LabVIEW™アップグレードノート

バージョン 8.2

アップグレードノートでは、Windows、Mac OS、および Linux 対応 LabVIEW をバージョン 8.2 にアップグレードするための手順、その際に起こり得る問題、新しい機能について説明します。

LabVIEW 7.1 以前のバージョンから LabVIEW 8.2 にアップグレードする場合、機能強化、変更点、および追加機能の詳細について『LabVIEW 8.0 アップグレードノート』を参照してください。LabVIEW 7.1 以前のバージョンからアップグレードする場合は、これらのアップグレードノートの他に、『LabVIEW 8.0 アップグレードノート』の「LabVIEW 8.0 の機能および変更点」のセクションを参照してください。『LabVIEW 8.0 アップグレードノート』を開くには、ナショナルインスツルメンツのウェブサイト ni.com/info（英語）で info code として upnote8 と入力してください。

LabVIEW 8.2 の機能の詳細、また LabVIEW プログラミング概念、LabVIEW の使用方法の手順、および LabVIEW の VI、関数、パレット、メニュー、ツール、プロパティ、メソッド、イベント、ダイアログボックスなどの詳細については、『LabVIEW ヘルプ』を参照してください。『LabVIEW ヘルプ』には、ナショナルインスツルメンツが提供している LabVIEW ドキュメントのリソースの一覧も記載されています。『LabVIEW ヘルプ』を開くには、**ヘルプ→ LabVIEW ヘルプを検索**を選択してください。

目次

NATIONAL INSTRUMENTS™

LabVIEW 8.2 へアップグレードする

アップグレードに際しての留意点については、この「LabVIEW 8.2 へアップグレードする」セクションの他に、このドキュメントの「*アップグレードおよび互換性の問題*」セクションの「LabVIEW x.x からアップグレードする」（*x.x* はアップグレードする前のバージョン）も参照してください。

VI を変換する

LabVIEW 4.0 以降のバージョンで保存された VI を LabVIEW 8.2 で開くと、その VI は自動的に変換、コンパイルされます。その後 LabVIEW 8.2 で保存してください。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、そのために余分なメモリリソースが消費されます。

また、未保存の変更（再コンパイルなど）がある VI では、パフォーマンスの効率が著しく低下することがあります。パフォーマンスやメモリに関する問題の詳細については、『LabVIEW ヘルプ』を参照してください。

メモ LabVIEW 8.2 で保存した VI は、旧バージョンの LabVIEW ではロードできません。VI を LabVIEW 8.0 でも実行できるように保存するには、**ファイル→旧バージョン用に保存**を選択してください。LabVIEW 8.0 以前のバージョンでも使用

する VI は、LabVIEW 8.2 で保存する前にバックアップコピーを作成してください。

すべての VI を一度に変換するだけのメモリがコンピュータにない場合は、VI を段階的に変換します。変換する VI の階層を確認して、まず下位階層にあるサブ VI をロードして保存してください。その後、徐々にその上位階層に進んでください。トップレベル VI は最後に開いて変換してください。**ツール→上級→一括コンパイル**を選択して VI のディレクトリを変換することもできます。ただし、一括コンパイルではディレクトリまたは LLB 内の VI はアルファベット順に変換されます。上位の VI から先に変換処理すると一括コンパイルとなり、トップレベルの VI を最初に開くときと同程度のメモリ容量が必要になります。

使用中のメモリ容量は、**ヘルプ→バージョン情報**を選択して確認することができます。

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 8.2 のインストール後、ツールキットおよびアドオンを 8.2 と互換性があることを確認した上で LabVIEW 8.2 のディレクトリに再インストールする必要があります。その場合、旧バージョンの LabVIEW からそれらのツールキットをアンインストールしなければならないことがあります。インストールの詳細については、LabVIEW ツールキットまたはアドオンツールのドキュメントを参照してください。

メモ LabVIEW モジュールのバージョンは、LabVIEW のバージョンと一致していなければなりません。たとえば、LabVIEW Real-Time モジュール 8.2 は、LabVIEW 8.2 と共に使用しなければなりません。

また、既存のツールキット、計測器ドライバ、アドオン VI を LabVIEW 8.2 で使用するために一括コンパイルする必要があります。VI の一括コンパイルの詳細については、このドキュメントの「*VI を変換する*」セクションを参照してください。

以下のツールキット、計測器ドライバ、およびアドオンは、LabVIEW 8.2 で使用するためにアップグレードまたはダウンロードする必要があります。

- LabVIEW アプリケーションビルダを使用する場合は、LabVIEW アプリケーションビルダ 8.2 にアップグレードする必要があります。LabVIEW 8.2 プロフェッショナル開発システムにはアプリケーションビルダ 8.2 が含まれています。LabVIEW アプリケーションビルダの詳細なインストール方法については、**(Windows)** `labview¥readme` ディレクトリにある『LabVIEW アプリケーションビルダ Readme』または **(Mac OS および Linux)** `labview` ディレクトリを参照してください。

- LabVIEW 8.*x* では、VI Analyzer 1.1 を使用しなければなりません。ナショナルインスツルメンツのウェブサイト ni.com/info（英語）にアクセスし、info code として exd8yy と入力すると、Upgrade Advisor にアクセスして VI Analyzer 1.1 を購入することができます。
- LabVIEW 8.*x* で Internet Toolkit 6.0 を使用するには、追加の VI をダウンロードする必要があります。ナショナルインスツルメンツのウェブサイト ni.com/info（英語）にアクセスし、info code として itkver6 と入力すると、必要な VI をダウンロードできます。
- HP/Agilent 34401A デジタルマルチメータ（DMM）の計測器ドライバは、ナショナルインスツルメンツの DMM テンプレートドライバとより類似した仕様になりました。このドライバは、LabVIEW 7.*x* 以前で使用した HP34401A ドライバと互換性がありません。LabVIEW 7.*x* HP34401A ドライバとの互換性が必要な場合は、ナショナルインスツルメンツ計測器ドライバネットワーク（ni.com/idnet）からドライバをダウンロードしてください。

ナショナルインスツルメンツの追加ソフトウェアをアップグレードする

LabVIEW 8.*x* では NI TestStand 3.5 以降を使用する必要があります。ナショナルインスツルメンツのウェブサイト ni.com/info（英語）にアクセスし、info code として exd8yy と入力すると、Upgrade Advisor にアクセスして NI TestStand 3.5 以降を購入することができます。

旧バージョンの LabVIEW から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 以降の LabVIEW は、labview*x.x* ディレクトリ（*x.x* はバージョン番号）にインストールされます。したがって、旧バージョンの LabVIEW をアンインストールせずに LabVIEW 8.2 をインストールできます。

LabVIEW の既存のバージョンを新しいバージョンに置換する

既存バージョンの LabVIEW を新しいバージョンに置換するには、既存バージョンの LabVIEW をアンインストールして、LabVIEW 8.2 のインストーラを実行し、インストールディレクトリを旧バージョンと同じ labview ディレクトリに設定します。

(Windows) コントロールパネルの「アプリケーションの追加と削除」を使用して既存の LabVIEW のバージョンを削除して、既存の LabVIEW のバージョンを LabVIEW 8.2 に置換することもできます。なお、アンイン

ストーラを実行した場合、labview ディレクトリにあるユーザが作成したファイルは削除されません。

メモ LabVIEW をアンインストールまたは再インストールすると、vi.lib ディレクトリにある .llb ファイル（.llb ファイルに保存されている VI および制御器を含む）もアンインストールされます。したがって、VI や制御器を**制御器**パレットや**関数**パレットに追加する際は、その VI または制御器を user.lib ディレクトリに保存してください。

旧バージョンの LabVIEW の環境設定をコピーする

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

注意 LabVIEW 8.2 環境設定ファイルを旧バージョンのファイルと置換する場合は、LabVIEW に追加された環境設定を旧バージョンに上書きします。

LabVIEW 8.2 をインストールした後、LabVIEW 環境設定ファイルを LabVIEW 8.2 ディレクトリにコピーします。

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

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

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

旧バージョンの LabVIEW の user.lib をコピーする

旧バージョンの LabVIEW の user.lib ディレクトリのファイルを使用するには、旧バージョンがインストールされている labview ディレクトリのファイルをコピーします。LabVIEW 8.2 をインストールしたら、ファイルを LabVIEW 8.2 ディレクトリの user.lib ディレクトリにコピーします。

アップグレードおよび互換性の問題

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

LabVIEW 8.2 の既知の問題、その他の互換性に関する問題、最新追加機能については、labview ディレクトリの readme.html を参照してください。

LabVIEW 8.0 からアップグレードする

LabVIEW 8.0 から LabVIEW 8.2 にアップグレードする際、以下のような互換性に関する問題が発生する可能性があります。

サポートされているプラットフォーム

LabVIEW 8.2 では、サポートするプラットフォームが以下のように変更されました。

- LabVIEW 8.2 では Windows XP x64 はサポートされていません。
- LabVIEW 8.2.x では、Mac OS X 10.3.8 以前のバージョンはサポートされていません。
- LabVIEW 8.2 では、Intel プロセッサを搭載した Macintosh コンピュータが一部サポートされています。Macintosh コンピュータのサポートに関する詳細情報を参照するには、ナショナルインスツルメンツのウェブサイト `ni.com/info`（英語）にアクセスし、info code として `macintel` と入力してください。

システム要件

(Mac OS) LabVIEW 8.2 では、LabVIEW の最小インストールには 500MB 以上のディスク容量、完全インストールには 700MB 以上のディスク容量が必要です。

(Linux) LabVIEW 8.2 の最小インストールには 430MB 以上のディスク容量、完全インストールには 620MB 以上のディスク容量が必要です。

印刷版ドキュメント

以下のドキュメントは、LabVIEW 8.2 では変更されませんでした。したがって、LabVIEW 8.2 での変更点が反映されていない可能性があります。

- *LabVIEW クイックリファレンスカード*
- *LabVIEW 基本機能*—『LabVIEW 基本機能』マニュアルは『LabVIEW ヘルプ』の**基本機能**ブックのサブセットです。最新情報は、『LabVIEW ヘルプ』の**目次**タブでこの**基本機能**ブックを開いて参照できます。

VI と関数の動作変更

以下の VI および関数の動作は LabVIEW 8.2 で変更されました。

アプリケーションインスタンス間でデータ交換する

LabVIEW 8.2 では、「キュー取得（Obtain Queue）」、「ノーティファイア取得（Obtain Notifier）」、「ユーザイベントを作成（Create User Event）」、「セマフォ生成（Create Semaphore）」、「ランデブー生成（Create Rendezvous）」といった関数を使用して LabVIEW アプリケーションインスタンス間でデータを交換することはできません。あるアプリ

ケーションインスタンスで取得したキュー、ノーティファイア、ユーザイベント、セマフォ、ランデブーリファレンスは、別のアプリケーションインスタンスで使用できません。

「固有ベクトル変換」VI

「固有ベクトル変換（Back Transform Eigenvectors）」VI の**ジョブ**、**指標 - 低**、**指標 - 高**、**スケール**が必須入力になりました。

「DataSocket 書き込み」関数

LabVIEW 8.0.1 では、「DataSocket 書き込み（DataSocket Write）」関数のデフォルト動作が非同期に変更されました。LabVIEW 8.0 と LabVIEW 8.2 の両方をコンピュータにインストールしている場合、`labview¥examples¥Shared Variable` ディレクトリの DataSocket API クライアント VI サンプルは、ユーザによって停止されるとエラーを返します。このサンプルを LabVIEW 8.2 で使用するためには、LabVIEW 8.0 を 8.0.1 にアップデートする必要があります。

ファイル I/O VI

「スプレッドシートファイルに書き込む（Write To Spreadsheet File）」VI と「スプレッドシートファイルから読み取る（Read From Spreadsheet File）」VI は、多態性 VI です。「スプレッドシートファイルに書き込む」VI は、**形式**入力に配線する値に適応します。「スプレッドシートファイルから読み取る」VI には、DBL、I64、文字列のインスタンスがあります。

「GPIB ステータス」関数

LabVIEW 8.0 では、「GPIB ステータス（GPIB Status）」関数は、**エラー入力**でエラーを受け取ると実行されませんでした。LabVIEW 8.2 では、「GPIB ステータス」関数は、**エラー入力**でエラーを受け取っても実行されます。

「ヒストグラム」VI

「ヒストグラム（Histogram）」VI の**間隔**入力のデフォルト値が `10` に変更されました。

「VI リファレンスを開く」関数

「VI リファレンスを開く（Open VI Reference）」関数の**オプション**入力では、デフォルト動作として見つからないリファレンス VI のサブ VI を検索する画面が表示されるようになりました。新しい値 0x20 によって、**検索**ダイアログボックスや見つからないリファレンス VI の検索画面が表示されないよう設定できます。

「多項式の根」VI

P(x) がゼロ以外の定数と等しい場合、「多項式の根（Polynomial Roots）」VI はエラーを返しません。ただし、**P(x)** が 0 の場合、「多項式の根」VI はエラー -20111 を返します。この VI の入力多項式係数をすべてゼロにすることはできません。

「ランプパターン」VI

「ランプパターン（Ramp Pattern）」VI で、**サンプル**が 1 で**最後を除外？**が TRUE の場合、VI によって**開始**の要素を 1 つ含む配列がエラーなしに返されるようになりました。LabVIEW8.0 では、これらの条件では VI によってエラーが返されていました。

「シンプルレジストリ値読み取り」VI

LabVIEW 8.0 では、VI が平坦化された文字列の配列で使用する REG_MULTI_SZ 文字列形式が正しく処理されませんでした。したがって、「シンプルレジストリ値読み取り（Read Registry Value Simple）」VI ではこのタイプのデータを処理するパーサを記述する必要がありました。LabVIEW 8.2 では、「シンプルレジストリ値読み取り」VI はこのタイプのデータを「シンプルレジストリ値書き込み（Write Registry Value Simple）」VI で使用されているデータと同じ形式で返します。つまり、パーサを記述する必要はなくなりました。LabVIEW 8.2 においてこれらの VI でパーサを記述すると、「シンプルレジストリ値読み取り」VI が不正なデータを返す原因となります。

「複数波形のリサンプル (単発)」VI

「複数波形のリサンプル (単発)（Resample Waveforms (single shot)）」VI の**開区間？**入力のデフォルト値は TRUE から FALSE に変更され、閉区間が選択されるようになりました。既存のコードをこれに合わせて更新しないと、VI から予想される値が返されない可能性があります。

サウンド VI

「サウンド入力読み取り（Sound Input Read）」VI と「サウンドファイル簡易読み取り（Sound File Read Simple）」VI では、**データ**出力の **t0** コンポーネントが最初に読み取られたサンプルのタイムスタンプを返すようになりました。LabVIEW は最初のサンプル読み取りの初回時間を概算します。

連続的なサウンドタスクにおいて、サウンドを停止するために「サウンド出力ストップ（Sound Output Stop）」VI を呼び出す必要がなくなりました。

「サウンド出力待機（Sound Output Wait）」VI が、**連続サンプル**モードと**有限サンプル**モードで動作するようになりました。

波形 VI

LabVIEW 8.2 では、以下の波形 VI が変更されました。

- 基本レベルトリガ検出（Basic Level Trigger Detection）」VI—この VI の両方のインスタンスにおいて、**勾配**入力が**トリガスロープ**に変更されました。
- 「部分波形取得（Get Waveform Subset）」VI—WDT 部分波形取得 DBL、WDT 部分波形取得 CDB、WDT 部分波形取得 EXT、WDT 部分波形取得 I16、WDT 部分波形取得 I32、WDT 部分波形取得 I8、WDT 部分波形取得 SGL というインスタンスが追加されました。**開始 / 遅延形式**入力の**絶対時間**オプションが廃止されました。**開始**入力は**開始サンプル / 時間**に、**実際の開始**出力は**実際の開始サンプル / 時間**にそれぞれ変更されました。
- 「波形時間配列取得（Get Waveform Time Array）」VI—**X 配列**出力のデータタイプが倍精度浮動小数点数からタイムスタンプに変更されました。
- 「Y 値取得（Get Y Value）」VI—この VI と対応する多態性インスタンスは、「XY 値を取得（Get XY Value）」VI に名前が変更されました。「XY 値を取得」VI では、**X 値**出力が追加され、**データ値**出力が **Y 値**に変更されました。
- 「波形サンプル数（Number Of Waveform Samples）」VI—WDT 波形サンプル数 DBL、WDT 波形サンプル数 CDB、WDT 波形サンプル数 EXT、WDT 波形サンプル数 I16、WDT 波形サンプル数 I32、WDT 波形サンプル数 I8、WDT 波形サンプル数 SGL というインスタンスを持つ多態性 VI です。
- 「ファイルから波形を読み取り（Read Waveform from File）」VI—EOF エラーの場合、**エラー出力**に TRUE のエラーステータスを返します。
- 「部分置換（Replace Subset）」VI—**開始**入力は**開始サンプル / 時間**に、**実際の開始値**出力は**実際の開始サンプル / 時間**にそれぞれ名前変更されました。
- 「デジタルパターン検索（Search for Digital Pattern）」VI—**開始**入力が**開始指標 / 時間**に変更されました。
- 「波形検索（Search Waveform）」VI—**最良フィット時間**出力と**フィット時間**出力のデータタイプが倍精度浮動小数点数からタイムスタンプに変更されました。
- 「波形最大最小（Waveform Min Max）」VI—**最小時間**出力と**最大時間**出力のデータタイプが倍精度浮動小数点数からタイムスタンプに変更されました。
- 「波形から XY ペア（Waveform to XY Pairs）」VI—**XY ペア**出力の **x** 要素のデータタイプが倍精度浮動小数点数からタイムスタンプに変更されました。

プロパティ、メソッド、イベントの動作変更

以下のプロパティ、メソッド、イベントの動作は LabVIEW 8.2 で変更されました。

- ActiveX GetVIReference メソッドの**オプション**入力では、デフォルト動作として見つからないリファレンス VI のサブ VI を検索する画面が表示されるようになりました。新しい値 `0x20` によって、**検索**ダイアログボックスや見つからないリファレンス VI の検索画面が表示されないよう設定できます。
- プロジェクト項目クラスの項目を追加メソッドは、プロジェクトで開かれていないライブラリにシェア変数を追加しようとするとエラーを返します。
- VI を実行メソッドの**リファレンスを自動破棄**入力が TRUE で、このメソッドがエラーを返す場合、LabVIEW はそのリファレンスを破棄しません。
- アプリケーション : 言語プロパティの値として、LabVIEW 環境の言語が簡体中国語であることを示す `zh-cn` が追加されました。
- 配列データタイプをリファレンスによって渡す .NET メソッドが、データを実際のデータタイプとして渡すようになりました。配列データタイプをリファレンスによって渡す .NET メソッドが、データを実際のデータタイプとして渡すようになりました。
- デジタル表、複数列リストボックス、表、ツリー制御器の各クラスの編集位置プロパティは、制御器のテキストに何も編集が行われていない場合、(–2, –2) という値を返します。リストボックスクラスの行を編集プロパティは、制御器のテキストに何も変更が行われていない場合、-2 という値を返します。
- LabVIEW 8.0 では、パネルアップデートを延期プロパティはサブパネルのフロントパネルの更新を延期しませんでした。LabVIEW 8.2 では、このプロパティがサブパネルを正常に処理します。
- 「アプリケーションの終了」イベントと「アプリケーションの終了 ?」イベントは、「アプリケーションインスタンスを閉じる」イベントと「アプリケーションインスタンスを閉じる ?」イベントに変更されました。「アプリケーションインスタンスを閉じる」イベントは、LabVIEW プロジェクトの外で実行されている VI 内で使用された場合に、LabVIEW をユーザインタフェースから、またはプログラム的に終了すると発生します。「アプリケーションの終了 ?」イベントは、LabVIEW をユーザインタフェースから終了すると生成されます。「アプリケーションインスタンスを閉じる」イベントと「アプリケーションインスタンスを閉じる ?」イベントは、LabVIEW プロジェクト内で実行されている VI で登録すると、アプリケーションインスタンスが閉じたとき、またはユーザによって LabVIEW が終了されたときに発生します。

