

LabVIEW™ アップグレードノート

アップグレードノートでは、Windows、Mac OS、Linux 用の LabVIEW をバージョン 8.6 にアップグレードする手順、アップグレードの際に発生する可能性のある問題、および新機能について説明します。発生し得る互換性の問題については、旧バージョンの LabVIEW で保存した VI を新しいバージョンの LabVIEW にロードする前に、このアップグレードノートをお読みください。この新しいバージョンの LabVIEW でファイルをロードする前に、旧バージョンの LabVIEW で保存したすべての LabVIEW ファイルのバックアップコピーを作成することを検討してください。

LabVIEW 7.1 以前のバージョンから LabVIEW 8.6 にアップグレードする場合、ナショナルインスツルメンツでは LabVIEW 7.1 から LabVIEW 8.6 までの各バージョンで追加された改善点、変更点、機能の詳細について、これらのアップグレードノート以外に以下のドキュメントを確認することを推奨します。

- 『**LabVIEW 8.0 アップグレードノート**』 — 「アップグレードおよび互換性の問題」セクションおよび「LabVIEW 8.0 の機能および変更点」セクションには、アップグレードに関する重要な情報が記載されています。『LabVIEW 8.0 アップグレードノート』にアクセスするには、ナショナルインスツルメンツのウェブサイト ni.com/jp/info で info code として upnote8 と入力してください。
- 『**LabVIEW 8.2 アップグレードノート**』 — 「アップグレードおよび互換性の問題」セクションおよび「LabVIEW 8.2 の機能および変更点」セクションには、アップグレードに関する重要な情報が記載されています。『LabVIEW 8.2 アップグレードノート』にアクセスするには、ナショナルインスツルメンツのウェブサイト ni.com/jp/info で info code として upnote82 と入力してください。
- 『**LabVIEW 8.5 アップグレードノート**』 — 「アップグレードおよび互換性の問題」セクションおよび「LabVIEW 8.5 の機能および変更点」セクションには、アップグレードに関する重要な情報が記載されています。『LabVIEW 8.5 アップグレードノート』にアクセスするには、ナショナルインスツルメンツのウェブサイト ni.com/jp/info で info code として upnote85 と入力してください。

LabVIEW のプログラミング概念、LabVIEW を使用するための段階的な手順、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リファレンスなどの情報の他、LabVIEW 8.6 の詳細は、『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』にアクセスするには、**ヘルプ→LabVIEW ヘルプを検索**を選択します。

目次

LabVIEW 8.6 へアップグレードする.....	2
VI を変換する.....	4
モジュール、ツールキット、計測器ドライバをアップグレードする.....	4
ナショナルインストルメンツの追加ソフトウェアをアップグレードする.....	6
旧バージョンの LabVIEW からアップグレードする.....	7
アップグレードおよび互換性の問題.....	8
LabVIEW 8.5 からアップグレードする.....	8
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 (Windows).....	11
アプリケーションビルダの共有コンポーネント.....	11
LabVIEW 8.2 からアップグレードする.....	11
LabVIEW 8.0 からアップグレードする.....	17
LabVIEW 7.x からアップグレードする.....	22
LabVIEW 6.x からアップグレードする.....	38
LabVIEW 5.x 以前のバージョンからアップグレードする.....	41
LabVIEW 8.6 の機能および変更点.....	42
LabVIEW をインストールする.....	42
LabVIEW のドキュメント.....	42
新しいサンプル VI.....	42
ブロックダイアグラムの改善点.....	42
フロントパネルの改善点.....	44
環境の改善点.....	44
新しく追加または変更された VI、関数、ノードでの改善点.....	47
新しいクラス、プロパティ、メソッド、イベント.....	52
ブロックダイアグラムを自動的にクリーンアップ.....	52
クイックドロップを使用してオブジェクトを配置.....	53
複数のオブジェクトのプロパティを編集.....	53
LabVIEW ウェブサービス (Windows、ベースパッケージにはありません).....	53
NI 分散システムマネージャ.....	53
LabVIEW MathScript での改善点 (ベースパッケージにはありません).....	54
固定小数点数のオーバーフローを管理する.....	56
LLB を統合する (ベースパッケージにはありません).....	57

LabVIEW 8.6 へアップグレードする

LabVIEW の以前のバージョンからアップグレードしている場合、最初にこの「LabVIEW 8.6 へアップグレードする」セクション、および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」を参照してください。ここで、x.x はアップグレードする前のバージョンを示します。

以下は、新しいバージョンの LabVIEW へのアップグレードに関連したタスクを完了するための推奨する手順、およびこれらのタスクを完了する際読むべきドキュメントについて説明します。ナショナルインストルメンツでは、LabVIEW の新しいバージョンにアップグレードする前に、『LabVIEW リリースノート』とこのドキュメントの両方をお読みになることを推奨します。

1. LabVIEW をインストールする前にすべての互換性の問題について理解したことを確認するには、新しいバージョンの LabVIEW をインストールする前に、このドキュメントの以下のセクションを参照してください。
 - **LabVIEW 8.6 へアップグレードする**—このセクションには、ツールキットとモジュールのアンインストール、旧バージョンの LabVIEW の環境設定と `user.lib` ファイルのコピー、VI の LabVIEW 8.6 への変換の操作手順が含まれています。
 - **アップグレードおよび互換性の問題**—このセクションには、旧バージョンの LabVIEW から新しいバージョンの LabVIEW にアップグレードする VI に影響する可能性がある互換性の問題が含まれています。アップグレード中の LabVIEW のバージョンに適用するサブセクションを参照してください。



メモ また、一部の互換性の問題で VI を評価できるテストをダウンロードするには、ナショナルインスツルメンツのウェブサイト ni.com/jp/info で info code として `exqmex` と入力すると参照できます。

