポイント制御器を右クリックして、ショートカットメニューから**未定義の名前を許可**を選択すると、制御器は Measurement & Automation Explorer (MAX) または別の構成ユーティリティで構成していない FieldPoint の項目名を受け入れることができます。
- オプションダイアログボックスの**7.0 での新しい点と変更点**プルダウンメニュー項目は、**7.x での新しい点と変更点**という名前に変更されました。
- **(Linux)** 解析 VI を使用するアプリケーションを実行するターゲットコンピュータの LabVIEW ランタイムエンジンをインストールする場合は、LabVIEW ランタイムエンジンの解析サポートパッケージ (labview71-rte-aal-7.1-1.i386.rpm) のインストールを促すプロンプトが表示されたらはいを入力します。
- LabVIEW 開発システムのサブパネル制御器を右クリックしてショートカットメニューを使用して、サブパネル制御器を構成することができます。LabVIEW 7.0 では、これを行うことができるのは LabVIEW プロフェッショナル開発システムに限られていました。

- Windows および Linux での 2 次元配列の操作のパフォーマンスが向上しました。
- ターゲット OS プロパティには、RTX という新しい値が追加されました。
- VI タイププロパティには、サブシステムという新しい値が追加されました。LabVIEW は、シミュレーションダイアグラムでのみ配置できる LabVIEW シミュレーションモジュールのサブ VI に対してサブシステムの値を返します。
- ウェブパブリッシュツールでは、LabVIEW ランタイムエンジンのどのローカライズ版を使用して組み込み式 VI を表示したら良いかを判断する HTML コードを生成できるようになりました。
- 端子への配線によってワイヤが壊れるような場合、カーソルは変更されません。
- National Instruments ソフトウェア使用許諾契約が変更されました。使用許諾の要件については、LabVIEW 7.1 のインストール CD にある『National Instruments ソフトウェア使用許諾契約』を参照してください。