廃止予定のプロパティ、メソッド、イベント

LabVIEW 8.2 は以下のプロパティ、メソッド、イベントをサポートしていません。

- LabVIEW 8.2 はコネクタペーンのプロパティをサポートしていません。
- LabVIEW 8.*x* は、変数クラスのデータタイププロパティをサポートしていません。代わりに、変数クラスのデータタイプ（バリアント）プロパティを使用してください。

名前変更されたプロパティ、メソッド、イベント

以下のプロパティ、メソッド、イベントの名前は LabVIEW 8.2 で変更されました。

クラス	LabVIEW 8.0 での名前	LabVIEW 8.2 での名前	タイプ
アプリケーション	スレーブから切断	LVRT: スレーブから切断	メソッド
アプリケーション	アプリケーションの終了	アプリケーションインスタンスを閉じる	イベント
アプリケーション	アプリケーションの終了？	アプリケーションインスタンスを閉じる？	イベント
強度グラフ、ミックスドシグナルグラフ、波形グラフ	カーソルパレット表示	カーソル凡例表示	プロパティ
ライブラリ	ライブラリタグを削除	ライブラリタグ : 削除	メソッド
ライブラリ	アイコンを取得	アイコン : 取得	メソッド
ライブラリ	ライブラリタグを取得	ライブラリタグ : 取得	メソッド
ライブラリ	ライブラリタグ名を取得	ライブラリタグ : 名前を取得	メソッド
ライブラリ	ロック状態を取得	ロック状態 : 取得	メソッド
ライブラリ	ソーススコープを取得	ソーススコープ : 取得	メソッド
ライブラリ	保存	保存 : ライブラリ	メソッド
ライブラリ	コピーを保存	保存 : コピー	メソッド
ライブラリ	アイコンを設定	アイコン : 設定	メソッド
ライブラリ	ライブラリタグを設定	ライブラリタグ : 設定	メソッド
ライブラリ	ロック状態を設定	ロック状態 : 設定	メソッド
ライブラリ	ソーススコープを設定	ソーススコープ : 設定	メソッド

クラス	LabVIEW 8.0 での名前	LabVIEW 8.2 での名前	タイプ
リストボックス、複数列リストボックス、ツリー制御器	ドラッグ / ドロップ : 項目のドラッグを許可	ドラッグ / ドロップ : ドラッグを許可	プロパティ
パスと文字列	ドロップを許可	ドロップを許可	プロパティ
プロジェクト項目	タグを削除	タグ : 削除	プロパティ
プロジェクト項目	タグを取得	タグ : タグを取得	プロパティ
プロジェクト項目	タグ名を取得	タグ : 名前を取得	プロパティ
プロジェクト項目	XML タグを取得	タグ :XML タグを取得	プロパティ
プロジェクト項目	タグを設定	タグ : タグを設定	プロパティ
プロジェクト項目	XML タグを設定	タグ :XML タグを設定	プロパティ
プロジェクト項目	ライブラリ項目タイプ文字列	ライブラリ項目タイプ : 文字列	プロパティ
プロジェクト項目	ライブラリ項目タイプ	ライブラリ項目 : タイプ	プロパティ

LabVIEW 7.x からアップグレードする

LabVIEW 7.*x* から LabVIEW 8.2 にアップグレードする際、以下のような互換性に関する問題が発生する可能性があります。アップグレードの際に起こる可能性のある他の問題点については、このドキュメントの「*LabVIEW 8.0 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 7.*x* から 8.0 までの各バージョンの『LabVIEW アップグレードノート』（ni.com/jp/manuals）を参照してください。

メモ LabVIEW 8.2 では、『LabVIEW クイックリファレンスカード』と『LabVIEW 基本機能』は改訂されませんでした。これらのドキュメントの PDF 版は、labview¥manuals ディレクトリにあります。これらのドキュメントの詳細については、このドキュメントの「*LabVIEW 8.0 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。

サポートされているプラットフォーム

LabVIEW 8.*x* では、サポートするプラットフォームが以下のように変更されました。

- LabVIEW7.1 以降では、Windows Me/98/95 はサポートされていません。LabVIEW8.*x* では Windows NT もサポートされていません。

- LabVIEW 8.*x* は、Mac OS X 10.2 以前のバージョンをサポートしていません。
- LabVIEW 8.*x* は Sun Solaris をサポートしていません。

システム要件

LabVIEW 7.*x* は少なくとも 128MB の RAM を必要としますが、256MB の RAM を推奨します。LabVIEW 8.*x* は少なくとも 256MB の RAM を必要としますが、512MB の RAM を推奨します。

LabVIEW 7.*x* は 800 × 600 ピクセルの画面の解像度を必要としますが、1024 × 768 ピクセルの画面の解像度を推奨します。LabVIEW 8.*x* には、1024 × 768 ピクセルの画面の解像度が必要です。

Windows

LabVIEW 7.*x* には、Pentium III 以降または Celeron 600MHz 以降のプロセッサが必要ですが、Pentium 4 以降のプロセッサを推奨します。LabVIEW 8.*x* には、Pentium III または Celeron 866MHz 以上のプロセッサが必要ですが、Pentium 4 以上のプロセッサを推奨します。

LabVIEW 7.*x* では、LabVIEW の最小インストールには最低 130MB のディスク容量、完全インストールには 550MB のディスク容量が必要です。LabVIEW 8.*x* インストールを完了するには 1.2 GB のディスク容量が必要です。

Mac OS

LabVIEW 7.*x* では、LabVIEW の最小インストールには最低 280MB のディスク容量、完全インストールには 350MB のディスク容量が必要です。LabVIEW 8.2 では、LabVIEW の最小インストールには 500MB 以上のディスク容量、完全インストールには 700MB 以上のディスク容量が必要です。

Linux

LabVIEW 7.*x* には、Pentium III 以降または Celeron 600MHz 以降のプロセッサが必要ですが、Pentium 4 以降のプロセッサを推奨します。LabVIEW 8.*x* には、Pentium III または Celeron 866MHz 以上のプロセッサが必要ですが、Pentium 4 以上のプロセッサを推奨します。

LabVIEW 7.*x* では、LabVIEW の最小インストールには最低 200MB のディスク容量、完全インストールには 300MB のディスク容量が必要です。LabVIEW 8.2 の最小インストールには 430MB 以上のディスク容量、完全インストールには 620MB 以上のディスク容量が必要です。

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

以降を推奨します。LabVIEW 8.*x* には、GNU C Library バージョン 2.2.4 以降が必要です。

LabVIEW 7.*x* は、Red Hat Linux 7.0 以降、Mandrake Linux 8.0 以降、SuSE Linux 7.1 以降、または Debian Linux 3.0 以降で動作します。LabVIEW 8.*x* では、Red Hat Enterprise Linux WS 以降、MandrakeLinux/Mandriva 10.0 以降、または SuSE Linux 9.1 以降で動作します。

カスタムパレットセット

LabVIEW 8.*x* はカスタムパレットセットをサポートしていません。カスタムパレットセットを使用せずにパレットセットを編集できます。LabVIEW 8.0 でのパレットの詳細な変更点については、ナショナルインスツルメンツのウェブサイト `ni.com/info`（英語）にアクセスし、info code として `lv8palette` と入力してください。

VI と関数の動作変更

LabVIEW 7.1 および 8.0 では以下の VI および関数の動作が変更されました。

.NET VI と .NET アプリケーション

LabVIEW 8.*x* で .NET の関数およびアプリケーションを使用するには、.NET Framework 1.1 Service Pack 1 またはそれ以降が必要です。.NET Framework 1.1 Service Pack 1 をインストールする前に Microsoft .NET Framework 1.1 Hotfix KB886904 を削除する必要があります。

LabVIEW 7.*x* 形式で保存された .NET VI を LabVIEW 8.*x* でロードすると、アセンブリファイルがこの VI と同じディレクトリに存在しても、また LabVIEW 7.*x* で**ツール→上級→ .NET アセンブリリファレンス**を選択してアセンブリファイルを登録済みであっても、VI が参照するアセンブリを検索するためのプロンプトが表示されることがあります。

解析 VI のアルゴリズム

LabVIEW 7.1 以降では、解析 VI で BLAS/LAPACK アルゴリズムが使用されています。これにより、生成される結果がより正確になりました。LabVIEW 8.*x* では、これらの VI は**数学**および**信号処理**パレットにあります。

「信号を付加」Express VI

LabVIEW 7.*x* では、「信号を付加（Append Signals）」Express VI の**入力信号 A** が空の場合、または配線されていない場合、単一信号または組み合わせた信号を**入力信号 B** に配線すると、**付加信号**出力が空になります。LabVIEW 8.*x* では、**入力信号 A** が空または配線されていない場合に、単一信号を**入力信号 B** に配線すると、Express VI は**入力信号 B** を返しま

す。連結信号のみを**入力信号 B** に配線する場合には、連結信号で各信号は次の信号に加えられ、結果として 1 つの信号が作成されます。

比較関数

LabVIEW 7.*x* 以前では、比較関数を使用してバリアントデータを比較すると、LabVIEW は 2 つのバリアントの長さを比較して、バリアントをビットごとに比較しました。LabVIEW 8.*x* では、バリアントの実際の入力情報をエンコードした入力コード付きのバリアントデータの比較を開始し、その他の入力特有の属性を比較します。

「内積」VI

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

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

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

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

ここで、y_i^{*} は、y_i の複素共役です。

「Easy Text Report」VI（Mac OS および Linux）

「Easy Text Report」VI のコネクタペーンが変更されました。LabVIEW 8.*x* で、LabVIEW 7.*x* 以前のバージョンで保存された「Easy Text Report」VI を含む VI を開くには、サブ VI を右クリックして、ショートカットメニューから **Relink To SubVI** を選択する必要があります。

「文字列にフォーマット」関数

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

LabVIEW 8.*x* では、これらの形式指定子の構文要素は入力を文字列に変換する前に、この関数で浮動小数点入力を 64 ビット整数に四捨五入します。

「数値結合」関数

LabVIEW 7.*x* 以前では、「数値結合（Join Numbers）」関数は 32 ビット整数入力を 16 ビット整数に強制的に変更し、1 つの 32 ビット整数を作成していました。LabVIEW 8.*x* では、「数値結合」関数で 32 ビット整数入力を組み合わせて 1 つの 64 ビット整数を生成できるようになりました。

メモ LabVIEW 7.*x* の VI を LabVIEW 8.*x* で開くと、32 ビット整数入力が 16 ビット整数に強制的に変更されます。

数学 VI と行列

LabVIEW 8.*x* では、**数学** VI では行列データタイプがサポートされています。LabVIEW 7.*x* で作成された VI を LabVIEW 8.*x* で開くと、その VI に行列データタイプが使用される関数に配線された数学 VI があると、赤い 7.*x* グリフが関数上に表示されます。これは、この関数が LabVIEW 7.*x* と同じ動作をすることを示します。

文字列 / 数値変換関数

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

LabVIEW 8.*x* では、これらの関数は入力を文字列に変換する前に、浮動小数点入力を 64 ビット整数に四捨五入します。ただし、LabVIEW 8.*x* で LabVIEW 7.*x* の VI を開く場合は、浮動小数点入力を 32 ビット整数に四捨五入することで互換性と機能を維持します。

「VI リファレンスを開く」関数

LabVIEW 7.*x* では、「VI リファレンスを開く（Open VI Reference）」関数の **VI パス**入力がパスで、メモリ内に同じ名前のの VI が存在する場合、メモリ内の VI のパスが指定したパスと一致しなくても、VI へリファレンスが返されます。

LabVIEW 8.*x* では、「VI リファレンスを開く」関数の **VI パス**入力が文字列で、**VI パス**がメモリ内の VI の限定されたファイル名に一致する場合にのみ、ターゲット上で VI が開きます。**VI パス**がパスの場合、同じターゲット上の同じパスでメモリ内の VI を検索します。一致するパスで VI を検索できない場合は、指定したパスでディスクから VI をロードしようとします。ファイルが見つからない場合、またはファイルがメモリやターゲット内の他の VI と競合する場合、エラーが発生します。

「クイックスケール」VI

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

を、または NaN の配列として **Yij=Xij/Max|X|** を返します。LabVIEW 8.*x* では、「クイックスケール（Quick Scale）」VI の **X** 入力が 0 の配列の場合、この VI は **max|X|** を **0** として返し、**Y(i)=X(i)/Max|X|** または **Yij=Xij/Max|X|** を 0 の配列として返します。

「キーを読み取る」VI

In LabVIEW 7.*x* 以前では、「キーを読み取る（Read Key）」VI を使用して、Shift-JIS でエンコードされた日本語のマルチバイト文字の文字を読み取ることができました。このためには、1 または <Shift-JIS> を、**マルチバイトエンコーディング**入力に配線する必要がありました。LabVIEW 8.*x* では、「キーを読み取る」VI は、オペレーティングシステムの地域設定を適切なエンコーディングに設定すると、デフォルトでエンコードされた文字列のマルチバイト文字が読み取られます。

「スケール」VI

LabVIEW 7.1 以前では、「スケール 1D（Scale 1D）」VI または「スケール 2D（Scale 2D）」VI の **X** 入力がゼロの配列の場合、この VI は **0** として**スケール**、**0** として**オフセット**、NaN の配列として **Y=(X- オフセット)/ スケール**を返します。LabVIEW 8.*x* では、スケール VI の **X** 入力がゼロの配列の場合、この VI は**スケール**を **1**、**オフセット**を **0**、**Y=(X- オフセット)/ スケール**として 0 の配列を返します。

セマフォ VI

LabVIEW 7.*x* では、「セマフォ解放（Release Semaphore）」VI および「セマフォ取得（Acquire Semaphore）」VI は**エラー入力**でエラーを受け取ると実行されませんでした。LabVIEW 8.*x* では、これらの VI は**エラー入力**でエラーを受け取っても実行が試みられます。ただし、LabVIEW 7.*x* の VI を LabVIEW 8.*x* で開くと、LabVIEW 7.*x* の機能が維持されます。

SMTP E メール VI

LabVIEW 7.*x* 以前では、SMTP E メール VI の**文字セット**入力に値を書き込んで文字セットを指定できました。LabVIEW 8.*x* では、STMP E メール VI はメッセージがシステムの文字セットで書かれていると想定します。これらの VI は、E メールを送信する前にメッセージを UTF-8 形式に変換します。SMTP E メール VI の**文字セット**または**字訳**パラメータは廃止されました。

「複素数をソート」VI

LabVIEW 7.*x* 以前では、「複素数をソート（Sort Complex Numbers）」VI の**メソッド**入力を**振幅**に設定すると、同じ振幅を持つ要素のシーケンスが変更されませんでした。LabVIEW 8.*x* では、**メソッド**を**振幅**に設定すると、同じ振幅を持つ要素をまず実部に基づいてソートし、次に虚部に基づいてソートします。

「単位ベクトル」VI

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

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

LabVIEW 8.*x* では、「単位ベクトル」VI によって入力ベクトルのノルムが以下の式に基づいて計算されます。

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

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

ユーザ VI

`labview¥help`、`labview¥project`、または `labview¥wizard` ディレクトリに配置する VI は、**ヘルプ**、**ツール**、および**ファイル**メニューにそれぞれ表示されます。LabVIEW 8.*x* 以降ではこれらのディレクトリに配置する VI をプライベートアプリケーションインスタンスで開くため、LabVIEW 7.*x* 以前では、これらの VI が LabVIEW 8.0 のように動作しないことがあります。

すべてのアプリケーションインスタンスでメモリ内のすべてのユーザ VI のリストを生成するには、`labview¥vi.lib¥Utility¥allVIsInMemory.llb` にある VIMemory Get VIs in Memory VI を使用してください。現在のアプリケーションインスタンスへのリファレンスを作成するには、`labview¥vi.lib¥Utility¥allVIsInMemory.llb` にある Get User Application Reference VI を使用してください。アプリケーションインスタンスの詳細については、『LabVIEW ヘルプ』を参照してください。

廃止予定の VI および関数

LabVIEW 8.*x* は以下の VI および関数をサポートしていません。

- LabVIEW 7.1 以降では、「多項式実ゼロカウンタ（Polynomial Real Zero Counter）」VI はインストールされません。「多項式実ゼロ（複数）カウンタ（Polynomial Real Zeros Counter）」VI を代用してください。
- **(Mac OS)** LabVIEW 7.1 以降では、PPC VI はインストールされません。TCP VI を代用してください。
- LabVIEW 8.*x* は「QR 因子分解（QR Factorization）」VI をサポートしていません。「QR 分解（QR Decomposition）」VI を代用してください。