- (オプション) **LabVIEW 8.6 の機能および変更**—このセクションには、このバージョンの LabVIEW の新機能についての簡単な説明が含まれています。これらの機能を使用する詳細は、『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→LabVIEW ヘルプを検索** を選択します。
2. (オプション) 旧バージョンの LabVIEW をアンインストールします。
 3. LabVIEW のアップグレードバージョンをインストールして、アクティブ化します。LabVIEW のインストールに関連したすべてのタスクを完了したことを確認するには、『LabVIEW リリースノート』の以下のセクションを参照してください。
 - システム要件
 - 「LabVIEW 8.6 をインストールする」とインストールしているプラットフォームに適切なサブセクション
 - LabVIEW プラットフォーム DVD の代わりに CD から LabVIEW ツールキットまたはモジュールをインストールしている場合、「LabVIEW アドオンをインストールする」サブセクション
 - **(Windows)** 「LabVIEW ライセンスをアクティブ化する」とすべてのサブセクション
 - (オプション) 「ハードウェアのインストールと構成」とインストールしているプラットフォームに適切なサブセクション
 - その他の情報
 4. 最新バージョンの LabVIEW で解決された問題、最新バージョンの LabVIEW における既知の問題に関する情報、『LabVIEW ヘルプ』に反映されていないドキュメントの追加事項の詳細は、『LabVIEW Readme』をお読みください。『LabVIEW Readme』にアクセスするには、`labview\readme` ディレクトリを参照して `readme.html` ファイルを開いてください。
 5. 旧バージョンの LabVIEW から環境設定をコピーします。環境設定をコピーする詳細については、この章の「[旧バージョンの LabVIEW の環境設定をコピーする](#)」セクションを参照してください。
 6. 旧バージョンの LabVIEW から `user.lib` ファイルをコピーします。`user.lib` ファイルをコピーする詳細については、このドキュメントの「[旧バージョンの LabVIEW の user.lib をコピーする](#)」セクションを参照してください。
 7. VI を LabVIEW 8.6 に変換します。旧バージョンの LabVIEW で保存された VI を変換する詳細は、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。

VI を変換する

NI 営業部に問い合わせ、使用するコードを LabVIEW 8.6 と互換性がある VI 形式にアップグレードするための変換ソフトウェアを入手せずに、LabVIEW 6.0 以前のバージョンで保存された VI を開くことはできません。LabVIEW 6.0 以降のバージョンで保存された VI を開くと、LabVIEW 8.6 はその VI を自動的に変換してコンパイルします。VI を LabVIEW 8.6 で保存しない場合、VI にアクセスする度に余分なメモリリソースを使用する変換処理が実行されます。また、再コンパイルなどの未保存の変更がある VI のパフォーマンスが実行時に著しく低下する場合があります。



メモ LabVIEW 8.6 で保存された VI は、旧バージョンの LabVIEW でロードできません。**ファイル→旧バージョン用に保存**を選択して、VI を LabVIEW 8.5、8.2、8.0 で実行できる形式で保存することができます。VI を LabVIEW 8.6 で保存する前に、LabVIEW 8.5、8.2、8.0 でも使用する VI のバックアップコピーを保管してください。

すべての VI を一括変換するのに十分なメモリがコンピュータにない場合、VI を段階的に変換する必要があります。変換する VI の階層を確認して、下位階層にあるサブ VI のロードを開始し、そのサブ VI を保存します。次に、階層の上位レベルに徐々に進みます。上位レベル VI を開いて変換する操作は最後に行います。また、**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 中の VI はアルファベット順に変換されます。変換処理で最初に上位レベル VI に遭遇した場合、上位レベル VI を最初に開いたかのように、一括コンパイルの処理でほぼ同じメモリ量が必要とされます。

ヘルプ→バージョン情報を選択して、現在使用中のメモリ容量の概要を表示して、メモリ使用量を監視できます。

モジュール、ツールキット、計測器ドライバをアップグレードする

LabVIEW の旧バージョンからアップグレードしている場合、LabVIEW の旧バージョン用にインストールしたモジュール、ツールキット、または計測器ドライバの最新の互換性のあるバージョンをインストールする必要があります。LabVIEW プラットフォーム DVD には LabVIEW 8.6 と互換性のあるモジュールとツールキットが含まれています。LabVIEW プラットフォーム DVD がないモジュールやツールキットで、LabVIEW の現行バージョンと互換性のある LabVIEW モジュールおよびツールキットの詳細については、ナショナルインスツルメンツのウェブサイト ni.com/jp/info で info code として `compat` と入力してください。

NI モジュールとツールキット

以下の表は、使用するオペレーティングシステムと LabVIEW アドオンに応じて、LabVIEW プラットフォーム DVD、あるいはモジュールまたはツールキットのインストール CD を使用するかどうかをリストします。

オペレーティングシステム	使用するメディア	主な留意点
Windows	DVD	LabVIEW プラットフォーム DVD を使用して、LabVIEW 8.6 および LabVIEW 8.6 と互換性のあるモジュールとツールキットのバージョンをインストールします。また、購入されていないモジュールまたはツールキットを評価することができます。LabVIEW プラットフォーム DVD では、以前のバージョンをアンインストールまたは変更せずに、LabVIEW 8.6 とツールキットの最新バージョンをインストールすることができます。LabVIEW、モジュール、ツールキットをインストールする詳細については、『LabVIEW リリースノート』を参照してください。
Windows、 LabVIEW プラット フォーム DVD に、 モジュール / ツール キットが付属してい ない場合	CD	モジュールまたはツールキットを購入した際入手したインストール CD を使用します。インストール CD を使用する前に、インストールしたいモジュールまたはツールキットの互換性のあるバージョンがあることを確認してください。LabVIEW の現在のバージョンでサポートされている LabVIEW モジュールおよびツールキットの詳細は、ナショナルインスツルメンツのウェブサイト ni.com/jp/info で info code として compat と入力してください。次に、LabVIEW 8.6 ディレクトリに互換性のあるモジュールとツールキットをインストールします。LabVIEW の以前のバージョンで保存した VI のすべてを一括コンパイルします。 詳細は、このドキュメントの「 LabVIEW の一括コンパイル 」セクションを参照してください。
Mac/Linux		



メモ ツールキットの一部のバージョンは LabVIEW 8.6 で動作しません。互換性がないツールキットをインストールすると、一部のツールキット機能や LabVIEW が不正な動作をする可能性があります。ナショナルインスツルメンツでは、ツールキットをインストールする前に互換性について確認することを推奨します。LabVIEW の現在のバージョンでサポートされている LabVIEW モジュールおよびツールキットの詳細は、ナショナルインスツルメンツのウェブサイト ni.com/jp/info で info code として compat と入力してください。互換性のないバージョンをインストールして、LabVIEW 8.6 のインストールが破損した場合、まずツールキットをアンインストールして、次に**プログラムの追加と削除**を使用して、LabVIEW のインストールを修復します。

計測器ドライバ

最新の計測器ドライバをインストールして、使用する予定の計測器ハードウェアを制御、通信する必要があります。LabVIEW の以前のバージョンと計測器ドライバをインストールした場合、以下のいずれかの方法を使用し、LabVIEW 8.6 でサポートする計測器ドライバを再インストールする必要があります。

- **NI モジュール式計測器ドライバ**—NI デバイスドライバ DVD または CD を使用して、NI モジュール式計測器ドライバをインストールします。
- **NI プラグアンドプレイ計測器ドライバ**—NI 計測器ドライバファインダを使用して、LabVIEW 開発環境を終了せずに、LabVIEW プラグアンドプレイ計測器ドライバを検索およびインストールします。
- **IVI ドライバまたは非認証の計測器ドライバ**—ナショナルインスツルメンツのウェブサイトの [計測器ドライバネットワーク](#) を使用して、IVI ドライバまたは非認証のドライバを検索します。



メモ NI 計測器ドライバファインダを使用して計測器ドライバを再インストールした場合、ナショナルインスツルメンツでは `labview\instr.lib` ディレクトリを一括コンパイルすることを推奨します。

他社製アドオン

他社製 LabVIEW アドオンの製造販売元に問い合わせ、使用するオペレーティングシステムでアドオンが LabVIEW 8.6 と互換性があるかどうかを確認します。アドオンに関連する VI のすべてを一括コンパイルすることを確認してください。

詳細は、このドキュメントの「LabVIEW の一括コンパイル」セクションを参照してください。

LabVIEW の一括コンパイル

最後に LabVIEW の旧バージョンで保存された VI を開くと、LabVIEW は VI の変換とコンパイルを自動的にを行います。LabVIEW の最新バージョンで VI を保存しないと、余分なメモリリソースを消費する変換プロセスが VI にアクセスするたびに発生します。LabVIEW プラットフォーム DVD 上にない LabVIEW モジュールとツールキットをインストールするか、他社製アドオンをインストールする場合、ナショナルインスツルメンツではモジュール、ツールキット、他社製アドオンでインストールした VI のすべてを一括コンパイルすることを推奨します。

VI の一括コンパイルの詳細は、**基本機能→VI およびサブ VI を作成する→操作手順→VI を保存する→VI の一括コンパイル** トピックを参照してください。

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。Upgrade Advisor にアクセスして NI TestStand 3.5 以降を購入するには、ナショナルインスツルメンツのウェブサイト ni.com/jp/info にアクセスして、info code として `exd8yy` と入力します。

NI TestStand 3.5 と NI TestStand 4.0 で以下の LabVIEW 8.6 Express VI の構成を試みると、エラーが発生します。

- データストレージを閉じる
- フォーミュラ
- プロパティを取得
- データストレージを開く
- データを読み取る

- プロパティを設定
- スペクトル計測
- データを書き込む

このエラーに関する詳細情報を参照するには、ナショナルインストルメンツのウェブサイト ni.com/jp/info で info code として rdtf10 と入力してください。



メモ NI TestStand 4.1 以降のバージョンではこの問題は解決されています。

LabVIEW と TestStand の問題の詳細については、ご使用の NI TestStand のバージョン用の `Readme.html` ファイル (NI TestStand CD および `<TestStand>\Doc` ディレクトリにあります。) を参照してください。

LabVIEW 8.x では、NI Spy 2.3 以降を使用する必要があります。NI Spy 2.5 は、ナショナルインストルメンツのデバイスドライバ CD から利用することができます。

LabVIEW 8.6 では Measurement Studio 8.0 以降をサポートします。Upgrade Advisor へのアクセスおよび Measurement Studio 8.0 以降のバージョンを注文するには、ナショナルインストルメンツのウェブサイト ni.com/jp/info で info code として exd8yy と入力してください。

旧バージョンの LabVIEW からアップグレードする

LabVIEW 8.6 は、旧バージョンの LabVIEW をアンインストールせずにインストールすることができます。LabVIEW の各バージョンはコンポーネントを共有することはありますが、新しいバージョンが異なるディレクトリにインストールされるため、LabVIEW の新しいバージョンにアップグレードする操作はコンピュータにインストールされている旧バージョンの LabVIEW に影響しません。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 (ここで、`x.x` はバージョン番号) にインストールされます。

LabVIEW の既存のバージョンを置換する

既存のバージョンの LabVIEW を置換するには、既存のバージョンの LabVIEW をアンインストールして、LabVIEW 8.6 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 8.6 に置換することもできます。アンインストーラ処理では、`labview` ディレクトリで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、LabVIEW は `.lib` ファイルに保存した VI および制御器を含む `vi.lib` ディレクトリの `.lib` ファイルをアンインストールします。VI および制御器は `user.lib` ディレクトリに保存して、**制御器パレット** および **関数パレット** に追加します。

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 8.6 環境設定ファイルを旧バージョンのファイルで置き換えると、旧バージョン以降に追加された環境設定がオーバーライドされる可能性があります。

LabVIEW 8.6 をインストールした後、LabVIEW 環境設定ファイルを LabVIEW 8.6 ディレクトリにコピーします。

(Windows) LabVIEW は環境設定を labview ディレクトリの labview.ini ファイルに保存します。

(Mac OS) LabVIEW 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) LabVIEW は環境設定をホームディレクトリの .labviewrc ファイルに保存します。

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 8.6 をインストールした後、ファイルを LabVIEW 8.6 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 のバージョンによって異なるアップグレードおよび互換性の問題点の詳細は、以下のセクションを参照してください。

新しいバージョンの LabVIEW の既知の問題、その他の互換性に関する問題、LabVIEW 8.6 の最新の追加機能の詳細は、labview ディレクトリの readme.html を参照してください。

LabVIEW 8.5 からアップグレードする

LabVIEW 8.5 から LabVIEW 8.6 にアップグレードする際、以下の互換性の問題が発生する場合があります。



メモ LabVIEW 8.5.x からアップグレードする際に遭遇する可能性がある問題の詳細は、ナショナルインスツルメンツのウェブサイト ni.com/jp/info で info code として upnote85 と入力して参照することもできます。