- LabVIEW 8.*x* は、「レーベンバーグ・マルカート（Levenberg Marquardt）」VI または「非線形レーベンバーグ • マルカートフィット（Nonlinear Lev-Mar Fit）」VI をサポートしていません。「非線形カーブフィット（Nonlinear Curve Fit）」VI を代用してください。
- LabVIEW 8.*x* では、「VISA ステータス記述（VISA Status Description）」関数が**関数**パレットに表示されなくなりました。「シンプルエラー処理 (Simple Error Handler)」VI または「一般エラー処理 (General Error Handler)」VI を代用してください。
- LabVIEW 8.*x* は、「カイ 2 乗分布（Chi Square Distribution）」、「F 分布（F Distribution）」、「正規分布（Normal Distribution）」、および「T 分布（T Distribution）」VI をサポートしていません。代わりに、「連続 CDF（Continuous CDF）」VI のカイ 2 乗、F、正規、およびステューデント t インスタンスをそれぞれ使用してください。
- LabVIEW 8.*x* は、「逆カイ 2 乗分布（Inv Chi Square Distribution）」、「逆 F 分布（Inv F Distribution）」、「逆正規分布（Inv Normal Distribution）」、および「逆 T 分布（Inv T Distribution）」VI をサポートしていません。代わりに、「連続 CDF」VI のカイ 2 乗、F、正規、およびステューデント t インスタンスをそれぞれ使用してください。
- LabVIEW 8.*x* では、「1D 線形評価（1D Linear Evaluation）」VI および「2D 線形評価（2D Linear Evaluation）」VI は、**関数**パレットに表示されなくなりました。「線形評価（Linear Evaluation）」VI を代用してください。
- LabVIEW 8.*x* では、「1D 多項式評価（1D Polynomial Evaluation）」VI および「2D 多項式評価（2D Polynomial Evaluation）」VI は、**関数**パレットに表示されなくなりました。「多項式評価（Polynomial Evaluation）」VI を代用してください。
- LabVIEW 8.*x* では、「1D 直交座標から極座標へ変換（1D Rectangular to Polar）」VI および「1D 極座標から直交座標へ変換（1D Polar to Rectangular）」VI が**関数**パレットに表示されなくなりました。代わりに「直交座標を極座標に変換（Re/Im To Polar）」関数および「極座標を直交座標に変換（Polar To Re/Im）」関数を使用してください。
- LabVIEW 8.*x* では、「高調波アナライザ（Harmonic Analyzer）」VI が**関数**パレットに表示されなくなりました。**THD** または**コンポーネントレベル**出力を測定するには「高調波ひずみ解析（Harmonic Distortion Analyzer）」VI を使用するか、**SINAD** または **THD+ ノイズ**出力を測定するには「SINAD アナライザ（SINAD Analyzer）」VI を使用してください。
- LabVIEW 8.*x* では、「ネットワーク関数 (平均)（Network Functions (avg)）」VI が**関数**パレットに表示されなくなりました。「周波数応答関数 (振幅 - 位相)（Frequency Response Function (Mag-Phase)）」、「周波数応答関数 (実 - 虚)（Frequency Response Function (Real-Im)）」、「クロススペクトル (振幅 - 位相)

（Cross Spectrum (Mag-Phase)）」、または「クロススペクトル (実 - 虚) （Cross Spectrum (Real-Im)）」VI を使用してください。

- LabVIEW 8.*x* では、「パルスパラメータ（Pulse Parameters）」VI が**関数**パレットに表示されなくなりました。「過度計測（Transition Measurements）」VI を代用して**スルーレート**、**持続時間**、**オーバーシュート**または、**プレシュート**出力を測定し、「パルス計測（Pulse Measurements）」VI を使用して**周期**、**パルス持続時間**、または**デューティサイクル**出力を測定し、「振幅とレベル（Amplitude and Levels）」VI を使用して**振幅**、**高状態レベル**、または**低状態レベル**出力を測定します。
- LabVIEW 8.*x* では、「伝達関数（Transfer Function）」関数が**関数**パレットに表示されなくなりました。「周波数応答関数 (振幅 - 位相)」または「周波数応答関数 (実 - 虚)」VI を代用してください。
- LabVIEW 8.*x* では、「NI DIAdem レポートウィザード（NI DIAdem Report Wizard）」Express VI が**関数**パレットに表示されなくなりました。「DIAdem レポート（NI DIAdem Report）」Express VI を代用してください。
- LabVIEW 8.*x* では、VISA リソース名定数および IVI 論理名定数は**関数**パレットに表示されなくなりました。VISA リソース名を指定するには、VISAVI の **VISA リソース名**入力を使用してください。IVI 論理名を指定するには、計測器を初期化する適切なドライバ VI の適切な入力を使用してください。
- LabVIEW 8.*x* では、**関数**パレット上にエラーリング定数がありません。32 ビット符号付き整数定数を代用して、希望するエラーコードを入力します。
- **(Windows および Linux)** LabVIEW 8.*x* では、サウンド VI は**サウンド**パレットに表示されますが、LabVIEW 7.*x* では**関数**パレットにありませんでした。LabVIEW 8.*x* ではサウンド VI を代用してください。LabVIEW 7.*x* で使用されていたサンプルは、LabVIEW 8.*x* では使用されていません。
- **(Mac OS)** LabVIEW 8.*x* のサウンド VI でも、LabVIEW 7.1 のサウンド VI が提供されています。LabVIEW 7.*x* で使用されていたサンプルは、LabVIEW 8.*x* では使用されていません。

ファイル I/O VI および関数

LabVIEW 8.*x* では、「ファイルから文字を読み取り（Read Characters From File）」VI が**関数**パレットに表示されなくなりました。「テキストファイルから読み取る（Read from Text File）」関数を代用してください。

LabVIEW 8.*x* では、「ファイルを開く / 作成 / 置換（Open/Create/Replace File）」VI が**関数**パレットに表示されなくなりました。「ファイルを開く / 作成 / 置換（Open/Create/Replace File）」関数を代用してください。以下の関数には、LabVIEW 7.*x* 以前の

「ファイルを開く / 作成 / 置換（Open/Create/Replace File）」VI の一部の機能が含まれます。

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

LabVIEW 8.*x* では、「ファイルから読み取り（Read File）」および「ファイルに書き込み（Write File）」関数が**関数**パレットに表示されなくなりました。「バイナリファイルから読み取る」および「バイナリファイルに書き込む」関数を代用してください。

LabVIEW 8.*x* では、「文字をファイルに書き込み（Write Characters To File）」VI が**関数**パレットに表示されなくなりました。「テキストファイルに書き込む（Write to Text File）」関数を代用してください。

LabVIEW 8.*x* では、「アクセス権（Access Right）」関数が**関数**パレットに表示されなくなりました。「許可を取得（Get Permissions）」および「許可を選択（Set Permissions）」関数を代用してください。

LabVIEW 8.*x* では、「EOF」関数が**関数**パレットに表示されなくなりました。「ファイルサイズを取得（Get File Size）」および「ファイルサイズを設定（Set File Size）」関数を代用してください。

LabVIEW 8.*x* では、「ディレクトリのリスト（Directory List）」関数が**関数**パレットに表示されなくなりました。「リストフォルダ（List Folder）」関数を代用してください。

LabVIEW 8.*x* では、「範囲をロック（Lock Range）」関数が**関数**パレットに表示されなくなりました。「アクセスを拒否（Deny Access）」関数を代用してください。

ブロックダイアグラムに「新規ディレクトリ（New Directory）」関数を含む LabVIEW 7.*x* で作成した VI を開くと、LabVIEW 8.*x* はその関数を「フォルダを作成（Create Folder）」関数に置換します。**パス**入力で指定したフォルダが存在しない場合は、「フォルダを作成」関数は、「新規ディレクトリ」関数と同じように、エラーを返す代わりにフォルダを作成します。

LabVIEW 8.*x* では、「シーク（Seek）」関数が**関数**パレットに表示されなくなりました。「ファイル位置を取得（Get File Position）」および「ファイル位置を設定（Set File Position）」関数を代用してください。

LabVIEW 8.*x* では、「タイプと作成者（Type and Creator）」関数が**関数**パレット上に表示されなくなりました。「タイプと作成者を取得（Get Type and Creator）」および「タイプと作成者を設定（Set Type and Creator）」関数を代用してください。

LabVIEW 8.*x* では、「ディレクトリのリスト（Directory List）」関数が**関数**パレットに表示されなくなりました。「ボリューム情報を取得（Get Volume Info）」関数を代用してください。

LabVIEW 8.*x* では、「ファイルを開く（Open File）」および「新規ファイル（Write File）」関数が**関数**パレットに表示されなくなりました。「ファイルから一行読み取り（Read Lines From File）」VI は**関数**パレットには表示されませんが、互換性のために LabVIEW に付属されます。

LabVIEW 8.*x* では、「I16 ファイルから読み取り（Read From I16 File）」、「SGL ファイルから読み取り（Read From SGL File）」、「I16 ファイルに書き込み（Write To I16 File）」、および「SGL ファイルに書き込み（Write To SGL File）」VI は**関数**パレットに表示されなくなりました。「バイナリファイルから読み取る」および「バイナリファイルに書き込む」VI を代用してください。

プロパティ、メソッド、イベントの動作変更

LabVIEW 7.1 および 8.0 では以下のプロパティ、メソッド、イベントの動作が変更されました。

アプリケーションプロパティとメソッド

LabVIEW 8.*x* では、一部のアプリケーションプロパティおよびメソッドの動作は、属するアプリケーションインスタンスによって異なります。たとえば、アプリケーション : メモリ内のすべての VI プロパティの動作は、使用するアプリケーションインスタンスによって異なります。このプロパティは、プロパティと同じアプリケーションインスタンスでメモリ内のすべての VI のリストを返します。ただし、アプリケーション : ディレクトリパスプロパティの動作は、使用するアプリケーションインスタンスに依存しません。このプロパティは、アプリケーションがあるディレクトリへの絶対パスを返します。この情報は、各インスタンスによって変更しません。

アプリケーションインスタンスの詳細については、『LabVIEW ヘルプ』を参照してください。

フロントパネル : 開く メソッド

「LabVIEW 7.0 の FP を開く」メソッドは、「LabVIEW 7.1 で旧フロントパネルを開く」に名前が変更されました。LabVIEW7.1 では、これとは異なる「FP を開く」メソッドが使用されています。「FP を開く」メソッドはフロントパネルがすでに開いているとエラーを返しません。
LabVIEW 7.1 の「FP を開く」メソッドは、LabVIEW 8.*x* では「フロントパネル : 開く」に名前が変更されました。LabVIEW 7.0 の「FP を開く」メソッドを使用する VI は、そのメソッドを「フロントパネル : 開く」メソッドに置換する必要があります。

VI を実行メソッド

LabVIEW 7.1 では、VI を実行メソッドの**リファレンスを自動破棄**入力を TRUE に設定すると、VI の実行を停止した後にリファレンスが自動的に破棄されます。LabVIEW 8.*x* 以降では、メソッドがエラーを返すと、ただちにリファレンスを破棄します。ブロックダイアグラムの一部がそのリファレンスに依存している場合、この動作によって VI が壊れることがあります。

キーダウンイベントとキー繰り返しイベント

VI のキーダウン、キーダウン？、キー繰り返し、キー繰り返し？の各イベントおよび制御器の **VKey** データフィールドは、英数字セクションの <Return> キーおよび数値キーパッドの <Enter> キーで異なる値が入力されます。LabVIEW 7.*x* 以前では、<Enter> キーまたは <Return> キーがイベントの 1 つを生成すると、**VKey** データフィールドで **<Enter>** を返します。LabVIEW 8.*x* では、<Enter> キーまたは <Return> キーがイベントの 1 つを生成すると、**VKey** データフィールドで、それぞれ、**<Enter>** または **<Return>** を返します。

(Mac OS) LabVIEW 8.*x* では、ショートカットメニューの <Control>- クリックのみが有効となり、<Command>- クリックのキーの組み合わせは無効です。イベントストラクチャで動作をエミュレートする場合は、VI を変更して新規の動作をエミュレートします。

リストボックスプロパティ

LabVIEW 7.*x* 以前では、リストボックスプロパティの最上行プロパティをリストボックスの一番下の項目より下の行に設定すると、表示されている一番下の行に強制的に変更されます。LabVIEW 8.*x* では、リストボックスの表示項目数は、このプロパティに配線できる行数を制限しません。

LabVIEW 8.*x* は、単数列リストボックスのダブルクリックプロパティをサポートしていません。ダブルクリックされた行を取得メソッドを代用してください。

所有 VI プロパティ

LabVIEW 7.*x* 以前では、所有 VI プロパティはオブジェクトが属する VI へのリファレンスを返します。このリファレンスによって、VI がメモリ内に維持されます。LabVIEW 8.*x* では、所有 VI プロパティが返すリファレンスは、VI をメモリ内に維持しません。所有 VI がメモリから削除されると、このリファレンスは無効になります。リファレンスを明示的に閉じるまでメモリに維持される VI へのリファレンスを取得するには、「VI リファレンスを開く（Open VI Reference）」関数を使用します。

テキストプロパティ

LabVIEW 7.*x* 以前では、テキストプロパティは標準表示で文字列を返します。LabVIEW 8.*x* では、テキストプロパティは文字列をフロントパネルオブジェクトと同じテキスト表示で返します。たとえば、パスワード表示に文字列制御器を表示すると、テキストプロパティはパスワード表示で文字列を返します。

ツリー制御器プロパティ

LabVIEW 7.*x* 以前では、アクティブセルプロパティ : セルサイズ : 高さおよびアクティブセルプロパティ : セルサイズ : 幅プロパティは、ツリー制御器内に各行に 17 ピクセルを返しました。LabVIEW 8.*x* では、アクティブセル : セルサイズ : 高さおよびアクティブセル : セルサイズ : 幅プロパティは、ツリー制御器内に各行に 16 ピクセルを返します。

VI 文字列メソッド

エクスポート VI 文字列メソッドを使用して前のバージョンの LabVIEW からエクスポートした文字列は、VI 文字列 : インポートメソッドを使用する場合、正しく LabVIEW 8.*x* にインポートされない場合があります。

廃止予定のプロパティ、メソッド、イベント

LabVIEW 8.*x* は以下のプロパティ、メソッド、イベントをサポートしていません。

カーソルプロパティ

LabVIEW 8.*x* は、カーソルロックスタイルプロパティをサポートしていません。カーソルモードプロパティを代用してください。

リストボックス、表、デジタル表、ツリー制御器プロパティ、およびイベント

LabVIEW 8.*x* は、複数列リストボックスのセルの前景色プロパティをサポートしていません。アクティブ セル プロパティ : セルフォント : カラープロパティを代用してください。

LabVIEW 8.*x* は、表またはデジタル表のセルの前景色プロパティをサポートしていません。表およびデジタル表については、アクティブセル : セルフォント : カラープロパティを代用してください。

LabVIEW 8.*x* は、ツリー制御器のアクティブ セル プロパティ : 前景カラープロパティをサポートしていません。アクティブ セル プロパティ : セルフォント : カラープロパティを代用してください。

LabVIEW 8.*x* は、ドラッグ、ドラッグ ?、ドロップ、ドロップ ? イベントをツリー制御器クラスでサポートしていません。ドラッグ終了、ドラッグ入る、ドラッグ出る、ドラッグオーバー、ドラッグソースアップデート、ドラッグ開始、ドラッグ開始 ?、ドロップイベントを制御器クラスで代用してください。

名前付き数値プロパティ

LabVIEW 8.*x* は、名前付き数値オブジェクトについて、名前付き数値カラー、名前付き数値カラー : 背景色、名前付き数値カラー : テキストカラーの各プロパティをサポートしていません。テキストカラー、テキストカラー : 背景色、テキストカラー : テキストカラーの各プロパティをそれぞれ代用してください。

パネルプロパティ

LabVIEW 8.*x* は、パネルクラスのカラープロパティをサポートしていません。このプロパティを LabVIEW 8.*x* で使用すると、プロパティは左上隅のペーンのみに適用されます。ペーンカラープロパティをペーンクラスで代用してください。

サブパネルプロパティ

LabVIEW 8.*x* では、サブパネル制御器のスクロールバー表示の構成、サブパネル制御器のフロントパネルのスケールはサブパネルでサブ VI のペーンで行います。

LabVIEW 8.*x* は、サブパネル制御器の X スクロールバー表示プロパティをサポートしていません。ペーンの水平スクロールバー表示プロパティを代用してください。

LabVIEW 8.*x* は、サブパネル制御器の Y スクロールバー表示プロパティをサポートしていません。ペーンの垂直スクロールバー表示プロパティを代用してください。

LabVIEW 8.*x* は、サブパネル制御器のパネルをスケールプロパティをサポートしていません。ペーンのスケールモードを設定メソッドを代用してください。

VI プロパティ、メソッド、イベント

LabVIEW 8.*x* はフロントパネルウィンドウ : 自動中央揃えプロパティをサポートしていません。フロントパネル : 中心メソッドを代用してください。

LabVIEW 8.*x* はフロントパネルウィンドウ : 画面サイズに合わせるプロパティをサポートしていません。フロントパネルウィンドウ : 状態プロパティを代用してください。

LabVIEW 8.*x* は、VI クラスのフロントパネルウィンドウ : 基点プロパティをサポートしていません。このプロパティを LabVIEW 8.*x* で使用すると、プロパティは左上隅のペーンのみに適用されます。ペーンクラスの基点プロパティを代用してください。

LabVIEW 8.*x* は、VI クラスのフロントパネルウィンドウ : スクロールバーを表示プロパティをサポートしていません。このプロパティを LabVIEW 8.*x* で使用すると、プロパティは左上隅のペーンのみに適用されます。ペーンクラスの水平スクロールバー表示および垂直スクロールバー表示プロパティを代用してください。

LabVIEW 8.*x* は、VI クラスのフロントパネルスケールモードを取得またはフロントパネルスケールモードを設定メソッドをサポートしていません。これらのメソッドを LabVIEW 8.*x* で使用すると、メソッドは左上隅のペーンのみに適用されます。ペーンクラスのスケールモードを取得およびスケールモードを設定メソッドを代用してください。

LabVIEW 8.*x* は、VI クラスのマウスダウン、マウスダウン ?、マウス境界内、マウス境界外、マウス移動、マウスアップイベントをサポートしていません。ペーンクラスのマウスダウン、マウスダウン ?、マウス境界内、マウス境界外、マウス移動、マウスアップイベントを代用してください。

アプリケーション項目のタグ

LabVIEW 8.*x* では、以下のアプリケーション項目のタグが削除されました。

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

前のバージョンの LabVIEW で保存され、削除されたタグが含まれているランタイムメニュー（.rtm）ファイルを使用すると、LabVIEW 8.*x* は、**メニューエディタ**ダイアログボックスでファイルを保存すると、自動的に .rtm ファイルからタグを削除します。削除されたアプリケーション項目タグは、LabVIEW によって予約され、ユーザタグとして使用できません。

HiQ のサポート

ナショナルインスツルメンツは、LabVIEW 8.*x* で HiQ 機能を提供していません。アプリケーションで HiQ VI が使用される場合は、代わりに信号処理および数学 VI を使用することを検討してください。

エラーリストウィンドウ

LabVIEW 7.*x* 以前では、**エラーリスト**ウィンドウの **VI リスト**にメモリ内にあるすべての VI のエラーが表示されます。LabVIEW 8.*x* では、**エラーリスト**ウィンドウの**エラーがある項目**セクションに、VI およびライブラリなど、メモリ内にあるすべての項目のエラーが表示されます。2 つ以上の項目が同じ名前の場合、このセクションは各あいまいな項目について特定のアプリケーションインスタンスを表示します。アプリケーションインスタンスの詳細については、『LabVIEW ヘルプ』を参照してください。

VI 文字列ファイルの構文

LabVIEW 8.*x* では、**ツール→上級→文字列をインポート**で VI 文字列ファイルをインポートするか、または VI 文字列 : インポートメソッドを使用すると、<GROUPER></GROUPER> という新規のタグセットを検索します。このタグセットは一緒にグループ化されたフロントパネルオブジェクトを示します。したがって、LabVIEW 8.*x* では、旧バージョンの LabVIEW で保存した VI 文字列ファイルをインポートすることはできません。

LabVIEW 7.1 以前では、プライベートデータの <ITEMS> セクションにリストボックスの文字列の一覧が表示されます。LabVIEW 8.*x* では、プライベートデータの <STRINGS> セクションにリストボックスの文字列の一覧が表示されます。また、LabVIEW 7.1 以前では、リストボックスで、プライベートデータの <LBLABEL> セクションに表示される 1 つのフォントしか使用できません。LabVIEW 8.*x* では、リストボックスで複数のフォントが使用できるようになり、それらのフォントはプライベートデータの <CELL_FONTS> セクションにリストされます。

LabVIEW 7.1 以前では、複数列リストボックスの文字列はデフォルトデータ内に一覧表示されます。ただし、複数列リストボックスのデフォルトデータは整数か整数の配列です。LabVIEW 8.*x* では、複数列リストボックスの文字列はプライベートデータ内に一覧表示されます。

LabVIEW 7.1 以前では、ツリー制御器の文字列とフォントはエクスポートできませんでした。LabVIEW 8.*x* では、ツリー制御器の文字列とフォン

トを、リストボックスや複数列リストボックスと同様の形式でエクスポートできます。

LabVIEW 8.*x* では、エクスポートファイルの各行ではプライベートデータまたはデフォルトデータのタグを 2 つ以上使用できません。また、ネスティングレベルごとに項目がインデントされます。

VI 文字列ファイルを LabVIEW 8.*x* 形式に変換するには、以下の手順に従ってください。

1. 旧バージョンの LabVIEW にある VI 文字列ファイルをインポートします。
2. VI を保存します。
3. LabVIEW 8.*x* で VI をロードします。
4. LabVIEW 8.*x* 形式で VI 文字列ファイルを保存するには、**ツール→上級→文字列をエクスポート**を選択します。

LabVIEW 7.x からタイプデスクリプタデータを双方向に変換する

タイプデスクリプタを保存する形式が LabVIEW 8.*x* で変更されました。LabVIEW 7.*x* は、16 ビットの平坦化表記でタイプデスクリプタを保存します。LabVIEW 8.*x* は、32 ビットの平坦化表記でタイプデスクリプタを保存します。この変更によって、タイプデスクリプタの 64KB サイズの制限がなくなりました。

LabVIEW 8.*x* には、LabVIEW 7.*x* で書き込まれたタイプデスクリプタを読み取り、LabVIEW 7.*x* で読み取りができるタイプデスクリプタを書き込むメカニズムが組み込まれています。「文字列に平坦化（Flatten To String）」関数には **7.x データの変換**ショートカットメニュー項目があります。関数を右クリックしてこのメニュー項目を選択すると、この関数は、入力データを LabVIEW 7.*x* で書き込まれたデータのように処理できます。LabVIEW 8.*x* で、**7.x データの変換**ショートカットメニュー項目を選択したとき**データ文字列**出力が配線されていると、関数上に赤い 7.x グリフが表示されます。このグリフは、データが LabVIEW 7.*x* 形式に、または LabVIEW 7.x 形式から変換中であることを示します。データ変換を回避するには、**7.x データの変換**ショートカットメニュー項目をもう一度選択してチェックマークを外します。

LabVIEW 8.*x* では、LabVIEW 7.*x* 以前のバージョンで保存した VI をロードすると、「文字列に平坦化（Flatten To String）」関数で **7.x データの変換**属性が自動的に設定されます。関数は、LabVIEW 7.*x* 以前と同じように操作されます。VI に LabVIEW 8.*x* のタイプデスクリプタ形式を使用する場合は、「文字列に平坦化」関数を右クリックし、ショートカットメニューから **7.x データの変換**を選択してチェックマークを外します。LabVIEW 7.*x* 以前のバージョンで書き込まれたデータを含むファイルを VI で操作する必要がなく、LabVIEW 7.*x* 以前のバージョンで実行している VI にデータを送受信することがない場合には、LabVIEW 8.*x* のタイプ

デスクリプタ形式を使用します。LabVIEW の今後のバージョンでは、前回のタイプデスクリプタ形式のサポートは継続されない可能性があります。

LabVIEW ビルトインソース管理プロバイダから移行する

LabVIEW 7.*x* 以前のビルトインソース管理プロバイダは、LabVIEW 8.*x* では使用できません。LabVIEW でソース管理を使用する場合は、他社のソース管理プロバイダを選択する必要があります。旧バージョンでビルトインのプロバイダを使用していた場合は、ファイルを他のプロバイダに移行して LabVIEW でソース管理を使用する必要があります。LabVIEW でサポートしている他社のソース管理プロバイダの最新リストについては、ナショナルインスツルメンツのウェブサイト ni.com/info（英語）にアクセスし、info code として exgucn と入力してください。

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

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

1. 旧バージョンの LabVIEW で、LabVIEW のビルトインソース管理プロバイダに含まれているファイルがすべてのユーザによってチェックインされていることを確認します。
2. 新規ソース管理プロバイダにファイルを追加するコンピュータで、ビルトインプロバイダを使用してすべてのファイルの最新バージョンを取得します。
3. ビルトインプロバイダを使用して、ソース管理からファイルをチェックアウトします。
4. 他社のソース管理プロバイダで、新規ソース管理プロジェクトで使用する設定を構成します。
5. LabVIEW を他社のソース管理プロバイダが使用できるように構成します。LabVIEW でサードパーティのソース管理プロバイダを使用する方法の詳細については、『LabVIEW ヘルプ』の**目次**タブで**基本機能→プロジェクトを構成および管理する→操作手順→ LabVIEW でソース管理を使用する**のブックアイコンを参照してください。
6. LabVIEW プロジェクトを作成します。ビルトインプロバイダに含まれるファイルをプロジェクトに追加します。LabVIEW がプロンプトを表示したら、ソース管理にファイルを追加します。また、他社のプロバイダで直接ファイルを追加することもできます。LabVIEW プロジェクトを作成する方法については、『LabVIEW ヘルプ』の**目次**タブで**基本機能→プロジェクトを構成および管理する→操作方法→ LabVIEW プロジェクトを作成する**のブックアイコンを参照してください。

NaN 文字列を整数タイプへ変換する（Windows）

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

LabVIEW 8.*x* では、NaN を明示的または暗示的に整数に変換すると、その値はその整数のデータタイプの最大値となります。たとえば、NaN を符号付き 16 ビット整数に変換すると、16 ビット符号付き整数では最大値である 32,767 となります。

ケーススストラクチャに配線されている定数

LabVIEW 7.*x* 以前では、サブ VI をメモリに常駐させるためには、定数をケーススストラクチャに配線し、実行されないケースにそのサブ VI を配置します。たとえば、ケーススストラクチャに TRUE 定数を配線して、サブ VI をケーススストラクチャの FALSE ケースに配置すると、呼び出す VI と一緒にサブ VI がロードされます。LabVIEW 8.*x* では、実行されないコードはすべて削除されます。したがって、ケーススストラクチャに定数が配線されている VI を旧バージョンの LabVIEW で保存し、LabVIEW 8.*x* でロードすると、旧バージョンの LabVIEW からの動作を維持するために定数が非表示の制御器に変更されます。

オペレーティングシステムのメッセージを遅延する

LabVIEW 7.*x* では、.NET および ActiveX イベントを処理するためのコールバック VI を実行中に、オペレーティングシステムのメッセージが処理されます。LabVIEW 8.*x* では、コールバック VI の実行が停止するまで、またはユーザがモーダルダイアログボックスをロードするまで、オペレーティングシステムのメッセージの処理が遅延されます。この遅延によって、コールバック VI が割り込みなしで実行され、他のイベントでイベントを発行してデットロック状態になる事態を回避できます。

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

LabVIEW 7.*x* では、LabVIEW は DLL または共有ライブラリ関数を実行中に、オペレーティングシステムのメッセージを処理しました。
LabVIEW 8.*x* では、LabVIEW は、DLL 関数への呼び出しが終わるまで、またはユーザがモーダルダイアログボックスを DLL からロードするまで、オペレーティングシステムのメッセージの処理を遅延します。この遅延によって、DLL 関数が割り込みなしで実行し、LabVIEW が DLL 関数が実行

中に同じ DLL を呼び出して、デットロック状態が発生することを避けることができます。

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

作成する DLL でオペレーティングシステムのメッセージを遅延するかどうかを選択できます。DLL 関数の実行中にオペレーティングシステムのメッセージを処理するには、**プロジェクトエクスプローラ**ウィンドウで DLL 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、**カテゴリ**リストから**上級**を選択し、**共有ライブラリでオペレーティングシステムメッセージを遅延**チェックボックスをオフにします。

Resource Manager（Mac OS）

LabVIEW 7.*x* 以前では、Macintosh リソースファイルの読み取り / 書き込み機能（マニュアルへの記載なし）が使用できます。LabVIEW 8.*x* では、これらのメソッドは存在しません。したがって、これらのマニュアルに記載されていなかった機能を利用したユーティリティは使用できません。Macintosh リソースの VI からの読み取りと書き込みはできなくなりました。

1 および 2 ボタンダイアログボックス

LabVIEW 7.*x* 以前では、1 ボタンダイアログまたは 2 ボタンダイアログを表示している VI をプログラム的には中止できません。LabVIEW 8.*x* では、これらのダイアログボックスを表示している VI を VI を中断メソッドを使用してプログラム的に中止できます。

プロパティおよびインボークノード

LabVIEW 7.*x* のカーソル凡例から暗示的にリンクされたプロパティノードまたはインボークノードを作成した場合に、LabVIEW 8.*x* で VI を開くと、LabVIEW はノードを削除します。

共有ライブラリを更新する

LabVIEW 7.*x* 以前で `labview.lib` にリンクされた共有ライブラリ（DLL）を作成した場合は、LabVIEW 8.*x* では代わりに `labviewv.lib` にリンクされます。`labviewv.lib` に共有ライブラリをリンクする方法の詳細については、『LabVIEW ヘルプ』を参照してください。

印刷の余白値

LabVIEW 7.*x* 以前では、**オプション**ダイアログボックスの**印刷**ページの**余白**オプションでは、余白値にセンチメートルが使用されました。LabVIEW 8.*x* の**余白**オプションでは、余白値にミリメートルが使用されます。

LabVIEW 6.x からアップグレードする

LabVIEW 6.*x* から LabVIEW 8.2 にアップグレードする際、以下のような互換性に関する問題が発生する可能性があります。アップグレードの際に起こり得る他の問題点は、このドキュメントの「*LabVIEW 7.x からアップグレードする*」、「*LabVIEW 8.0 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。

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

波形データタイプの変更点

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

LabVIEW 7.*x* 以降では、「ファイルから波形を読み取り（Read Waveform from File）」VI で、ファイル内の古い波形データタイプ形式を新しい形式に変換します。この VI は、変換を実行するかどうかを尋ねるダイアログボックスを表示します。LabVIEW ランタイムエンジンでは、「ファイルから波形を読み取り」VI によるこの変換が行われず、エラーが返されます。LabVIEW 6.*x* から LabVIEW 7.*x* への波形データの移行の詳細情報を参照するには、ナショナルインスツルメンツのウェブサイト `ni.com/info`（英語）にアクセスし、info code として `exd9zq` と入力してください。

シリアル互換性 VI

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

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

成したシリアル VI を含むすべての VI は、LabVIEW 7.1 以降でも動作します。

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

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

ループのデフォルトデータ

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

リモートフロントパネルライセンス

LabVIEW 開発システムとアプリケーションビルダには、1 つのクライアントがリモートでフロントパネルを表示および制御することができるリモートフロントパネルライセンスが含まれています。LabVIEW プロフェッショナル開発システムには、5 つのクライアントがフロントパネルを表示および制御することができるリモートフロントパネルライセンスが含まれています。

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

マルチスレッド割り当て

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

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

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

計測器ドライバ

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

単位および変換ファクタ

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

LabVIEW 7.1 以降の単位変換ファクタは、米国商務省標準技術局（NIST）発行の *Guide for the Use of the International System of Units（SI）*でのガイドラインにさらに準拠しています。また、カロリーの単位はカロリー（熱）となり、馬力は馬力（電気）となりました。これらの単位の略語は以前と同じです。次の表は、LabVIEW 6.1 と 7.*x* 以降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.*x* 以降の定義
天文単位（AU）	149,498,845,000 m	149,597,900,000 m
英温度単位 (平均)	1055.79 J	1055.87 J
電子ボルト（eV）	1.602e-19 J	1.60217642e-19 J
フィートキャンドル	10.764 lx	10.7639 lx
馬力 v.s. 馬力（電気）	745.7 W	746 W 新しい変換は厳密です。
英ガロン	4.54596 l	4.54609 l
光年	9.4605 Pm	9.46073 Pm
重量ポンド	4.448 N	4.448222 N
ロッド	502.92 cm	5.029210 m

単位	6.1 の定義	7.*x* 以降の定義
スラグ	14.594 kg	14.59390 kg
統一された原子質量（u）	1.66057e-27 kg	1.66053873e-27 kg

パネルアップデートを延期プロパティ

LabVIEW 6.1 以前では、「パネルアップデートを延期」プロパティが FALSE になるまで、変更待ちのフロントパネルオブジェクトは再描画されませんでした。LabVIEW 7.0 以降では、このプロパティを TRUE に設定すると、変更待ちになっているすべてのフロントパネルオブジェクトが再描画され、その後のフロントパネルの更新要求がすべて保留にされます。場合によっては、この変更によってフロントパネルの変更された要素が余分な回数再描画されることがあります。

数値制御器のデータ範囲

LabVIEW 6.1 以前では、いくつかの数値制御器のデフォルト設定が最小値 `0.00`、最大値 `0.00`、増分値 `0.00`、範囲外の動作が**無視**に設定されています。LabVIEW 7.*x* では、これらの数値制御器はそのデータタイプのデフォルトのデータ範囲値を使用します。

強制ドットとタイプ定義

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

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

ファイルダイアログボックスのボタンラベル

LabVIEW 6.1 以前では、「ファイルダイアログ（File Dialog）」関数によって表示されるファイルダイアログボックスには、ユーザが新規ファイル名を入力すると、**保存**というボタンラベルが表示されます。選択モードの入力パラメータによっては、ボタンラベルは**開く**になっています。LabVIEW 8.*x* では、「ファイルダイアログ（File Dialog）」Express VI が表示するファイルダイアログボックスのボタンのラベルは、ユーザが変更しない限り、すべて **OK** になりました。ボタンのラベルを変更するには、「ファイルダイアログ」Express VI の**ボタンラベル**入力を使用します。既存の VI で「ファイルダイアログ」Express VI を使用する場合は、VI の動

作を再確認して、デフォルトラベルの **OK** が VI の機能面で適切かどうかを判断する必要があります。

「オンラインヘルプを制御」関数

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

ロード時にフロントパネルを表示する

LabVIEW 7.*x* では、VI がロードされたときにフロントパネルを表示するよう VI が構成されていても、ユーザが VI サーバを使用して VI をロードした場合は、フロントパネルは表示されません。その場合は、フロントパネル : 開くメソッドを使用してプログラム的にフロントパネルを表示する必要があります。

「VI リファレンスを開く」関数

LabVIEW 6.1 以前では、「VI リファレンスを開く（Open VI Reference）」関数の**オプション**パラメータに値を配線しないと、テンプレートがメモリにない場合はテンプレートから VI がインスタンス化されます。テンプレートがメモリにある場合は、テンプレートへのリファレンスが開きます。LabVIEW 7.0 と 7.1 以降では、、「VI リファレンスを開く」関数を使用してメモリに既存のテンプレートへのリファレンスを作成すると、**オプション**パラメータに 0x02 を指定しないと関数によってエラーが返されます。 LabVIEW 8.0 以降では、「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すると、そのテンプレートがすでにメモリにあっても VI はテンプレートからインスタンス化されます。

指数表記

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

IVI Configuration Store ファイル

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

技術サポートフォーム

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

LabVIEW 5.x 以前のバージョンからアップグレードする

LabVIEW 5.*x* 以前から LabVIEW 8.2 へのアップグレードに関する情報は、ナショナルインスツルメンツのウェブサイト `ni.com/info`（英語）で info code として `ext8h9` と入力して参照できます。

LabVIEW 8.2 の機能および変更点

LabVIEW 8.2 の機能の詳細（プログラミングの概念、操作手順、参考情報など）については、『LabVIEW ヘルプ』を参照してください。『LabVIEW ヘルプ』を開くには、**ヘルプ→ LabVIEW ヘルプを検索**を選択します。

LabVIEW 8.2 の既知の問題、その他の互換性に関する問題、最新追加機能については、`labview` ディレクトリの `readme.html` を参照してください。

LabVIEW のドキュメント

LabVIEW 8.2 では、ドキュメントが以下の点で改良されました。

- 概要トピックで、右上隅にナビゲーションのためのリンク一覧が表示されるようになりました。各リンクをクリックすると、関連したサブトピックにジャンプします。

新しいサンプル VI

LabVIEW 8.*x* で新しく追加されたサンプルを内容確認、起動するには、NI サンプルファインダの**参照**タブで **LabVIEW 8.*x* の新しいサンプル**フォルダを展開します。

起動時間の改善

LabVIEW 8.2 は、パフォーマンスが最適化され LabVIEW 8.0 よりも高速に起動します。

ブロックダイアグラムの機能強化

LabVIEW 8.2 では、ブロックダイアグラムまたはそれに関する機能が以下の点で強化または変更されました。

デフォルトの色の変更

以下のブロックダイアグラムコンポーネントのデフォルトの色が変更され、見やすくなりました。

- エラークラスタのワイヤと端子は、ダイアグラム上でピンクではなく黄色で表示されるようになりました。
- 強制ドットのデフォルトの色がグレーから赤に変更されました。強制ドットの色を変更するには、**ツール→オプション**を選択し、**カテゴリ**リストから**色**を選択します。**デフォルトの色を使用**チェックボックスをオフにし、**強制ドット**の色ボックスをクリックします。

VI 階層からブレークポイントを削除する

VI 階層からすべてのブレークポイントを削除するには、VI で**編集→階層からブレークポイントを削除**します。動的に呼び出される VI や「スタティック VI リファレンス（Static VI Reference）」関数によって参照される VI のブレークポイントは、手動で削除する必要があります。

定数のパフォーマンスの最適化

LabVIEW は、VI のパフォーマンスを最適化するために、定数の畳み込みを使用します。定数の畳み込みで VI をコンパイルする場合、実行時に計算する代わりに、定数の値が格納されます。ストラクチャに配線されている定数では、VI のコンパイル時にストラクチャの出力値が計算、格納され、実行時に使用されます。

ブロックダイアグラム上で定数畳み込みのハッシュマークを表示するには、**ツール→オプション**を選択し、**カテゴリ**リストから**ブロックダイアグラム**を選択して、**ワイヤの定数の畳み込みを表示**および**ストラクチャの定数の畳み込みを表示**チェックボックスをオンにします。**ワイヤの定数の畳み込みを表示**チェックボックスがオンの場合、畳み込まれている状態の定数が配線されているワイヤに、ハッシュマークが表示されます。**ストラクチャの定数の畳み込みを表示**チェックボックスがオンの場合、定数に配線されているストラクチャの内部にグレーのハッシュマークが表示されます。このハッシュマークは、VI を実行または保存するまで VI に表示されない可能性があります。