サポートされているプラットフォーム

LabVIEW 8.6 には、サポートされているプラットフォームで以下の変更が追加されています。

- LabVIEW 8.6 では、PowerPC プロセッサを搭載した Macintosh コンピュータをサポートしていません。

システム要件

(Windows) LabVIEW 8.6 の最小インストールには 1.6 GB のディスク容量が必要です。

(Mac OS) LabVIEW 8.6 の最小インストールには 262 MB 以上のディスク容量、完全インストールには 828 MB 以上のディスク容量が必要です。

(Linux) LabVIEW 8.6 の最小インストールには 365 MB 以上のディスク容量、LabVIEW の完全インストールには 651 MB 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 8.6 で変更されていません。そのため、LabVIEW 8.6 での変更点がドキュメントの内容に反映されていない可能性があります。

- 『LabVIEW スタートアップガイド』マニュアル
- **(LabVIEW 8.2, 8.5, 8.6)** 『LabVIEW 基本機能』マニュアル



メモ 『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なため、更新された内容の詳細は、『LabVIEW ヘルプ』の**目次**タブの**基本機能**ブックを参照してください。

VI と関数の動作変更

以下の VI および関数の動作は、LabVIEW 8.6 で変更されました。

レポート生成 VI

「レポート生成 (Report Generation)」VI は、LabVIEW クラスを使用して書き換えられました。**レポート入力**制御器と**レポート出力**表示器は、リファレンス番号データタイプから LabVIEW クラスデータタイプに変更されました。タイプ定義 Refnum を右クリックして、定数、制御器、表示器を作成しなかった場合、これらのオブジェクトを更新できないため、VI が正常に動作しないことがあります。また、**レポート入力**と**レポート出力**パラメータの以前の refunum データタイプを呼び出すすべてのリファレンス呼び出しノードは予想どおり動作しなくなります。

レポート生成 VI を使用して HTML レポートを作成して、ターゲットでその VI を実行する場合、レポートを生成する際にターゲットを基準とすることを確認してください。ホストコンピュータで HTML レポートを作成して、ターゲットを参照せずにターゲットにデプロイする場合、VI は壊れて表示され、実行できません。

外部コード (DLL および CINS)

メモリマネージャ関数には、1つのメモリゾーンつまり DS (データスペース) があります。LabVIEW のメモリを管理する C、C++ CIN、DLL を使用して作業する場合、DS の対応する関数ですべてのリファレンスを AZ (アプリケーションゾーン) メモリ関数に置換します。

その他の VI および関数の動作における変更点

LabVIEW 8.6 では、以下のとおり、その他の VI および関数の動作変更があります。

- 数学および信号処理の VI の多くは再入可能でない VI から再入可能 VI へ変わっています。これらの変更により、インスタンス間でクローンを共有する目的で一連の再入可能 VI からこれらの VI の多くを呼び出さないでください。インスタンス間でクローンを共有する VI セットから呼び出しができない VI の詳細は、ナショナルインストルメンツのウェブサイト ni.com/jp/info で info code として exrehi と入力してください。
- 単一プロセスのシェア変数はターゲット相対 (関係) に強制されます。単一プロセスのシェア変数を絶対 (関係) に構成することはできません。
- 空のパスをライブラリ関数呼び出しノードの**パス入力**の入力に配線すると、エラーが返されなくなります。

廃止された VI および関数

LabVIEW 8.6. は以下の VI と関数をサポートしていません。

- **セマフォ作成 (Create Semaphore)**—「セマフォリファレンス取得 (Obtain Semaphore Reference)」VI を代用してください。「セマフォリファレンス取得」VI は「セマフォ作成」VI とは異なります。これは、同じ名前でも複数のセマフォを作成するために「セマフォ作成」VI を使用する場合は、そのセマフォに対して単一リファレンスの複数のコピーが作成されるからです。ただし、この「セマフォリファレンス取得」VI を使用して同じ名前のセマフォを参照する複数のリファレンスを取得すると、各リファレンスには固有の番号が付けられます。LabVIEW は自動的に既存の VI を変換して「セマフォリファレンス取得」VI を使用しないため、旧バージョンの LabVIEW で保存された VI は手動で更新する必要があります。

- **セマフォ破棄 (Destroy Semaphore)**—「セマフォリファレンス解放 (Release Semaphore Reference)」VI を代用してください。「セマフォリファレンス解放」VI は「セマフォ破棄」VI とは異なります。これは、セマフォを破棄するために「セマフォ破棄」VI を使用する場合は、そのセマフォに対する別のリファレンスコピーもすべて破棄されるからです。ただし、リファレンスをセマフォに解放するために「セマフォリファレンス解放」VI を使用する場合は、そのセマフォに対する他のリファレンスは有効な状態で残り、そのセマフォに対して他のリファレンスが存在しない場合はそのセマフォを破棄します。LabVIEW は自動的に既存の VI を変換して「セマフォリファレンス解放」VI を使用しないため、旧バージョンの LabVIEW で保存された VI は手動で更新する必要があります。



メモ 予期せぬ結果を回避するには、「セマフォ作成 (Create Semaphore)」または「セマフォ破棄 (Destroy Semaphore)」VI で使用するリファレンスを「セマフォリファレンス取得」VI または「セマフォリファレンス解放」VI (またはその逆) に使用しないでください。

- **Xmath スクリプトノード**—MathScript ノードを代用します。MathScript 構文は Xmath 構文と異なるため、MathScript ノードで動作する既存のスクリプトを変更する必要があります。

プロパティ、メソッド、イベントの動作変更

以下のプロパティ、メソッド、イベントの動作は LabVIEW 8.6 で変更されました。

- シーングラフ表示クラスのカメラコントローラ: タイププロパティは、**転置値**を含みます。
- シーンウィンドウクラスのカメラコントローラ: タイププロパティは、**転置値**を含みます。