LabVIEW 8.2 では、ケースストラクチャのセレクタ端子と While ループの条件端子に配線する計算された定数も畳み込まれます。

ブロックダイアグラムのその他の機能強化

LabVIEW 8.2 では、ブロックダイアグラムの以下のその他の機能が強化されました。

- プロパティまたはインボークノードの**リファレンス出力**を右クリックしてショートカットメニューから**作成**を選択すると、表示されるメニューにはそのプロパティまたはインボークノードと同じクラスのプロパティまたはメソッドが表示されます。

- 表示項目 : 階層ライン表示プロパティは、3 つの値を受け入れます。**表示**の場合、ツリー制御器内で項目の左側に階層を表す縦方向または横方向のアウトラインが常に表示されます。**非表示**の場合、階層ラインは常に非表示になります。**OS デフォルト**の場合、オペレーティングシステムのツリーで階層ラインが表示されている場合は LabVIEW は階層ラインを表示します。

フロントパネルの機能強化

LabVIEW 8.2 では、フロントパネルまたはそれに関する以下の機能が強化または変更されました。

ペーンの背景画像を設定する

ペーンの背景画像を設定またはインポートすることができます。ペーンのスクロールバーを右クリックして、ショートカットメニューから**プロパティ**を選択します。**ペーンプロパティ**ダイアログボックスで、**背景**リストから画像を選択します。

背景リストに表示されていない画像を選択するには、**参照**ボタンをクリックします。LabVIEW で背景画像としてサポートされているグラフィック形式は、BMP、JPEG、PNG です。ペーンの背景画像は背景画像プロパティを使用してプログラム的に設定することもできます。

メモ **背景**リストに表示されていない画像を選択した場合、その画像は**背景**リストには追加されるわけではありません。リストに画像を追加するには、`labview¥resource¥backgrounds` ディレクトリに画像を保存する必要があります。

背景画像が設定されたペーンを持つ VI を保存すると、その画像は VI とともに保存されます。

ノブとダイアルを最小値または最大値でロックする

ノブとダイアルは、デフォルトでは最大値 / 最小値を超えて回転させることはできません。

このロック動作を無効にするには、ノブ / ダイアル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、**最小値と最大値でロックする**チェックボックスをオフにします。ノブまたはダイアルを固定すると、最小値から最大値、または最大値から最小値へジャンプできなくなります。ロックを無効にすると、意図せずに値が変更されてしまうことがあります。

LabVIEW 8.2 で、最後に LabVIEW 8.0 以前で保存された VI を開くと、ロックが解除されています。ロックを有効にするには、**最小値と最大値でロックする**チェックボックスをオンにしてください。

ツリー制御器またはリストボックス内で複数項目をドラッグする

ツリー制御器やリストボックスから、またはツリー制御器やリストボックスへ複数の項目を選択してドラッグアンドドロップすることができます。複数の項目をドラッグアンドドロップできるように設定するには、それらの制御器を右クリックしてショートカットメニューから**選択モード→0項目以上**または**選択モード→1項目以上**を選択します。複数の項目を選択してドラッグを開始すると、すべての選択項目が移動します。

デジタル波形グラフの機能強化

デジタル波形グラフの詳細については、『LabVIEW ヘルプ』の**目次**タブの**基本機能→グラフおよびチャート**を参照してください。

LabVIEW 8.2 には、デジタル波形グラフが以下の点が機能強化または更されました。

線の太さを設定する

デジタル波形グラフのプロット凡例のショートカットメニューで、**太線の位置**が**ラインスタイル**に変更されました。線の太さを設定するには、プロット凡例でプロットを右クリックして、ショートカットメニューから**ラインスタイル**を選択します。LabVIEW 8.2 では、線全体の太さを設定する新しい**ラインスタイル**オプションが追加されました。

比較データを暗くする

デジタル波形グラフにドライブと比較ロジック状態の両方のデジタルデータがある場合、デフォルトでは比較データがプロット上でドライブデータよりも暗く表示されます。比較データが暗く表示されないようにするには、プロットを右クリックしてショートカットメニューから**上級→比較データを暗くする**を選択して、チェックマークをオフにします。また、比較データを暗くするプロパティを使用して、プログラム的に比較データの色を暗くすることもできます。

メモ この機能は主にデジタル I/O 信号を生成しているユーザに当てはまります。比較データの詳細については、『LabVIEW ヘルプ』の**目次**タブで**基本機能→概要→グラフとチャートをカスタマイズする**ブックを参照してください。

フロントパネルのその他の機能強化

LabVIEW 8.2 では、フロントパネルがその他以下の点で機能強化されました。

- LabVIEW 8.0 では、初めてフロントパネルオブジェクトのキャプションを表示すると、ラベルが横へ自動的に移動していました。LabVIEW 8.2 ではこの場合ラベルが非表示になり、キャプションだけが表示されます。

- **XY グラフのプロパティ**ダイアログボックスで、**任意の平面を表示**プルダウンメニューと選択した平面の設定オプションが、**スケール**ページから**外観**ページへ移動されました。
- 色の設定ツールを使用して、システム表の背景色を変更できます。
- システム複数列リストボックス、リストボックス、表、ツリー制御器のヘッダおよびセルの色を変更できます。
- 表示器のショートカットメニューオプションである**デフォルト値に再度初期化**、**データを切り取り**、**データを貼り付け**は、VI が実行モードになっている間は無効になります。これらのショートカットメニューオプションは、実行モードの制御器のショートカットメニューでは有効です。
- タブ制御器を内容に合わせてサイズ調整するには、制御器を右クリックして、ショートカットメニューから**上級→合うようにサイズ変更**を選択します。
- 列挙体タイプ制御器、リング制御器、コンボボックスを右クリックしてショートカットメニューから**項目を編集**を選択した場合、**項目**または**値**列の項目を編集するにはセルをダブルクリックする必要があります。スライド制御器またはノブ制御器でテキストラベル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**を選択した場合も、同様です。
- LabVIEW 8.0 では、制御器エディタウィンドウで**ファイル→変更を適用**を選択すると、元の制御器のラベル、キャプション、値が保存されていました。LabVIEW 8.2 では、制御器エディタで行ったすべての変更が保存され、元の制御器の情報は保持されません。この処理は、カスタム制御器にのみ適用され、タイプ定義には適用されません。
- グラフまたはチャートの軸を自動スケールする場合、非表示のプロットは対象になりません。非表示プロットも自動スケールの対象にするには、プロットを非表示ではなく透明にしてください。プロットの色を変更するには、プロット凡例を右クリックしてショートカットメニューから**色**を選択します。
- **(Windows)** LabVIEW のラジオボタンとチェックボックスの動作は、Windows と同じです。ラジオボタンとチェックボックスは、スペースキーでのみトグルできます。ダイアログボックスボタンは、英数字キーボード上の <Enter> キー、数値キーパッド上の <Enter> キー、スペースキーのいずれかでトグルできます。LabVIEW のキーボードショートカット <T> と <F> は、ダイアログボックスボタンをトグルしません。
- LabVIEW 8.2 では、タイムスタンプ制御器の上級編集のための新しいトークンが追加されました。制御器を右クリックしてショートカットメニューから**プロパティ**を選択し、**タイムスタンププロパティ**ダイアログボックスを表示します。**形式と精度**ページで**上級編集モード**オプションを選択して、**絶対時間形式コード**リストを表示します。LabVIEW 8.2 では以下の新しい形式コードが使用されています。

形式コード	値
<%^<>T>	グリニッジ標準時コンテナ
<%z>	ロケール時間とグリニッジ標準時の差

環境の機能強化

LabVIEW 8.2 では、LabVIEW 環境に以下の機能強化および変更が導入されました。

回復用の自動保存

正常でないシャットダウンやシステム障害が発生した場合、LabVIEW はシャットダウンまたはシステム障害の発生時に開かれていた、変更が加えられた .vi、.vit、.ctl、または .ctt ファイルのすべてを一時の格納場所にバックアップします。LabVIEW はプロジェクト（.lvproj）、プロジェクトライブラリ（.lvlib）、または XControl（.xctl）、LabVIEW クラス（.lvclass）のバックアップを作成しません。

正常でないシャットダウンやシステム障害が発生する前に LabVIEW が自動的にファイルを保存した場合、次回 LabVIEW を起動するとき、**回復するためにファイルを選択**ダイアログボックスが表示されます。回復するファイルを選択して**回復**ボタンをクリックします。ファイルを回復したくない場合はすべてのファイルを選択解除し、**破棄**ボタンをクリックします。**キャンセル**ボタンをクリックして、デフォルトデータディレクトリの LVAutoSave¥archives サブディレクトリにすべての選択したファイルを移動します。

ツール→オプションを選択し、次に**カテゴリ**リストから**環境**を選択して、回復用の保存を有効または無効に設定し、LabVIEW がファイルをバックアップする頻度を指定します。

ダイアログボックスの機能強化

LabVIEW 8.2 では、ダイアログボックスが以下の点で機能強化されました。

オプションダイアログボックスの機能強化と変更点

LabVIEW 8.2 では、**オプション**ダイアログボックスが以下の点で機能強化されました。削除される予定のオプションに関する情報については、ナショナルインスツルメンツのウェブサイト ni.com/info（英語）にアクセスし、info code として ex6rkc と入力してください。

- **パフォーマンス & ディスク**ページ（**起動時に空ディスク容量をチェック**および**マルチスレッドで実行**の各オプションも含む）は廃止されました。

- **フロントパネル**ページでは、**システムのデフォルトファンクションキー設定を無視**および**描画時にスムーズアップデートを使用**の各オプションが廃止されました。
- **ブロックダイアグラム**ページでは、**配線ガイドを表示**オプションが廃止されました。
- **パス**ページでは、プルダウンメニュー項目の**ライブラリディレクトリ**オプションが廃止されました。
- **環境オプション**ページでは、**短縮メニューを使用**オプションが廃止されました。
- **環境**ページでは、**ジャストインタイムアドバイスを有効**チェックボックスがデフォルトで選択されなくなりました。
- **ブロックダイアグラムオプション**ページでは、**ドロップ時にサブ VI 名を表示**チェックボックスがグローバル変数にも適用されるようになりました。グローバル変数をブロックダイアグラムに配置したときにラベルが表示されるようにするには、このチェックボックスをオンにしてください。
- **ウェブサーバ：表示可能 VI** ページでは、**現在選択された VI** オプションが**表示可能 VI** オプションに変更されました。

LLB マネージャの機能強化

LabVIEW 8.2 では、**LLB マネージャ**ウィンドウが以下の点で機能強化されました。

- **参照**ボタンが**ディレクトリ**フィールドの右側に表示されるようになりました。このボタンをクリックして、修正する LLB ファイルを選択することができます。
- **削除**ボタンがツールバーに表示されるようになりました。**ファイル**リストでファイルを選択してこのボタンをクリックすると、そのファイルが LLB から削除されます。
- **ファイル**リストの列を右クリックして並べ替えることができます。

データバインディングページ

LabVIEW 8.2 では、すべてのフロントパネル制御器の**プロパティ**ダイアログボックスの**データバインディング**ページが以下の点で機能強化されました。

- **現在選択されているネットワーク項目**と**現在選択されているプロジェクト項目**の各テキストボックスが、**パス**テキストボックスに変更されました。
- **モード**セクションが、**アクセスタイプ**プルダウンメニューに変更されました。
- **アラームがオンの間は点滅**チェックボックスは廃止されました。
- **データバインディングセレクション**プルダウンメニューから**シェア変数エンジン (NI-PSP)** を選択すると表示される**参照**ボタンをクリック

すると、**ネットワーク項目を選択**ダイアログボックスではなく、**ソース項目を選択**ダイアログボックスが表示されるようになりました。**ネットワーク項目を選択**ダイアログボックスは廃止されました。

- **データバインディング**ページの代わりに**ソース項目を選択**ダイアログボックスに**ネットワークで共有されるソース**プルダウンメニューが表示されるようになりました。

ダイアログボックスのその他の機能強化

LabVIEW 8.2 では、ダイアログボックスがその他以下の点で変更されました。

- **VI 複雑度測定**ウィンドウの**測定基準の統計**の表に、各列の値の平均値を表示する行が追加されました。また、**統計からこのフォルダにあるファイルを除く**チェックボックスをオンにして、特定のフォルダにあるファイルを除外することもできます。
- **VI プロパティ**ダイアログボックスの**セキュリティ**ページは**保護**ページに変更されました。
- ライセンスされたプロジェクトライブラリのライセンスが評価モードまたは無効な場合、そのライブラリに属する多態性 VI の**多態性 VI** ウィンドウには、**ライセンスの警告を表示**ボタンが表示されるようになりました。このボタンをクリックすると警告メッセージが表示されます。ライセンスの状態の詳細については、警告メッセージ内の**ヘルプ**ボタンをクリックしてください。
- **形式文字列の編集**ダイアログボックスと**スキャン文字列を編集**ダイアログボックスで、変換タイプとして**選択された処理**プルダウンメニューから SI 表記を選択できるようになりました。
- **簡略化された画像をエクスポート**ダイアログボックスの**クリップボードへ保存**オプションが**クリップボードにエクスポート**に変更されました。
- **計測器ドライバ VI を作成**ダイアログボックスの**制御器設定**ページの**出力データタイプ**プルダウンメニューから、**なし**オプションが削除されました。
- **エラーコードファイルエディタ**ウィンドウの**現在のエラーコードと説明**オプションが**エラーコード**オプションおよび**エラーテキスト**オプションに変更されました。

非表示の制御器および表示器を表示する

カスタム制御器とグローバル変数のフロントパネルにあるすべての制御器と表示器を表示するには、**編集→非表示の制御器および表示器を表示**を選択します。このオプションは、カスタム制御器 VI またはグローバル変数の**編集**メニューにのみ表示されます。

カスタム制御器やグローバル変数ではない VI の制御器と表示器は、`labview¥project¥_ShowHidden` ディレクトリの ShowHidden Core.vi を実行して表示することもできます。

エラーリストウィンドウの機能強化

エラーリストウィンドウの**エラーがある項目**リストで、壊れた項目には赤い x グリフが表示されるようになりました。これらの項目は、**エラーがある項目**リストの上方に集めて表示されます。編集中であるために他の項目のエラーの原因となる項目には、鉛筆のアイコンが表示されます。名前の横にアイコンが表示されていない項目は、その項目が従属する項目のエラーが原因でエラーが発生していることを示します。

検索と置換の機能強化

LabVIEW 8.2 では、検索と置換に関する以下の機能が強化または変更されました。

- **テキスト検索オプション**で**クローンを無視**チェックボックスをオンにすると、VI 内でテキスト検索するときにフロントパネルのクローン（再入可能フロントパネル）が検索対象外となります。
- **検索**ダイアログボックスで、**オブジェクトを選択**ボタンが選択されたオブジェクトのファイル名ではなくタイトルとともに表示されます。
- **検索結果ウィンドウ**は、複数のオブジェクトが検索された場合にのみ表示されます。オブジェクトが 1 つだけ検索された場合は、そのオブジェクトがフロントパネルまたはブロックダイアグラムでハイライト表示されます。
- **検索**ダイアログボックスでは、「正規表現で一致（Match Regular Expression）」関数や変数など、Express VI およびノードも検索および置換することもできます。

開いているウィンドウを管理する

ウィンドウメニューには、開いているウィンドウが 10 個まで表示されます。開いているすべてのウィンドウを管理するには、**ウィンドウ→すべてのウィンドウ**を選択するか <Ctrl-Shift-W> を押して**すべてのウィンドウ**ダイアログボックスを開きます。このダイアログボックスでは、右側に表示されるボタンをクリックして特定のウィンドウを手前に表示したり、閉じたり、保存することができます。

最後に保存してから変更が行われたウィンドウは、**タイトル**列の名前の末尾にアスタリスクマークが付けられます。

パレットの機能強化

LabVIEW 8.2 では、パレットが以下の点で機能強化されました。

- **お気に入りを整理**ダイアログボックスを使用して、**お気に入り**パレットカテゴリの項目の順番を変更できます。
- **メニュードキュメント**ダイアログボックスは**パレットドキュメント**ダイアログボックスに変更されました。**メニューの説明**フィールドは**パレットの説明**に変更されました。

- パレットがプロジェクトライブラリに属している場合、そのプロジェクトライブラリへのパスを表示することができます。**ツール→上級→パレットセットを編集**を選択し、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**パスをパレットファイルに表示**を選択します。パレットがライブラリに属している場合、パレットの実際のパスと、所有ライブラリのパスが表示されます。
- **制御器および関数パレットセットを編集**ダイアログボックスに、**保存する前に変更をプレビュー**チェックボックスが追加されました。**保存する前に変更をプレビュー**チェックボックスをオンにし、**変更を保存**ボタンをクリックすると、**パレット変更をプレビュー**ダイアログボックスが開きます。
- **VI をドロップ**ショートカットメニュー項目は、**VI を配置**に変更されました。このショートカットメニューは、**関数**パレットで VI を右クリックすると表示されます。
- LabVIEW 8.2 では、アドオンをインストールするまで空の状態のパレットが表示されます。たとえば、**制御器**パレットの**制御系設計 & シミュレーション**サブパレットは、LabVIEW 制御系設計 & シミュレーション製品をインストールするまで空の状態で表示されます。

階層の複製を保存する

ソース配布を作成することなく VI とそのサブ VI の階層の複製を作成して保存することができます。**ファイル→別名で保存**を選択して **VI 階層の複製を別の場所に保存**オプションを選択し、VI とその階層を新しい場所へ保存します。

環境のその他の機能強化

LabVIEW 8.2 では、フロントパネルがその他以下の点で機能強化されました。

- LabVIEW の学生版または評価版で VI のサブパネルのサブ VI に、ウォーターマークが表示されなくなりました。また、LabVIEW のデバッグデプロイメントバージョンでもウォーターマークが表示されなくなりました。
- 購入した LabVIEW のバージョンの有効なライセンスがない場合は、多態性 VI を編集できないことがあります。
- 指定タイプ定義の制御器へのリファレンスは、その制御器を右クリックし、ショートカットメニュー**作成→リファレンス**を選択して作成できます。
- **表示**メニューに、**プロジェクトエクスプローラ**ウィンドウでプロジェクトライブラリ、XControl、LabVIEW クラスをハイライト表示するための**この VI のライブラリ**、**この VI の XControl**、**この VI のクラス**という項目が追加されました。メニュー項目は、VI が属しているライブラリのタイプに応じて異なります。指定したライブラリ、XControl、クラスが LabVIEW プロジェクト内にない場合は、現在

の VI が属するライブラリ、XControl、クラスだけが表示された新しいウィンドウが開きます。

- LabVIEW 8.0 では、**ヘルプ→エラーの説明**オプションが**スタートアップ**ウインドウに表示されませんでした。LabVIEW 8.2 では、この項目が**スタートアップ**ウインドウに表示されるようになりました。
- **(Mac OS) 詳細ヘルプ**ウィンドウを表示するには、**ヘルプ→詳細ヘルプを表示**を選択するか、<Command-Shift-H> キーを押します。<Command-H> ショートカットキーを押すと、開いている LabVIEW アプリケーションが非表示になります。キーボードショートカットのカスタマイズの詳細については、『LabVIEW ヘルプ』を参照してください。
- **ツール→計測→上級開発**を選択して、下記のような計測ドライバを使用する LabVIEW の詳細な開発オプションを表示することができます。
  - **ドライバガイドラインを表示**を選択すると、ウェブブラウザが開きのナショナルインスツルメンツ計測器ドライバネットワーク（ni.com/idnet）の Instrument Driver Guidelines が表示されます。
  - **Icon Art Glossary を表示**を選択すると、ウェブブラウザが開きナショナルインスツルメンツ計測器ドライバネットワーク（ni.com/idnet）の Icon Art Glossary が表示されます。
  - **その他のリソース**を選択すると、ウェブブラウザが開きナショナルインスツルメンツ計測器ドライバネットワーク（ni.com/idnet）の Development Tools and Resources が表示されます。