廃止されたプロパティ、メソッド、イベント

LabVIEW 8.6 以降では、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。

- VI クラスの制御器の値: 設定 (平坦化) メソッド 制御器の値: 設定メソッドを代用してください。
- VI クラスの制御器の値: すべてを取得 (平坦化) メソッド 制御器の値: すべてを取得メソッドを代用してください。
- VI クラスの制御器の値: 取得 (平坦化) メソッド 制御器の値: 取得メソッドを代用してください。
- VI プロパティ (ActiveX) クラスの VIModificationBitSet プロパティ。VIModificationBitSet2 プロパティを代用してください。
- VI プロパティクラスの修正: VI 修正ビットセットプロパティ。新規の修正: VI 修正ビットセットプロパティを代用してください。LabVIEW 8.5 以前のバージョンでは、修正: VI 修正ビットセットプロパティは 32 ビットの値を返します。LabVIEW 8.6 では、新規の「修正: VI 修正ビットセット」プロパティは 64 ビット値を返します。

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 8.6 以降で変更されました。

クラス	LabVIEW 8.5 での名前	LabVIEW 8.6 での名前	タイプ
G オブジェクト	境界：高さ	境界：領域の高さ	プロパティ
G オブジェクト	境界：幅	境界：領域の幅	プロパティ
プロジェクト項目	ディスクから接続解除	自動更新を停止	メソッド
ツリー制御器	展開 / 収縮記号：インデントレベル 0 で表示	展開 / 収縮記号：ルートで記号を表示	プロパティ
VI	制御器の値：設定 (バリエーション)	制御器の値：設定	メソッド
VI	制御器の値：取得 (バリエーション)	制御器の値：取得	メソッド
VI	制御器の値：すべてを取得 (バリエーション)	制御器の値：すべて取得	メソッド

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 (Windows)

LabVIEW を使用して、アプリケーションまたはフロントパネルをリモートで参照することをユーザに許可します。LabVIEW は、5、20、50 または無制限数のクライアントが一度にフロントパネルにリモートで接続することを許可するライセンスをサポートしています。サーバコンピュータでは 1 つのライセンスのみ許可されます。許可したい数に対応できる接続を許可するライセンスを購入します。LabVIEW 8.5.1 以前のバージョン用にリモートフロントパネルライセンスをアップグレードした場合、既存のシリアル番号を使用して、NI License Manager で同等の新規のライセンスをアクティブ化する必要があります。

アプリケーションビルダの共有コンポーネント

ローカルコンピュータ上ですべてのバージョンの LabVIEW で共有する共有ライブラリを作成するには、コンポーネントをインストールします。LabVIEW 8.6 のインストール後に旧バージョンの LabVIEW をインストールすると、共有コンポーネントは旧バージョンによって置換されます。現行バージョンの LabVIEW で共有ライブラリの作成を試行すると、共有コンポーネントに LabVIEW 8.6 に必要な機能がないため、エラーが発生します。この問題を解決するには、LabVIEW 8.6 を再インストールします。

LabVIEW 8.2 からアップグレードする

LabVIEW 8.2 から LabVIEW 8.6 にアップグレードする際、以下の互換性の問題が発生する場合があります。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 8.5 からアップグレードする](#)」セクションを参照してください。



メモ LabVIEW 8.2 からアップグレードする際に遭遇する可能性がある問題の詳細は、ナショナルインスツルメンツのウェブサイト ni.com/jp/info で info code として ex5d8c と入力して参照することもできます。

サポートされているプラットフォーム

LabVIEW 8.5 以降には、サポートされているプラットフォームで以下の変更点が含まれています。

- LabVIEW 8.5 以降は Windows Vista x86 および x64 をサポートしています。
- LabVIEW 8.5 以降は Intel および PowerPC プロセッサ搭載の Macintosh コンピュータをサポートしています。

システム要件

(Windows) LabVIEW 8.5 の最小インストールには 1.2 GB のディスク容量が必要です。LabVIEW 8.6 の最小インストールには 1.6 GB のディスク容量が必要です。

(Mac OS) LabVIEW 8.5 の最小インストールには 502 MB 以上のディスク容量、完全インストールには 734 MB 以上のディスク容量が必要です。LabVIEW 8.6 の最小インストールには 262 MB 以上のディスク容量、完全インストールには 828 MB 以上のディスク容量が必要です。

(Linux) LabVIEW 8.5 の最小インストールには 450 MB 以上のディスク容量、LabVIEW の完全インストールには 640 MB 以上のディスク容量が必要です。LabVIEW 8.6 の最小インストールには 365 MB 以上のディスク容量、完全インストールには 651 MB 以上のディスク容量が必要です。

Windows Vista 互換性の問題

LabVIEW 8.5 以降では、以下の機能上の変更点を含める 32 および 64 ビットシステムの Windows Vista OS においてサポートされています。

「In Port」および「Out Port」VI では、Vista OS ではセキュリティの理由で推奨されないシステムの I/O ポートのすべてへの読み書きのアクセスが許可されているため、**関数**パレットに表示されません。

- **(Windows Vista x86)** VI のコンポーネントは適切にインストールされますが、Windows Defender ログで「unsigned」と表示されます。VI は正常に動作します。
- **(Windows Vista x64)** これらの VI はエラー -4850 を返します。

VI と関数の動作変更

以下の VI および関数の動作は LabVIEW 8.5 以降で変更されました。

解析 VI/ 関数の改善点

LabVIEW の各バージョンで、ナショナルインストルメンツでは LabVIEW および C 関数の多くのアルゴリズムを改善しています。また、最新のコンパイラが使用されるように LabVIEW をアップデートしています。コンピュータのハードウェアとソフトウェアの変更に伴って、これらの改善点は LabVIEW 8.2 以前のバージョンと LabVIEW 8.5 以降間で結果が数値的に異なる原因となる場合があります。倍精度浮動小数点数を比較すると、 $1E-16$ の順序が多少異なることに気が付くかもしれません。浮動小数点数の比較の詳細は、ナショナルインストルメンツのウェブサイト (ni.com/jp/info) でインフォコードとして `exiigr` と入力してください。

数学 VI

LabVIEW 8.5 以降では、以下の**数学 VI** が変更されました。

- **f(x) 全ゼロ検索 (Find All Zeroes of f(x))** —この VI の名前は「f(x) 全ゼロ検索 (Find All Zeros of f(x))」VI に変更されました。(英語版のみに適用)
- **f(x) のゼロと極値 (Zeroes and Extrema of f(x))** —この VI の名前は「f(x) のゼロと極値 (Zeros and Extrema of f(x))」に変更されました。(英語版のみに適用)