新しく追加または変更された VI、関数、ノードの機能強化

LabVIEW 8.2 では、以下の VI および関数が追加または変更されました。VI、関数、ノードの詳細については、『LabVIEW ヘルプ』の**目次**タブの**VI と関数のリファレンス**ブックを参照してください。

新規の VI および関数

LabVIEW 8.2 では、以下の VI および関数が新しく追加されました。

上級ファイル VI

上級ファイル関数パレットには、以下の VI が新しく追加されました。

- 「ファイルまたはフォルダが既存するかチェック（Check if File or Folder Exists）」VI
- 「2 つのパスを比較（Compare Two Paths）」VI
- 「一時ファイルパスを生成（Generate Temporary File Path）」VI
- 「ファイル拡張子を取得（Get File Extension）」VI

- 「MD5 チェックサムファイル（MD5Checksum File）」VI
- 「再帰ファイルリスト（Recursive File List）」VI

デジタル波形関数

デジタル波形パレットには、以下の VI が新しく追加されました。

- 「波形作成（Build Waveform）」関数
- 「デジタルデータを作成（Build Digital Data）」関数
- 「波形要素取得（Get Waveform Components）」関数
- 「デジタルデータコンポーネントを取得（Get Digital Data Components）」関数

数学 VI

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は、**数学**パレットに以下の VI が新しく追加されました。

- 「クロネッカー積（Kronecker Product）」VI
- 「リヤプノブ方程式（Lyapunov Equations）」VI

.NET VI

.NET パレットには、以下の VI が新しく追加されました。

- 「NET オブジェクトに変換（To .NET Object）」VI
- 「.NET オブジェクトからバリアントに変換（.NET Object To Variant）」VI

スケーリング VI

スケーリングパレットには、以下の VI が新しく追加されました。

- 「RTD 読み取り値の変換（Convert RTD Reading）」VI
- 「ひずみゲージ読み取り値の変換（Convert Strain Gauge Reading）」VI
- 「サーミスタ読み取り値の変換（Convert Thermistor Reading）」VI
- 「熱電対読み取り値の変換（Convert Thermocouple Reading）」VI

信号処理 VI

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は、**信号処理**パレットに以下の VI が新しく追加されました。

- 「ボーマンウィンドウ（Bohman Window）」VI
- 「デシメート (連続)（Decimate (continuous)」VI
- 「デシメート (単発)（Decimate (single shot)）」VI
- 「FIR フィルタ（FIR Filter）」VI

- 「FIR フィルタ初期状態付（FIR Filter with I.C.）」VI
- 「ガウス変調正弦パターン（Gaussian Modulated Sine Pattern）」VI
- 「ガウシアンモノパルス（Gaussian Monopulse）」VI
- 「逆チャープ Z 変換（Inverse Chirp Z Transform）」VI
- 「修正バートレット・ハニングウィンドウ（Modified Bartlett-Hanning Window）」VI
- 「パーゼンウィンドウ（Parzen Window）」VI
- 「周期性 Sinc パターン（Periodic Sinc Pattern）」VI
- 「パルス列（Pulse Train）」VI
- 「有理リサンプル（Rational Resample）」VI
- 「サビツキー・ゴーレイフィルタ（Savitzky-Golay Filter）」VI
- 「サビツキー・ゴーレイフィルタ係数（Savitzky-Golay Filter Coefficients）」VI
- 「三角パターン（Triangle Pattern）」VI
- 「アップサンプル（Upsample）」VI
- 「ウェルチウィンドウ（Welch Window）」VI

TDM ストリーム VI および関数

TDM ストリーミングパレットには、以下の VI と関数が新しく追加されました。

- 「TDMS ファイルビューア（TDMS File Viewer）」VI
- 「TDMS を閉じる（TDMS Close）」関数
- 「TDMS を最適化（TDMS Defragment）」関数
- 「TDMS を更新（TDMS Flush）」関数
- 「TDMS プロパティを取得（TDMS Set Properties）」関数
- 「TDMS の内容をリスト（TDMS List Contents）」関数
- 「TDMS を開く（TDMS Open）」関数
- 「TDMS 読み取り（TDMS Read）」関数
- 「TDMS プロパティを設定（TDMS Set Properties）」関数
- 「TDMS 書き込み（TDMS Write）」関数

ストレージパレットには、以下の VI が新しく追加されました。

- 「TDM から TDMS に変換（Convert TDM to TDMS）」VI
- 「TDMS から TDM に変換（Convert TDMS to TDM）」VI

TDM ストリーミングファイル形式の詳細については、このドキュメントの「*TDM ストリーミングファイル形式*」のセクションを参照してください。

変更された VI、関数、ノード

LabVIEW 8.2 では、以下の VI、関数、ノードが変更されました。

DataSocket 関数

DataSocket パレットでは、以下の VI が変更されました。

- 「DataSocket 書き込み（DataSocket Write）」関数を使用する際、NI-PSP データ項目の同期通知を有効にできるようになりました。同期通知を有効にする際、`psp` URL の末尾に `?sync="true"` を追加すると、**ミリ秒タイムアウト**の値をゼロ以外に設定できます。関数は、処理が完了するかタイムアウトになるまで待機します。同期通知が有効になっていると、パフォーマンス（特に RT ターゲット）が非常に遅くなります。
- 「DataSocket 読み取り（DataSocket Read）」関数に、PSP サーバまたは FieldPoint コントローラからの警告やエラーをレポートする**ステータス**出力が追加されました。

数学 VI

LabVIEW ベースパッケージでは、「A x B」VI には新しい A x ベクトルおよび複素 A x ベクトルが含まれています。

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は、**数学**パレットの以下の VI が変更されました。

- 「指数フィット（Exponential Fit）」VI に、観測の重み付けの配列（**X**, **Y**）を指定する**重み付け**入力が新しく追加されました。主に信号ノイズが原因で得られる負の値によって指数フィットが失敗することがなくなりました。
- 「p(x) と q(x) の GCD（GCD of p(x) and q(x)）」VI に、VI で多項式最大公約数の計算に使用するアルゴリズムを指定するための**アルゴリズム**入力が追加されました。
- 「一般多項式フィット（General Polynomial Fit）」VI に、観測の重み付けの配列（**X**, **Y**）を指定する**重み付け**入力が新しく追加されました。
- 「ヒストグラム（Histogram）」、「ヒストグラム PtByPt（Histogram PtByPt）」、「一般ヒストグラム（General Histogram）」、「一般ヒストグラム PtByPt（General Histogram PtByPt）」の各 VI に、入力シーケンス **X** のヒストグラムを表す棒グラフを表示する**ヒストグラムグラフ**出力が追加されました。*y* 軸がヒストグラムのカウントで、*x* 軸がヒストグラムの間隔（ビン）のヒストグラム中央値です。
- 「p(x) と q(x) の LCM（LCM of p(x) and q(x)）」VI に、VI で多項式最小公倍数の計算に使用するアルゴリズムを指定するための**アルゴリズム**入力が追加されました。

- 「部分分数展開（Partial Fraction Expansion）」VI に、VI がどのように**分子**および**分母**の余分子を処理する方法を指定する**オプション**入力が追加されました。
- 「シルベスター行列式（Sylvester Equations）」VI に、**X** の計算速度を高める入力タイプ **A** および **B** を指定するための**行列タイプ**入力が追加されました。
- 「3D 直交座標回転 (方向)（3D Cartesian Coordinate Rotation (Direction)）」VI の配列インスタンスの**方向余弦**入力は、**回転行列**に変更されました。

数値関数

数値パレットでは、以下の関数が変更されました。

- 「商 & 余り（Quotient & Remainder）」関数で、大きい 64 ビットの負の除数を使用した場合により正確な結果が得られるようになりました。
- 「バイトスワップ（Swap Bytes）」関数および「ワードスワップ（Swap Words）」関数の**任意のデータ**入力が、**データ**に変更されました。文字列、タグ、パス、ブール、非整数数値、列挙体、エラークラスタ、ピクチャ、行列、またそれらのタイプのデータの配列およびクラスタを配線し、データを変更することなく渡すことができます。この関数は、入力に配線された 16、32、64 ビットの整数に対して各ワードまたは各ロングワード内の各ワードペアのすべてのバイトペアをスワップします。オカーレンス、Refnum、バリアントは入力に配線できません。
- 86 ベースのプラットフォームでは、「2 の累乗でスケール（Scale By Power Of 2）」関数は **x** 入力と **n** 入力が両方とも浮動小数点数の場合に正しいデータを生成します。

プロトコル VI および関数

プロトコルパレットでは、以下の VI と関数が変更されました。

- 「UDP オープン（UDP Open）」関数と「UDP マルチキャストオープン（UDP Multicast Open）」VI に、使用されたポート番号を返す**ポート**出力が追加されました。
- 「UDP オープン」関数と「UDP マルチキャストオープン」VI に、ネットワークアドレスがリスンする**ネットアドレス**出力が追加されました。
- TCP 接続を開く（TCP Open Connection）」関数の**アドレス**入力は、オプションから必須に変更されました。
- 「TCP リスン（TCP Listen）」VI に、リモートアドレスの「IP アドレスを文字列に変換（IP to String）」関数を呼び出すかどうかを指定する**リモートアドレスを解決**入力が追加されました。デフォルト値は TRUE です。

「2 次計画」VI

「2 次計画（Quadratic Programming)」VI は、以下の点が変更されました。

- 「2 次計画」VI は、2 次計画 IP と 2 次計画 AS というインスタンスを持つ多態性 VI です。2 次計画 IP インスタンスは、LabVIEW 8.0 の「2 次計画」VI と同じ機能性を持ちます。2 次計画 AS インスタンスは、アクティブセットアルゴリズムによって最小値を検出します。
- 「2 次計画」VI のどちらのインスタンスにも、最適化プロセスが開始される *n* 次の点を指定する**開始**入力があります。
- また、どちらのインスタンスにも、最適化プロセスが開始されてから終了するまでに許容される最大時間を指定する**最大時間 (秒)** 入力があります。
- 「2 次計画」VI の 2 次計画 AS インスタンスには、最適化のウォームスタートを許可するかどうかを指定する**ウォームスタート ?** 入力があります。

信号処理 VI

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は、**信号処理**パレットの以下の VI が変更されました。

- 「中央値フィルタ (Median Filter)」VI では、LabVIEW 8.0 の**ランク（階級）**入力が**左ランク**および**右ランク**に変更されました。VI は入力シーケンス **X** にランクの中央値フィルタを適用します。ここで、**右ランク**が 0 より大きい場合はランクは**右ランク**、または**右ランク**が 0 より小さい場合はランクは**左ランク**になります。

メモ **ランク**入力に値が配線された「中央値フィルタ (Median Filter)」VI を含む VI を LabVIEW 8.0 で作成して保存し、それを LabVIEW 8.2 で開いた場合、**左ランク**の値が使用されます。

- 「正弦波（Sine Wave)」、「正弦波 PtByPt（Sine Wave PtByPt)」、「三角波（Triangle Wave)」、「三角波 PtByPt（Triangle Wave PtByPt)」、「ノコギリ波（Sawtooth Wave)」、「ノコギリ波 PtByPt（Sawtooth Wave PtByPt)」、「方形波（Square Wave)」、「方形波 PtByPt（Square Wave PtByPt)」、「任意波形（Arbitrary Wave)」VI の **f** 入力は、**周波数**に変更されました。
- 「サビツキー・ゴーレイフィルタ（Savitzky Golay Filter PtByPt)」VI の**サイドポイント数**入力が、**サイドポイント**に変更されました。
- 「ポイントバイポイント（Point By Point)」VI のアイコンが変更されました。
- 「対称ウィンドウ（Symmetric Window)」VI の**ウィンドウ**入力と「スケール時間領域ウィンドウ（Scaled Time Domain Window)」VI に、修正バートレット・ハニング、ボーマン、パーゼン、ウェルチというオプションが追加されました。

サウンド VI（Linux）

サウンド VI を使用するには、Open Sound System (OSS) ドライバが必要です。OSS 著作権情報を参照するには、『LabVIEW ヘルプ』の**目次**タブで**重要情報→著作権**ブックを展開してください。

メモ LabVIEW は、`¥dev¥dsp` または `¥dev¥dspX`（このとき *X* は `0` ～ `16` の整数）という名前のファイルを検索してデバイスをプローブします。LabVIEW は、各入出力デバイスを開こうとします。サウンドカードが検出されない場合は、デバイスのファイル名 `¥dev¥dsp` または `¥dev¥dspX` がローカルシステムにあり、またそのデバイスの読み取り / 書き込み権限を持っていることを確認してください。このデバイスをデフォルト以外の場所に移動すると、LabVIEW はシンボリックリンクを使用します。

Linux でのサウンド VI は、以下の点が変更されました。

- サウンド VI はモノラルまたはステレオサウンドをサポートします。
- 波形はサウンドデータを表します。符号なし 8 ビット、符号付き 16 ビット、または符号付き 32 ビットの整数、あるいは単精度および倍精度のデータタイプの要素を使用すると、**Y** 配列データを表すことができます。各波形は 1 つのチャンネルを定義します。
- サウンドデータの形式は、PCM（Pulse Code Modulated）です。
- VI は、連続するサウンド出力を生成できます。
- VI は、波形ファイルのストリーミング表示を可能にします。
- VI は、エラーチェックの機能が向上しています。

文字列関数

文字列パレットでは、以下の関数が変更されました。

- 「文字列からスキャン（Scan From String）」関数の**形式文字列**入力は、国際単位（SI）表記を受け入れます。
- 「スプレッドシート文字列を配列に変換（Spreadsheet String to Array）」関数は、**配列タイプ**入力が複素数の場合でも正しく動作します。

VISA 関数

特定の VISA 関数の端子を右クリックし、ショートカットメニューから**作成→定数**または**作成→制御器**または**作成→表示器**を選択すると、リングオブジェクトが表示されます。オブジェクトは、選択肢を繰り返し表示できるプルダウンメニューとして表示されます。LabVIEW でサポートされるすべての値がプルダウンメニューに表示されます。

この変更は、以下の関数と端子に影響します。

関数	端子
VISA 割り込み信号アサート	モード
VISA トリガアサート	プロトコル
VISA ユーティリティ信号アサート	バス信号
VISA イベント無効	イベントタイプ
VISA イベント破棄	イベントタイプ
VISA イベント有効	イベントタイプ
VISA リソース検索	検索モード
VISA GPIB 制御 ATN	モード
VISA GPIB 制御 REN	モード
VISA トリガマップ	トリガソース、トリガ送信先
VISA オープン	アクセスモード
VISA トリガマップ解除	トリガソース、トリガ送信先
VISA VXI コマンドまたはクエリ	モード
VISA イベント待機	イベントタイプ入力、イベントタイプ出力

ライブラリ関数呼び出しノード

ライブラリ関数呼び出しノードにエラー端子が追加されました。**ライブラリ関数呼び出し**ダイアログボックスもまた以下の点が変更されました。

- 複数ページによる構成となり、パラメータの設定がより簡単になりました。
- **関数**ページでは、**ダイアグラムでパスを指定**チェックボックスをオンにして、ライブラリパスをプログラム的に指定できるようノードを構成することができます。
- **パラメータ**ページでは、文字列や配列パラメータを構成する場合に**最小サイズ**テキストボックスを使用してメモリを正確に割り当てることができます。
- **コールバック**ページで、ユーザ定義のコールバックを指定できます。

VI、関数、ノードのその他の変更点

LabVIEW 8.2 では、VI、関数、ノードの以下の点が変更されました。

- 「時間領域演算（Time Domain Math）」Express VI の構成ダイアログボックスの**演算**オプションは、**微分**が**微分（dX/dt）**に、**差**が**差（dX）**に、**積分**が**積分（Sum(Xdt)）**に、**総和**が**総和（Sum(X)）**に変更されました。

- **(Windows)** 「サウンドファイル情報（Sound File Info)」VI のサウンドファイル情報 (Refnum) インスタンスの **Refnum 入力**の入力は、**サウンドファイル Refnum** に変更されました。
- **(Mac OS、Linux)** 「レポートフッタテキスト設定（Set Report Footer Text)」VI は、フッタを LabVIEW の以前のバージョンと若干異なるサイズで生成します。また「レポートフッタテキスト設定」VI には新しく **HTML フッタサイズ**入力が追加されました。
- 「タイミングストラクチャの開始を同期化（Synchronize Timed Structure Starts)」VI に、グループに指定したタイミングストラクチャを追加する前にすべてのタイミングストラクチャとグループ全体を削除する**クリア**入力が追加されました。この入力は、タイミングループに一致しないタイミングストラクチャをすべて削除するために使用します。またこの VI には、同期グループにタイミングストラクチャの名前が追加された後にグループ内のすべてのタイミングストラクチャ名を返す**タイミングストラクチャ名出力**の出力が追加されました。
- **(Windows)** LabVIEW 8.0 では、「リストフォルダ（List Folder)」関数の**パターン**入力に拡張子がない場合、.* が追加されます。LabVIEW 8.2 では、この関数は**パターン**入力を変更しませんが、ファイル名に拡張子がなく**パターン**に拡張子がある場合は、ファイル名に . を追加します。
- ブロックダイアグラムに「スタティック VI リファレンス（Static VI Reference)」関数を配置してダブルクリックすると、ファイルダイアログボックスが表示され、VI を選択することができます。LabVIEW 8.0 では、関数をダブルクリックするとファイルが見つからないというエラーダイアログボックスが表示されます。
- 「ファイルにフォーマット（Format Into File)」、「文字列にフォーマット（Format Into String)」、「ファイルからスキャン（Scan From File)」、または「文字列からスキャン（Scan From String)」関数の**形式文字列**入力は、必須入力ではなくなりました。
- シェア変数ノードを含む VI をプロジェクト内で開いた際、関連付けられたシェア変数を**プロジェクトエクスプローラ**ウィンドウ内で見つけられないと、そのシェア変数ノードは破損します。見つからないシェア変数に関連付けられているフロントパネル制御器もすべて破損します。この現象は、Windows でのみ発生し、また VI ノードをプロジェクト内で開いた場合にのみ発生します。VI をメインアプリケーションインスタンスで開いた場合は、シェア変数が見つからないというメッセージが表示されません。旧バージョンの LabVIEW では、シェア変数ノードがプロジェクトエクスプローラウィンドウ内でシェア変数を見つけられなくてもメッセージが表示されません。
- LabVIEW 8.0.1 では、タイミングループの**起動理由**出力はメニューリングです。LabVIEW 8.2 では、この出力は LabVIEW 8.0 と同じ列挙体タイプです。

- 以下の VI と関数の複製出力の名前が**複製 (出力)** から **(出力) 出力**に変更されました。この変更によって、これらの VI や関数の機能が影響を受けることはありません。
  - リファレンス呼び出しノード
  - クエリ表記を実行
  - ファイル / ディレクトリ情報
  - オブジェクト情報取得
  - プロパティを取得
  - インポークノード
  - リストフォルダ
  - プロパティノード
  - 文字列からトークン読み取り
  - Refnum から ID に変換
  - プロパティを設定

新しいプロパティ、メソッド、イベント

LabVIEW 8.2 では、新しい VI サーバクラス、プロパティ、メソッド、イベントが追加されました。新しいクラス、プロパティ、メソッド、イベントの一覧は、『LabVIEW ヘルプ』の**目次**タブで **LabVIEW 8.2 の機能および変更点→新規の VI サーバのクラス、プロパティ、メソッド、イベント**ブックを展開して参照できます。

LabVIEW 8.2 では、さらに PXI/PCI Settings:Is PCI Express、PXI/PCI Settings:Maximum Link Width、PXI/PCI Settings:Actual Link Width、PXI/PCI Settings:Slot Link Width、PXI/PCI Settings:D-Star Bus Number、PXI/PCI Settings:D-Star Set といった VISA プロパティが追加されました。

LabVIEW MathScript の機能強化

LabVIEW MathScript の詳細については、『LabVIEW ヘルプ』の**目次**タブの**基本機能→フォーミュラと方程式**を参照してください。

LabVIEW 8.2 では、MathScript の以下の機能強化および変更が導入されました。

新しい MathScript 関数

LabVIEW 8.2 では、以下の MathScript 関数が新しく追加されました。これらの関数は、**LabVIEW MathScript ウィンドウ**または MathScript ノードで使用できます。

- plots クラス : `area`, `bar`, `bar3`, `bar3h`, `barh`, `contour`, `contour3`, `contourf`, `errorbar`, `ezcontour`, `ezcontourf`, `ezmesh`, `ezmeshc`, `ezplot`, `ezplot3`, `ezpolar`, `ezsurf`, `ezsurfc`, `feather`, `fill`,

fplot, gplot, meshc, pie, plotmatrix, plotyy, polar, quiver, scatter, scatter3, shg, stem3, strips, surfc, treeplot, waterfall.