数値関数

LabVIEW 8.5 以降には、以下の数値関数での変更点が追加されています。

- **切り上げ整数化 (Round To +Infinity)** —この関数は「切り上げ整数化 (Round Toward +Infinity)」という名前に変更されました。(英語版のみに適用)
- **切り下げ整数化 (Round To -Infinity)** —この関数は「切り下げ整数化 (Round Toward -Infinity)」という名前に変更されました。(英語版のみに適用)

信号処理 VI

「遷移計測 (Transition Measurements)」VI の**プレシュート**出力は**遷移前**に変更されました。また、この出力は 64 ビット倍精度小数点数データタイプからクラスタデータタイプに変更されました。**オーバーシュート**出力は**遷移後**に変更されました。また、この出力は 64 ビット倍精度小数点数データタイプからクラスタデータタイプに変更されました。

双曲線関数

LabVIEW 8.5 以降には、以下の双曲線関数での変更点が追加されています。

- 入力値が関数の範囲外の実数値の場合、「Inverse Hyperbolic Cosine」関数は NaN を返します。
- 入力値が関数の範囲外の実数値の場合、「Inverse Hyperbolic Secant」関数は NaN を返します。

ライブラリ & 実行可能ファイル VI および関数

パスカル文字列ポインタを構成する場合、ライブラリ関数呼び出しノードでは、ブロックダイアグラムで値を文字列入力に配線する必要があります。C 文字列ポインタを構成する場合、値を文字列入力に配線するか、または**ライブラリ関数呼び出し**ダイアログボックスの**パラメータ**タブにある**最小サイズ**プルダウンメニューで文字列サイズを指定する必要があります。VI は、文字列の値を指定するまで実行できません。

「VI リファレンスを開く」関数

LabVIEW 8.0 および 8.2 では、**VI パス**入力からの VI の名前がそのターゲットのメモリにある VI 名と一致する場合、LabVIEW はメモリにある VI へのリファレンスを返して、**VI パス**入力で指定された VI はロードしません。LabVIEW 8.5 以降では、**VI パス**入力からの VI の名前がそのターゲットのメモリにある VI 名と一致していても、パスが異なる場合は「VI リファレンスを開く (Open VI Reference)」関数はエラーを返します。

64 ビットおよび倍精度数値データタイプをサポートする多態性 VI の端子

倍精度数値および 64 ビット整数タイプの両方をサポートする多態性 VI の端子に拡張精度数値データを配線した場合、LabVIEW は拡張精度数値データを倍精度数値データに強制変換します。この強制変換では、元のデータの分数のコンポーネントの一部が保持されます。

その他の VI および関数の動作における変更点

LabVIEW 8.5 以降には、以下の VI と関数の動作での変更点が追加されています。

- 「Instr Get Attribute」VI および「Instr Set Attribute」VI は、現在 LabVIEW と出荷されていません。これらの VI のどちらかをアプリケーションで使用する場合、同じ機能が使用できるように、これらの VI を **VISA 上級**パレットのプロパティノードで置換します。
- 「再帰ファイルリスト (Recursive File List)」VI の**全フォルダ**パラメータにはフォルダショートカットを含めることができますが、VI はそれらを再帰呼び出ししません。

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 8.5 以降で変更されました。

- 制御器クラスのデータバインディングパスプロパティは、VI が実行中に読み書きと設定の操作を行うことができます。このプロパティを書き込むには、書き込みを開始する前に NI Publish-Subscribe-Protocol (NI パブリッシュ - サブスクライブプロトコル) URL に制御器を連結する必要があります。
- アプリケーションクラスのターゲット: CPU プロパティには、AMD/Intel x64 の値が含まれます。
- アプリケーションクラスのターゲット: オペレーションシステムプロパティには、Windows x64 および Linux x64 という値が含まれます。
- ツリーの列ヘッダ内にポイントを配線すると、ツリー制御器クラスのポイントから行列メソッドは TREE_COLUMN_HEADERS というタグを返します。
- LabVIEW クラス: 作成メソッドには、名前入力が含まれています。**名前**入力を配線しない場合、LabVIEW は実行時にクラスの名前を指定するようにユーザにプロンプトします。
- 制御器の値: 取得 (バリエーション)、制御器の値: 取得 (平坦化)、制御器の値: 設定 (バリエーション)、制御器の値: 設定 (平坦化) メソッドでは、制御器の検索時に先行または末尾のスペースが削除されなくなりました。

廃止されたプロパティ、メソッド、イベント

LabVIEW 8.5 以降では、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。

- LV クラスライブラリクラスのデフォルトインスタンスプロパティ。「LV クラスデフォルト値を取得 (Get LV Class Default Value)」VI を代用してください。
- シーンオブジェクトクラスのジオメトリプロパティ。描画領域プロパティを代用してください。
- グラフチャートクラスの補助カラープロパティ。グラフスケールクラスのグリッドカラープロパティを代用してください。
- グラフチャートクラスのグリッドカラー: X カラープロパティ。グラフスケールクラスのグリッドカラー: 主カラープロパティとグリッドカラー: 補助カラープロパティを代用してください。
- グラフチャートクラスのグリッドカラー: Y カラープロパティ。グラフスケールクラスのグリッドカラー: 主カラープロパティとグリッドカラー: 補助カラープロパティを代用してください。
- 波形チャートクラスの凡例: プロット表示プロパティ。凡例: 行数プロパティを代用してください。
- 波形グラフクラスの凡例: プロット表示プロパティ。凡例: 行数プロパティを代用してください。
- リストボックスクラスのピクセル幅プロパティ。境界: 領域の幅プロパティを代用してください。
- ピクチャクラスのスクロールバー表示プロパティ。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。
- シーンオブジェクトクラスのジオメトリを設定メソッド。描画領域を設定メソッドを代用してください。
- アプリケーションクラスのシーン: ジオメトリ: 新規メッシュメソッド。シーン: 描画領域: ジオメトリ: 新規メッシュメソッドを代用してください。

- 制御器クラスのドラッグ開始イベント。適切な制御器クラスのドラッグ開始イベントを代用してください。
- 制御器クラスのドラッグ開始？イベント。適切な制御器クラスのドラッグ開始？イベントを代用してください。

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 8.5 以降で変更されました。

クラス	LabVIEW 8.2 での名前	LabVIEW 8.5 での名前	タイプ
AbsTime、数値	データ範囲	データエントリ制限	プロパティ
AbsTime、数値	データ範囲：増分	データエントリ制限：増分	プロパティ
AbsTime、数値	データ範囲：最大	データエントリ制限：最大	プロパティ
AbsTime、数値	データ範囲：最小	データエントリ制限：最小	プロパティ
AbsTime、数値	範囲外の動作	範囲外の値への対応	プロパティ
AbsTime、数値	範囲外の場合の動作：増分	範囲外の値への対応：増分	プロパティ
AbsTime、数値	範囲外の場合の動作：最大	範囲外の値への対応：最大	プロパティ
AbsTime、数値	範囲外の場合の動作：最小	範囲外の値への対応：最小	プロパティ
アプリケーション	ライブラリ：プロジェクト ライブラリファイルバージョンを取得	ライブラリ：ファイル LabVIEW バージョンを取得	メソッド
アプリケーション	シーン：ジオメトリ： 新規の立方体	シーン：描画領域：ジオメトリ： 新規ボックス	メソッド
アプリケーション	シーン：ジオメトリ： 新規の円錐	シーン：描画領域：ジオメトリ： 新規の円錐	メソッド
アプリケーション	シーン：ジオメトリ： 新規の円柱	シーン：描画領域：ジオメトリ： 新規の円柱	メソッド
アプリケーション	シーン：ジオメトリ： 新規ハイトフィールド	シーン：描画領域：ジオメトリ： 新規ハイトフィールド	メソッド
アプリケーション	シーン：ジオメトリ： 新規メッシュ	シーン：描画領域：ジオメトリ： 新規メッシュ	メソッド
アプリケーション	シーン：ジオメトリ： 新規の球	シーン：描画領域：ジオメトリ： 新規の球	メソッド
アプリケーション (ActiveX)	ライブラリ：プロジェクト ライブラリファイルバージョンを取得	ライブラリ：ファイル LabVIEW バージョンを取得	メソッド
デジタル、数値テキスト、スケール	形式と精度	表示形式	プロパティ
デジタル、数値テキスト、スケール	形式と精度：形式	表示形式：形式	プロパティ

クラス	LabVIEW 8.2 での名前	LabVIEW 8.5 での名前	タイプ
デジタル、数値テキスト、スケール	形式と精度：精度	表示形式：精度	プロパティ
デジタル表	列ヘッダ表示	信号番号表示	プロパティ
デジタル表	行ヘッダ表示	遷移表示可能	プロパティ
シーングラフ表示およびシーンウィンドウ	色をクリア	背景色	プロパティ
シーンオブジェクト	ジオメトリを設定	描画領域を設定	メソッド
VI	コネクタペーン	コネクタペーン：設定	プロパティ

LabVIEW MathScript の動作変更 (Windows、ベースパッケージにはありません)

LabVIEW 8.5 以降では、以下の LabVIEW MathScript が変更されました。

- MathScript 関数を使用して検索パスリストまたは作業ディレクトリに加えられた変更は、関数を呼び出す操作を行う **LabVIEW MathScript ウィンドウ** または MathScript ノードの現在のインスタンスにのみ適用されます。
 - addpath
 - cd
 - path
 - rmpath

LabVIEW MathScript ウィンドウ を閉じたり、MathScript ノードを含む VI の実行を停止すると、検索パスリストおよび作業ディレクトリはデフォルト値にリセットされません。

- qz 関数の構文は、 $[q, z, \alpha, \beta, \text{evec}] = \text{qz}(a, b)$ から $[S, T, Q, Z, R, L] = \text{qz}(A, B, \text{type})$ に変更されました。

LabVIEW クラスアイコン

LabVIEW 8.2 で LabVIEW クラスアイコンを作成して、ブロックダイアグラムにクラス制御器または表示器を配置した際アイコンを表示させたい場合、クラスマスクによってクラスアイコンの一部が隠されないように、アイコンが表示される領域を小さくするようにクラスアイコンを更新する必要があります。幅 32 ピクセル x 高さ 19 ピクセル未満の画像を使用してください。

LabVIEW で LLB を開く

オプション ダイアログボックスの **環境** ページの **LLB ファイルに Windows エクスプローラを有効** オプションは削除されました。LabVIEW は、**LLB マネージャ** ウィンドウで LLB を開きます。LLB を開く詳細は、ナショナルインスツルメンツのウェブサイト ni.com/jp/info にアクセスし、info code として exvfc5 と入力してください。

タイミングループの優先度に関する制限

LabVIEW 8.2.x 以前では、タイミングループの優先度に対して 2^32 まで選択できます。LabVIEW 8.5 以降では、65,535 未満の優先度のみサポートしています。

波形データタイプ

波形配列の範囲外に指標付けすると、結果の波形は 0 に等しい dt 値を含む不適切な波形ではなく、1 に等しい dt 値を含む適切な波形となります。また、スカラ出力トンネルを含む For ループを 1 度も実行しない場合にも同じ結果となります。

LabVIEW 8.0 からアップグレードする

LabVIEW 8.0 から LabVIEW 8.6 にアップグレードする際、以下の互換性の問題が発生する場合があります。起こり得るその他のアップグレードの問題の詳細は、このドキュメントの「[LabVIEW 8.2 からアップグレードする](#)」および「[LabVIEW 8.5 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の詳細は、LabVIEW のバージョン 8.0 と 8.5 間の各バージョンの『LabVIEW アップグレードノート』(ni.com/jp/manuals) を参照してください。

サポートされているプラットフォーム

LabVIEW 8.2 以降には、サポートされているプラットフォームで以下の変更点が含まれています。

- LabVIEW 8.2 以降では、Windows XP x64 はサポートされていません。
- LabVIEW 8.2 以降では、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 コンピュータの部分的なサポートを提供しています。Macintosh のサポートの詳細は、ナショナルインスツルメンツのウェブサイト ni.com/jp/info にアクセスし、info code として macintel と入力してください。LabVIEW 8.5 では、Intel および PowerPC プロセッサを搭載した Macintosh コンピュータがサポートされています。