- dsp クラス : ac2poly, ac2rc, arburg, arcov, armcov, aryule, barthannwin, bartlett, besselap, besself, bilinear, bitrevorder, blackman, blackmanharris, bohmanwin, buttap, cceps, cheb1ap, cheb2ap, chebwin, chirp, conv2, convmtx, corrmtx, czt, dct, dftmtx, digitrevorder, diric, downsample, dst, ellipap, eqtflength, filternorm, filtic, firgauss, firrcos, flattopwin, freqspace, gauspuls, gausswin, gmonopuls, goertzel, hann, icceps, iczt, idct, idst, impinvar, intfilt, invfreqs, invfreqz, is2rc, kaiserord, lar2rc, latc2tf, levinson, lp2bp, lp2bs, lp2hp, lp2lp, lpc, lsf2poly, maxflat, medfilt1, nuttallwin, parzenwin, phasedelay, phasez, poly2ac, poly2lsf, poly2rc, polyscale, polystab, prony, pulstran, rc2ac, rc2is, rc2lar, rc2poly, rceps, rectpuls, rectwin, resample, residuez, rlevinson, schurrc, seqperiod, sgolay, sgolayfilt, sinc, sos2ss, sos2tf, sos2zp, sosfilt, spline, ss2sos, ss2tf, ss2zp, stepz, stmcb, tf2latc, tf2sos, tf2ss, tf2zp, tf2zpk, triang, tripuls, tukeywin, udecode, uencode, upfirdn, upsample, vco, xcorr, xcorr2, xcov, zerophase, zp2sos, zp2ss, zp2tf.
- support クラス : csvread, csvwrite, dlmread, dlmwrite, labviewroot, type, uiload, what.
- string クラス : eval.
- libraries クラス : loadlibrary, calllib, unloadlibrary, libisloaded, libfunctionsview. これらの関数を使用して、**LabVIEW MathScript ウインドウ**または MathScript ノードから共有ライブラリを呼び出すことができます。MathScript から共有ライブラリを呼び出す例としては、labview¥examples¥MathScript¥MathScript Shared Libraries ディレクトリにある MathScript Shared Libraries.lvproj を参照してください。

MathScript のその他の機能強化と変更点

LabVIEW 8.2 では、MathScript がその他以下の点で変更されました。

- **LabVIEW MathScript ウィンドウ**のパフォーマンスが全体的に向上しました。MathScript ノードでのコンパイル時間が短縮されました。
- LabVIEW ランタイムエンジンは MathScript をサポートします。アプリケーションビルドで MathScript ノードを使用したスタンドアロンアプリケーションや共有ライブラリを作成できます。ただし、現時点では一部の MathScript 関数が LabVIEW ランタイムエンジンでサポートされていません。スクリプトにこれらのサポートされていない関数が含まれている場合は、アプリケーションや共有ライブラリを作成する前にスクリプトを変更する必要が生じる可能性があります。

LabVIEW ランタイムエンジンでサポートされていない MathScript 関数の一覧については、『LabVIEW ヘルプ』の「LabVIEW ランタイムエンジンでサポートされていないプロパティとメソッド」のトピックを参照してください。

- MathScript で `help` コマンドを呼び出すと、HTML ヘルプウィンドウが開き、ヘルプが表示されます。**LabVIEW MathScript ウィンドウ**の**出力ウィンドウ**にヘルプを表示するには、**ファイル→ MathScript 環境設定**を選択して **HTML ヘルプを表示？**チェックボックスをオフにします。関数を定義すると、その定義した関数のヘルプが常に**出力ウィンドウ**に表示されます。
- MathScript ノードは、入力端子にサポートされていないデータタイプを配線すると他のスクリプトノードとは異なる動作をします。このような場合、MathScript ノードではデータタイプがサポートされているタイプに変換されるか、壊れたワイヤが表示されます。LabVIEW がデータタイプを変換した場合、端子上でデータが変換された位置に強制ドットが表示されます。MathScript ノードに入力を配線した後、入力のデータタイプを参照するには、入力端子を右クリックし、ショートカットメニューから**データタイプを表示**を選択します。
- MathScript では、Unicode 以外の文字をテキスト文字列ではサポートしますが、変数名ではサポートしていません。変数名では ASCII 文字だけを使用できます。たとえば、テキスト文字列で `á` を使用することはできますが、以下のスクリプトを **LabVIEW MathScript ウィンドウ**または MathScript ノードから呼び出すことはできません。

  ```
  á=rand(50, 1)
  plot(á)
  ```

  非 Unicode 文字が使用されたパスにデータを保存できます。また、LabVIEW を Unicode 以外の文字が使用されたディレクトリパスにインストールすることができます。この場合にも、MathScript は正常に動作します。
- LabVIEW では、MathScript で複合論理の数式評価に短絡評価（short circuit evaluation）を使用します。たとえば、`if 0 == 0 || foo(a) == 2` というコマンドを実行する場合、数式の最初の部分が TRUE であるため `foo(a)` を常に実行しません。同様に、`if 0 ~= 0 && foo(a) == 2` というコマンドを実行する場合、数式の最初の部分が FALSE であるため `foo(a)` を実行しません。

  LabVIEW 8.0 では、MathScript で複合論理式が TRUE であっても FALSE であっても全体が評価されるため、複合論理式を含む LabVIEW 8.0 のスクリプトが LabVIEW 8.2 では正常に動作しない場合があります。たとえば、LabVIEW 8.0 では、コマンド `if 0 == 0 || foo(a) == 2` の `foo(a)` が実行されるため、スクリプト内で `foo(a)` の結果を変数の定義に使用することができます。このスクリプトは、LabVIEW 8.2 では異なる動作をします。たとえば、LabVIEW 8.2 では、コマンド `if 0 == 0 || foo(a) == 2` の

foo(a) が実行されないため、スクリプト内で foo(a) の結果を使用できません。LabVIEW で短絡評価が実行されないようにするには、既存のコードから複合論理式を削除してください。

- MathScript では、ユーザ定義関数内で nargin 関数と nargout 関数を使用できます。
- MathScript は return キーワードをサポートします。また、end キーワードを行列指標で使用することもできます。
- prod 関数と sum 関数に、積または和を計算する次元を指定するための **b** 入力が追加されました。

3D ピクチャ制御器

3D ピクチャ制御器の詳細については、『LabVIEW ヘルプ』の**目次**タブの**基本機能→グラフおよびサウンド VI** を参照してください。

3D ピクチャ制御器の各 VI、プロパティ、およびメソッドは 3D オブジェクトの集合を表示および操作可能な 3D シーンに変換します。複数の 3D オブジェクトを生成し、オブジェクトのサイズ、形状、動き、外観、およびシーンにおける他のオブジェクトとの関係を指定できます。

以下の VI、プロパティ、およびメソッドを使用して、3D オブジェクトの外観を設定します。

- **オブジェクト** VI を使用して、3D オブジェクトを作成または検索します。また、シーンオブジェクトプロパティおよびシーンオブジェクトメソッドを使用して、シーンに 3D オブジェクトを配置し、オブジェクトの特性をプログラム的に割り当てることもできます。
- **ファイルロード** VI を使用して、既存のモデルやシーンファイルを 3D シーンに追加します。
- **ジオメトリ** VI をシーンジオメトリプロパティおよびシーンジオメトリメソッドと使用して、3D オブジェクトの幾何学的形状を指定します。
- **変換** VI を使用して、3D シーンのオブジェクトを配置します。

ヘルパー VI を使用して、一般的な 3D シーン操作を行います。シーンに対して別のウインドウを構成、シーンに新規のクリップ面を作成、光源を追加、3D オブジェクトにテクスチャを適用、または LabVIEW のカラー値を 3D ピクチャ制御器に表示されるように変換することができます。

また、以下のプロパティおよびメソッドを使用して、3D シーンをプログラム的に構成することもできます。

- シーンウインドウプロパティを使用して、別のウインドウにシーンを作成、ウインドウを構成、およびシーンとカメラコントローラの相互的動作を設定します。
- シーンクリッププレーンプロパティを使用して、オブジェクトが表示されたり、消えるシーンの平面を指定します。

- シーンの光プロパティを使用して、シーンに対する光源を構成します。
- シーンテクスチャプロパティおよびシーンテクスチャメソッドを使用して、3D オブジェクトにテクスチャを適用します。

3D ピクチャ制御器で 3D シーンを作成する例については、`labview¥examples¥picture¥3D Picture Control` ディレクトリの solarsystem VI を参照してください。

LabVIEW オブジェクト指向プログラミング

LabVIEW でのオブジェクト指向プログラミングの詳細については、『LabVIEW ヘルプ』の**目次**タブで**基本機能→ LabVIEW オブジェクト指向プログラミング**ブックを参照してください。

LabVIEW オブジェクト指向プログラミングでは、クラス構造、カプセル化、継承といった、C++ や Java など一般的なオブジェクト指向プログラミングと同様のコンセプトが採用されています。これらのコンセプトを使用しながら、アプリケーション内の他の部分のコードに影響することなく簡単に管理、修正できるコードを作成することができます。また、LabVIEW 内でオブジェクト指向プログラミングを使用してユーザ定義データタイプを作成することができます。

LabVIEW クラスを作成する

LabVIEW でユーザ定義データタイプを作成するには、LabVIEW クラスを作成します。LabVIEW クラスは、オブジェクトに関連付けるデータ、またデータに対して実行するアクションを定義するメソッドを定義します。

LabVIEW では、クラスのデータはプライベートです。つまり、クラスのメンバーである VI のみがそれらのデータにアクセスできます。クラスのデータは、プライベートデータ制御器で定義します。LabVIEW クラスを作成して保存すると、新しいデータタイプを定義するためのクラスライブラリファイル（`.lvclass`）がダイナミックに作成されます。クラスライブラリファイルには、プライベートデータ制御器と、作成されたメンバー VI のリストやプロパティなどの情報が記録されます。クラスライブラリは、プロジェクトライブラリ（`.lvlib`）に類似しています。ただし、新しいデータタイプを定義する点で異なります。

クラスは、以下のいずれかの手順で作成できます。

- **プロジェクトエクスプローラ**ウィンドウの**マイ コンピュータ**を右クリックして、ショートカットメニューから**新規→クラス**を選択します。
- **ファイル→新規**を選択して**新規**ダイアログボックスを開き、**新規作成**リストで**他のファイル→クラス**を選択します。

プライベートデータ制御器を定義する

LabVIEW クラスを作成すると、クラスのプライベートデータ制御器がダイナミックに作成されます。

クラスのプライベートデータ制御器をカスタマイズするには、制御器エディタウィンドウを使用します。**プロジェクトエクスプローラ**ウィンドウでクラスのプライベートデータ制御器をダブルクリックすると、制御器エディタウィンドウが表示されます。LabVIEW クラスのプライベートデータタイプを定義するには、**クラスプライベートデータのクラスタ**内に制御器と表示器を配置します。**クラスプライベートデータのクラスタ**の制御器のデフォルト値には、クラスのデフォルト値が使用されます。

メンバー VI を作成する

メンバー VI は、LabVIEW クラス用に作成されたメソッドを実装します。メンバー VI を作成して、クラスのプライベートデータに対する操作を実行できます。メンバー VI は、作成先の LabVIEW クラスのメンバーとして、**プロジェクトエクスプローラ**ウィンドウでそのクラスのプライベートデータ制御器の下に表示されます。ほとんどのメソッドは、1 つのクラスで単独のメンバー VI を使用して定義できますが、クラス階層全体に渡って複数のメンバー VI を作成して定義するメソッドもあります。

エラー処理とクラスオブジェクトを含む VI テンプレート、ブランク VI、または先祖メンバー VI からメンバー VI を作成することができます。クラスを右クリックしてショートカットメニューから以下のいずれかの項目を選択します。

- **新規→ VI**—ブランクメンバー VI を開きます。
- **新規→ダイナミック VI**—**エラー入力**クラスタ、**エラー出力**クラスタ、エラー処理用のケーストラクチャ、入力 LabVIEW クラス、出力 LabVIEW クラスを持つ新しいメンバー VI が作成されます。
- **新規→ VI を上書き**—先祖メンバー VI を上書きするメンバー VI を作成します。

LabVIEW クラスを他の開発者やユーザに配布する

作成した LabVIEW クラスを別の LabVIEW クラスの開発者やユーザに配布することができます。クラスはいくつかの方法で配布できますが、状況に最も適した方法を選択する必要があります。なお、アプリケーションビルダを使用して、配布するクラスの Zip ファイルを作成することができます。また、配布する前に LabVIEW クラスをロックしてクラスユーザによるプライベートデータやメンバー VI へのアクセスを制限することもできます。クラスをロックするとアプリケーションでのエラー発生を防止することができます。

LabVIEW プロジェクトの機能強化

LabVIEW 8.2 では、LabVIEW プロジェクトまたはそれに関する以下の機能が強化または変更されました。

アプリケーションビルダの機能強化

LabVIEW 8.2 のプロフェッショナル開発システムでは、アプリケーションビルダの以下の機能強化が追加されました。

- **自動増分製品番号**—インストーラを複数回作成すると、LabVIEW によって各インストーラに自動的に新しい製品バージョン番号が付けられます。**インストーラプロパティ**ダイアログボックスの**製品情報**ページには**自動増分製品番号**チェックボックスが表示されます。
- **インストーラの保存メディアサイズを指定**—インストーラコンポーネントを保存するとき、メディアサイズをカスタマイズできます。**インストーラプロパティ**ダイアログボックスの**上級**ページには**メディアスパニングを有効**チェックボックスが表示されます。メディアのプルダウンメニューに、**カスタム**オプションが追加されました。このオプションを選択すると、**メディアサイズ (MB)** テキストボックスで任意のメディアサイズを指定できます。
- **Windows 2000 以降を要件として指定**—アプリケーションの要件として Microsoft Windows 2000 Service Pack 3 以降のバージョンを指定することができます。**インストーラプロパティ**ダイアログボックスの**上級**ページの**システム要件**セクションに **Windows 2000 以降**オプションが追加されました。
- **インストーラコンポーネントのキャッシュ**—インストーラを複数回作成し、そのインストーラに追加のインストーラまたはコンポーネントが含まれる場合は、キャッシングによってそれらの追加コンポーネントの場所を毎回指定する手間を省略することができます。インストーラを初めて作成すると、**配布を検索**ダイアログボックスが開き、追加コンポーネントを含む配布の場所を指定できます。**この配布からのキャッシュコンポーネント**チェックボックスをオンにすると、その配布からローカルシステムの正式な場所にファイルがコピーされます。これらのコンポーネントを含むインストーラを次回作成するときは、配布 CD のセットを要求するメッセージが表示されず、アプリケーションビルダによって自動的にコンポーネントがローカルシステムからファイルがコピーされます。
- **ダイナミック VI およびサポートファイル**—**ソースファイル設定（アプリケーション）**ページでダイナミック VI として指定されている VI が、ビルドされたアプリケーション以外で使用される場合は、そのダイナミック VI のすべての従属ファイルは、従属関係を維持するのではなくアプリケーションビルダによって新しい場所へ移動されます。複数のダイナミック VI またはトップレベル VI が 1 つの VI を呼び出して複数の場所へ移動しようとすると、アプリケーションビルダはその VI とすべてのサブ VI をビルドされたアプリケーションへ移動します。**ソースファイル設定**ページで**参照されている場合のみ含む**として

指定された VI を新しい場所へ移動するには、その VI をダイナミック VI として指定する必要があります。

- **共有ライブラリでプロジェクトエイリアスファイルを使用する**—**共有ライブラリプロパティ**ダイアログボックスの**上級**ページに、共有ライブラリとプロジェクトエイリアスファイルを関連付けるための**デフォルトのプロジェクトエイリアスファイルを使用**チェックボックスが追加されました。チェックボックスをオフにした場合は、**プロジェクト中のエイリアスファイル**テキストボックスで使用するエイリアスファイルを指定します。
- **共有ライブラリに LabVIEW ヘッダフィルタを追加**—**共有ライブラリプロパティ**ダイアログボックスの**上級**ページに、ビルドプロセス中に生成されたヘッダファイルが参照する追加 LabVIEW ヘッダファイルをすべてコピーするための**追加の LabVIEW ヘッダフィルタを含む**チェックボックスが追加されました。これらの追加ヘッダファイルをコピーすることで、それらを必要とする C 言語やその他の言語で LabVIEW ビルド共有ライブラリを使用できるようになります。
- **ソース配布**ダイアログボックスの**配布設定**ページにある**新規出力先**ボタンが、**追加**ボタンに変更されました。また、ソース配布からのファイルの除外に関するすべてのオプションが、**追加除外**ページに移動されました。
- **アプリケーションプロパティ**ダイアログボックスと**共有ライブラリのプロパティ**ダイアログボックスの**出力先**ページにある**新規出力先**ボタンが、**追加**ボタンに変更されました。
- アプリケーションまたはソース配布は、VI を保存しないで作成できるようになりました。
- アプリケーションビルダで作成する zip ファイルには、手動で `.zip` 拡張子を追加しなくても、自動的に追加されます。

新規ダイアログボックスのページ

LabVIEW 8.2 では、**アプリケーションビルダ**ダイアログボックスに以下のページが追加されます。

- **インストーラプロパティ**ダイアログボックス
  - **ダイアログ情報**—このページでは、インストーラのユーザインタフェースを設計できます。また、テキストの言語設定、カスタムの Readme やライセンス契約の表示、初期画面のタイトルやメッセージの設定も行えます。このページは、LabVIEW 8.0 の**製品情報**ページにある**ダイアログ情報**セクションに代わるものです。

メモ **製品情報**ページの**ライセンスファイル**オプションは、**ダイアログ情報**ページの**カスタム使用許諾契約書を使用**オプションに変更されました。

  - **ハードウェア構成**—このページでは、インストーラに使用するハードウェア構成情報のソースを指定できます。新しい**インポー**

トモードセクションには、Measurement & Automation Explorer からハードウェア構成ファイルをインポートする際の詳細なオプションが追加されました。このページは、LabVIEW 8.0 の**上級**ページにある**ハードウェア構成**セクションに代わるものです。

- **アプリケーションプロパティ**、**共有ライブラリプロパティ**、**ソース配布プロパティ**ダイアログボックス
  - **追加除外**—このページでは、タイプ定義、未使用の多態性 VI インスタンス、およびプロジェクトライブラリの参照されていないメンバーの削除または接続解除の設定を構成できます。ビルドのサイズを小さくするには、これらの設定を使用します。

ビルド仕様の複製と並べ替え

プロジェクトエクスプローラウィンドウでビルド仕様の複製を作成できます。**ビルド仕様**の下のビルド仕様のコピーを作成するには、そのビルド仕様を右クリックし、ショートカットメニューから**複製**を選択します。

また、**ビルド仕様**内でビルド仕様項目をドラッグアンドドロップして順序を変更することもできます。

アプリケーションビルダをアクティブ化する（Windows）

アクティブ化された LabVIEW 開発システムまたはベースパッケージのバージョンでは、**ヘルプ→アプリケーションビルダをアクティブ化**を選択してアプリケーションビルダをアクティブ化することができます。ライセンスは LabVIEW を再起動すると有効になります。

プログラム的にプロジェクトライブラリを作成してシェア変数を追加する

プロジェクトまたはターゲット、フォルダ、ライブラリなどの親項目にプログラム的にライブラリを追加するには、`labview¥vi.lib¥Utility¥Variable` ディレクトリの CreateOrAddLibrary.vi を使用します。

または、`labview¥vi.lib¥Utility¥Variable` ディレクトリの AddSharedVariableToLibrary.vi を使用して、シェア変数をライブラリにプログラム的に追加できます。

LabVIEW プロジェクトとプロジェクトライブラリを旧バージョン形式で保存する

プロジェクトライブラリ、LabVIEW プロジェクトを、LabVIEW 8.0 で読み取り可能な形式で保存できます。旧バージョンの LabVIEW プロジェクトを保存するには、**プロジェクトエクスプローラ**ウィンドウで**ファイル→旧バージョン用に保存**を選択します。旧バージョンのプロジェクトライブラリを保存するには、**プロジェクトエクスプローラ**ウィンドウでライブラリファイルを右クリックしてショートカットメニューから**旧バージョン用**

に保存を選択するか、プロジェクトライブラリを開いて**ファイル→旧バージョン用に保存**を選択します。

アプリケーションインスタンスを選択する

LabVIEW は、LabVIEW プロジェクトで各ターゲットのアプリケーションインスタンスを作成します。**プロジェクトエクスプローラ**ウィンドウから VI を開くと、VI はターゲットのアプリケーションインスタンスで開かれます。また LabVIEW は、プロジェクトに含まれない開いている VI、およびプロジェクトから開かれていない VI を含む、メインアプリケーションインスタンスを作成します。VI は、アプリケーションインスタンスのショートカットメニューを使用して、そのアプリケーションインスタンスで開くことができます。VI を開くアプリケーションインスタンスを選択するには、フロントパネルまたはブロックダイアグラムの左下端にあるプルダウンメニューを右クリックしてショートカットメニューを表示します。新しいアプリケーションインスタンスを選択すると、VI が選択されたアプリケーションインスタンス内で再度開かれます。元のアプリケーションインスタンスの VI も開かれたままになります。

シェア変数の機能強化

LabVIEW 8.2 では、シェア変数が以下の点で機能強化されました。

- LabVIEW では、ネットワークで共有されるシェア変数のネットワークバッファサイズの計算にネットワーク : バッファサイズプロパティ、ネットワーク : 要素サイズプロパティ、ネットワーク : 各波形のポイントプロパティが適宜使用されます。
  - ネットワークで共有されるスカラのシェア変数では、バッファする値の数を示すネットワーク : バッファサイズプロパティが使用されます。
  - ネットワークで共有される配列または文字列のシェア変数では、ネットワーク : バッファサイズプロパティ、ネットワーク : 要素サイズプロパティが使用されます。
  - ネットワークで共有される波形のシェア変数では、ネットワーク : バッファサイズプロパティ、ネットワーク : 各波形のポイントプロパティが使用されます。
  - ネットワークで共有される配列のシェア変数では、ネットワーク : バッファサイズプロパティ、ネットワーク : 要素サイズプロパティ、ネットワーク : 各波形のポイントプロパティが使用されます。
- ノードがデータを書き込むよう構成されている場合、シェア変数ノードの変数入力が必須になります。
- **シェア変数プロパティ**の**変数**ページは、以下の点が変更されました。
  - **データタイプ**プルダウンメニューの**カスタム**項目が、**カスタム制御器から**に変更されました。

  - **データタイプ**プルダウンメニューに、新しく**バリアント**項目が追加されました。
  - ネットワークバッファの要素サイズが表示されなくなりました。
  - **アクセスタイプ**を**読み取りのみ**または**書き込みのみ**に設定するには、データの読み取りまたは書き込みのみを行うよう構成されたシェア変数を作成します。読み取りのみまたは書き込みのみのソースにバインディングされたシェア変数を右クリックすると、ショートカットメニュー**書き込みに変更**および**読み取りに変更**が無効になります。
  - LabVIEW 8.0 では、シングルプロセスのシェア変数のタイムスタンプ情報を取得するには、シェア変数ノードを右クリックしてショートカットメニューから**タイムスタンプを表示**を選択します。LabVIEW 8.2 では、シングルプロセスのシェア変数のタイムスタンプ情報を取得するには、まずプロジェクト内でシェア変数を右クリックし、ショートカットメニューから**プロパティ**を選択し、**変数**ページで**タイムスタンプ有効**チェックボックスをオンにします。シェア変数のタイムスタンプ情報を取得したり、**タイムスタンプ出力**をシェア変数ノードへ追加するには、シェア変数ノードを右クリックしてショートカットメニューから**タイムスタンプを表示**を選択します。LabVIEW 8.0 で作成したシングルプロセスシェア変数を LabVIEW 8.2 にロードした場合、タイムスタンプはデフォルトで有効になります。
- LabVIEW 8.0 では、**変数を選択**ダイアログボックスのリストを選択できない項目はグレー表示されます。LabVIEW 8.2 では、リストから無効な項目を選択すると **OK** ボタンがグレー表示されます。

プロジェクトのその他の機能強化

LabVIEW 8.2 では、プロジェクトがその他以下の点で機能強化されました。

- LabVIEW 8.0 では、プロジェクトライブラリでタイプ定義を作成し、プライベートとしてタイプ定義のアクセススコープを設定する場合でも、プロジェクトライブラリの外でプライベートタイプ定義を参照できます。LabVIEW 8.2 では、ライブラリの外にあるプライベートタイプ定義を参照することはできません。
- XControl 内に VI を作成するには、XControl を右クリックしてショートカットメニューから**新規→ VI** を選択します。
- LabVIEW 8.0 では、デバッグ中にブレークポイントを設定した後、XControl の機能、プロパティ、メソッド VI を保存する必要があります。LabVIEW 8.2 では、デバッグ中にブレークポイントを設定した後、機能、プロパティ、メソッド VI を保存する必要はありません。
- LabVIEW プロジェクト内で、有効なライセンスがないライブラリのパスワードを入力した場合、項目をライブラリの外へドラッグアンドドロップできません。

- プロジェクトライブラリがパスワード保護されており、そのパスワードが LabVIEW パスワードキャッシュにない場合は、プロジェクトライブラリを右クリックしてショートカットメニューから**パスワードを入力**を入力してプロジェクトライブラリをロック解除します。

VI を複数のクライアントからリモート制御する

複数のクライアントから 1 つのアプリケーションまたは VI を同時にリモート制御できます。同時制御を可能にするには、VI は再入可能でなければなりません。再入可能 VI を作成するには、**ファイル→ VI プロパティ**を選択し、**カテゴリ**リストから**実行**を選択し、**再入実行**チェックボックスをオンにします。リモートフロントパネルに対する各クライアントからのリクエストに対して、その再入可能 VI のクローンが自動的に開きます。フロントパネルのリモート接続では、ウェブサーバ : VI アクセスリストプロパティを使用して、メモリにロードされているクローンへのアクセスをプログラム的に制限できます。

共有ライブラリファイルから関数をインポートする

Windows の `.dll` ファイル、Mac OS の `.framework` ファイル、Linux の `.so` ファイル内のエクスポートされた関数に対して VI を生成、更新できます。共有ライブラリファイルのラッパー VI を作成するには、**ツール→インポート→共有ライブラリ**を選択して画面の指示に従います。このとき、シェアライブラリファイルとヘッダ `.h` ファイルの名前を入力する必要があります。

『LabVIEW ヘルプ』の**目次**タブで**基本機能→テキストベースのプログラミング言語で書かれたコード呼び出し→操作手順→共有ライブラリファイルから関数をインポートする**を参照してください。

計測器ドライバテンプレート

LabVIEW 8.2 では、「新規計測器ドライバプロジェクトを作成」ウィザードに以下のテンプレートが新しく追加されました。

- **General Purpose（レジスタベース）**—クラス固有のテンプレートがないレジスタベースの計測器を使用します。一般的なレジスタベースの計測器には、VXI と PXI があります。
- **スペクトルアナライザ**—周波数範囲やスイープカプリングの設定などの基本的操作を制御します。テンプレートには、マーカの構成やクエリなどの上級機能も含まれています。

.NET および ActiveX の機能強化（Windows）

.NET コントロールのユーザインタフェーススレッドの依存性がなくなりました。したがって、VI またはサブ VI を .NET コントロール操作のためのユーザインタフェーススレットで実行するために設定する必要がなくなりました。デフォルトでは、すべての .NET コントロールが正しいスレッド

（ほとんどの場合はインタフェーススレッド）で実行されるよう設定されます。

LabVIEW は ActiveX/.NET コールバック VI でのデバッグをサポートしていません。ActiveX/.NET コールバック VI でブレークポイントを使用しても、LabVIEW はブレークポイントで一時停止しません。

LabVIEW 8.0 以前では、.NET Refnum プローブはリファレンスの 16 進数値のみを表示します。LabVIEW 8.2 では、.NET Refnum プローブは、ToString() メソッドを呼び出した結果を、.NET refnum、オブジェクトのタイプおよびハッシュコード、リファレンスの 16 進数値オブジェクトによってオブジェクトに表示します。値が無効な Refnum の場合に、このプローブを使用してブレークポイントを設定することができます。

LabVIEW 8.2 では、.NET および ActiveX メニューが以下のように変更されました。

- **ツール→ .NET & ActiveX** メニュー項目である **ActiveX コントロールをパレットに追加**、**ActiveX コントロールをパレットに追加**は、**ツール→インポート→ .NET コントロールをパレットに追加**および**ツール→インポート→ ActiveX コントロールをパレットに追加**にそれぞれ変更されました。
- **ツール→ .NET&ActiveX → ActiveX プロパティを参照**メニュー項目は、**表示→ ActiveX プロパティブラウザ**に変更されました。

NI サンプルファインダの機能強化

最新サンプルフォルダはデフォルトで NI サンプルファインダの**参照**上に表示されます。**最新**サンプルフォルダに表示するサンプルの最大数を設定するには、NI サンプルファインダで**設定**ボタンをクリックし、**一般**タブをクリックします。

NI サンプルファインダの機能強化の詳細については、「NI サンプルファインダヘルプ」を参照してください。NI サンプルファインダのヘルプを表示するには、NI サンプルファインダで**ヘルプ**ボタンをクリックします。

ソース管理の機能強化

LabVIEW プロジェクトのソース制御器の詳細については、『LabVIEW ヘルプ』の**目次**タブにある**基本機能→プロジェクトを構成および管理する**のブックアイコンを参照してください。

以下のサードパーティソース制御器のプロバイダは、LabVIEW 8.2 プロフェッショナル開発システムとの統一性および基本機能が動作保証されています。

- Seapine Surrond SCM
- Borland StarTeam
- Telelogic Synergy

- PushOK（CVS および SVN プラグイン）
- ionForge Evolution

メモ 現在、ionForge Evolution 2.8 以上は LabVIEW で動作確認済みです。

また、追加のサードパーティソース管理プロバイダは、LabVIEW の VI のグラフィカルディファレンシング（相違点認識）を行うためのプロシージャとの互換性が LabVIEW 8.0 より向上しました。

LabVIEW プロジェクトフォルダのソース管理操作

LabVIEW プロジェクト内の項目のフォルダに対してソース管理操作を行うことができます。LabVIEW では、階層内のすべての適切な項目に対して操作を適用します。たとえば、ソース管理にファイルを追加すると、ソース管理にまだ追加されていないフォルダ内のファイルだけが追加されます。項目のフォルダに対してソース管理操作を実行すると、それらのフォルダに関連するソース管理の構成オプションがすべて表示されます。

メモ ソース管理操作は、フォルダ内にあるプロジェクトライブラリの中にある項目に対しては実行されません。プロジェクトライブラリの中にある項目に対してソース管理操作を行うには、それらの項目を手動で選択する必要があります。ただし、プロジェクトライブラリ内の 1 つのフォルダに対してソース管理を実行できます。

ソース管理操作での未保存のファイルの処理

変更が保存されていないファイルをチェックインしようとすると、**保存されていないファイル**ダイアログボックスが開き、変更を保存するか無視するかを選択できます。このダイアログボックスは、**ソース管理操作**ダイアログボックスで **OK** ボタンをクリックしても、ソース管理操作に含む LabVIEW ファイルタイプに変更が保存されていない場合に表示されます。またこのダイアログボックスは、変更が保存されていないライブラリまたはプロジェクトファイルで**ツール→ソース管理→差異を表示**を選択して表示することもできます。

ソース管理のその他の機能強化

LabVIEW 8.2 では、ソース管理がその他以下の点で機能強化されました。

- LabVIEW 8.0 では、未保存ファイルに対しても**ツール→ソース管理**メニューで**ソース管理に追加**オプションが表示されました。LabVIEW 8.2 では、ファイルを保存するまで**ソース管理に追加**オプションは表示されません。
- ソース管理にファイルを追加する際に `vi.lib` ディレクトリおよび `instr.lib` ディレクトリのファイルを除外するには、**ツール→ソース管理→ソース管理を構成**を選択して、**vi.lib を除く**チェックボック

スと **instr.lib を除く**チェックボックスをオンにします。ソース管理をファイルを追加する際に依存性も追加されるよう LabVIEW を設定すると、これらのオプションを選択することで不要なファイルを操作対象から除外することができます。

- **Perforce リビジョン履歴**ダイアログボックスに、**このリビジョンに同期**および**チェンジリストを説明**というオプションが追加されました。これらのオプションは、**履歴**リストでリビジョンを右クリックすると表示されます。
- **Perforce リビジョン履歴**ダイアログボックスの**チェンジリスト内のアクション**フィールドには、現在のチェンジリストで実行された操作が表示されます。たとえば、追加、編集、削除、分岐、統合などの操作が挙げられます。このフィールドにはまた、操作対象となったファイルのタイプも表示されます。
- **(Windows)** Perforce のコアインストーラをインストールすると、**オプション**ダイアログボックスの**ソース管理**ページの**ソース管理プロバイダ名**プルダウンメニューに **Perforce SCM** が表示されます。

TDM の機能強化

LabVIEW 8.2 では、TDM が以下の点で機能強化されました。

TDM ストリーミングファイル形式

LabVIEW 8.2 では、バイナリデータを保存するための TDM ストリーミング（.tdms）ファイル形式が追加されました。.tdms ファイル形式は、LabVIEW 8.0 以前で使用されていた .tdm ファイル形式よりも書き込みが高速です。また、.tdms ファイル形式は、より簡単なインタフェースでプロパティを定義できます。

TDM VI および関数の詳細については、このドキュメントの「*TDM ストリーム VI および関数*」セクションを参照してください。

ユーザ定義 TDM/TDMS プロパティ

バイナリ計測ファイルに対してユーザ定義プロパティを作成し、NI-DAQmx プロパティを指定することができます。.tdm ファイルまたは .tdms ファイルのユーザ定義プロパティを構成するには、「計測ファイルへ書き込み（Write To Measurement File）」、「データを書き込む（Write Data）」、「プロパティを設定（Set Properties）」、「プロパティを取得（Get Properties）」Express VI を使用します。

プロパティを構成するには、「計測ファイルへ書き込み」Express VI の構成ページで**上級**ボタンをクリックします。プロパティを構成するには、「データを書き込み」、「プロパティを設定」、「プロパティを取得」Exprress VI の構成ページを表示します。

ウェブサービスをインポートする（Windows）

LabVIEW 8.2 では、複雑な管理を必要とすることなくウェブサービスを使用できます。あらゆるウェブサービスを VI のプロジェクトライブラリに変換して、ウェブサービスをスタンドアロンローカルコンピュータで使用するような感覚でプログラムすることができます。このためには、ウェブサービス記述言語（WSDL）による有効な URL が必要です。WSDL は、ウェブサービスとその機能性を記述するための XML 形式言語です。**ツール→インポート→ウェブサービス**を選択すると、ウェブサービスのメソッドをインポートして VI のライブラリを作成するためのウィザードが起動します。

メモ 「ウェブサービスをインポート」ウィザードを使用するには、コンピュータに .NET Framework 1.1 Service Pack 1 またはそれ以降がインストールされていなければなりません。

ウェブサービスのインポートの詳細については、『LabVIEW ヘルプ』の**目次**タブで**基本機能→ Windows コネクティビティ→概念→ウェブサービス**ブックを参照してください。

外部コード関数の変更

メモリマネージャインタフェースの外部ソースコードによって使用される特定の部分で、データタイプ `int32` の代わりに、`size_t` が使用されるようになりました。この変更は、既存の DLL と互換性があります。

メモリマネージャ関数の詳細については、『LabVIEW ヘルプ』の「*Memory Manager Functions*」のトピックを参照してください。

371780B-0112 0806