システム要件

(Windows) LabVIEW 8.2 では、LabVIEW のインストールには最低 1.2 GB のディスク容量が必要です。LabVIEW 8.6 では、LabVIEW のインストールには最低 1.6 GB のディスク容量が必要です。

(Mac OS) LabVIEW 8.2 の最小インストールには 500 MB 以上のディスク容量、完全インストールには 700 MB 以上のディスク容量が必要です。LabVIEW 8.6 の最小インストールには 262 MB 以上のディスク容量、完全インストールには 828 MB 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 430 MB 以上のディスク容量、完全インストールには 620 MB 以上のディスク容量が必要です。LabVIEW 8.6 の最小インストールには 365 MB 以上のディスク容量、完全インストールには 651 MB 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 で変更されていません。そのため、LabVIEW 8.2 での変更点がドキュメントの内容に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なため、更新された内容の詳細は、『LabVIEW ヘルプ』の**目次タブの基本機能**ブックを参照してください。

VI と関数の動作変更

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

アプリケーションインスタンス間でデータを交換する

LabVIEW 8.2 以降では、「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)」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。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**指標 - 低、指標 - 高、スケール** は必須の入力です。

「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 以降の両方が使用するコンピュータにインストールされている場合、`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 VI を停止するとエラーを返します。LabVIEW 8.2 以降でこのサンプルを使用するには、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應します。「スプレッドシートファイルから読み取る」VI には、以下の DBL、I64、文字列のインスタンスが含まれています。

「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**の入力でエラーを受信した場合は実行されませんでした。LabVIEW 8.2 以降では、「GPIB ステータス」関数は**エラー入力**の入力でエラーを受信した場合でも必ず実行されます。

「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**入力のデフォルトの動作は、リファレンス VI の見つからないサブ VI を検出するとロードするダイアログボックスが表示されます。LabVIEW で自動的に検索できなかった VI の検索は、ユーザにはプロンプト表示されません。0x20 の新規値は、ロードするダイアログボックスを表示しないように指定し、リファレンス VI の見つからないサブ VI を検索します。見つからないサブ VI を検索するようユーザにプロンプト表示するには、オプションの 0x10 値を使用します。

「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 の入力多項式係数のすべては、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 では、**サンプル**が 1 および**最後を除外?**が TRUE の場合、VI は**開始**の 1 つの要素を含む配列をエラーなしで返します。LabVIEW 8.0 では、上記の状態において VI はエラーを返しました。

「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。この問題によって、「シンプルレジストリ値読み取り (Read Registry Value Simple)」VI に対してこのデータタイプを処理するようにパーサを作成することが必要でした。LabVIEW 8.2 以降では、「シンプルレジストリ値読み取り」VI は「シンプルレジストリ値書き込み (Write Registry Value Simple)」VI で使用されるデータタイプと同じ形式でこのデータタイプを返します。現在独自のパーサを追加する必要はありません。LabVIEW 8.2 以降でこれらの VI とパーサを使用すると、「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 **閉区間?** 入力のデフォルト値は、閉区間が選択されるように、TRUE から FALSE に変更されました。必要に応じて既存のコードを更新しない場合、VI は予想する結果を返さ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 は、データ出力の t0 コンポーネントは最初に読み取られたサンプルのタイムスタンプを返します。LabVIEW は最初のサンプル読み取りの初期時間を概算します。

連続的なサウンドタスクで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 は **連続サンプルモード** と **有限サンプルモード** で動作します。

波形 VI

LabVIEW 8.2 以降では、以下の波形 VI での変更点が含まれています。

- 「**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**入力は**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含まれます。現在、開始 / 遅延形式入力には、**絶対時間**オプションは含まれていません。開始入力は開始サンプル / 時間、実際の開始出力は実際の開始サンプル / 時間に変更されました。
- 「**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—この VI は、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含む多態性 VI です。
- 「**ファイルから波形を読み取り (Read Waveform from File)**」VI—エラーが EOF—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、実際の開始出力は実際の開始サンプル / 時間に変更されました。

- 「**デジタルパターン検索 (Search for Digital Pattern)**」 VI—開始入力は開始指標 / 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は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は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は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 以降で変更されました。

- ActiveX GetVIReference メソッドの**オプション**入力のデフォルトの動作は、リファレンス VI の見つからないサブ VI を検出するようにユーザにプロンプトします。0x20 の新規値は、**検索**ダイアログボックスを表示しないか、またはリファレンス VI の見つからないサブ VI を検出するようにユーザにプロンプトします。
- 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、メソッドがエラーを返す場合、LabVIEW はリファレンスを破棄しません。
- アプリケーション: 言語プロパティの有効な値には、簡体中国語が LabVIEW 環境の言語であること示すように zh-cn が追加されています。
- LabVIEW 8.0 では、配列データタイプをリファレンスによって渡す .NET メソッドはすべてのデータを refnum データタイプとして渡します。LabVIEW 8.2 以降では、配列データタイプをリファレンスによって渡す .NET メソッドはデータを実際のデータタイプとして渡します。
- 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、ユーザが制御器のテキストを編集していないことを示すように、(-2, -2) の値を返します。リストボックスクラスの行を編集プロパティは、ユーザが制御器のテキストを編集していないことを示すように -2 の値を返します。
- 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保留にしませんでした。LabVIEW 8.2 以降では、パネルアップデートを延期プロパティはサブパネルと動作します。
- 「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？」イベントを置換しました。LabVIEW プロジェクト以外で実行中の VI で「アプリケーションインスタンスを閉じる」イベントを使用する場合、ユーザインタフェースまたはプログラムを介して LabVIEW を終了すると、LabVIEW はイベントを生成します。「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を介して終了すると生成されます。LabVIEW プロジェクトで実行中の VI に対して「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？」イベントを登録する場合、アプリケーションインスタンスが閉じられるか、LabVIEW を終了すると、LabVIEW はイベントを生成します。

廃止されたプロパティ、メソッド、イベント

LabVIEW 8.2 以降では、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 以降では、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。変数クラスのデータタイプ (バリエーション) プロパティを代用してください。

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 以降で変更されました。

クラス	LabVIEW 8.0 での名前	LabVIEW 8.2 以降での名前	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、ミックスドシングルグラフ、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
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

クラス	LabVIEW 8.0 での名前	LabVIEW 8.2 以降での名前	タイプ
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

アプリケーションビルダでの変更点

LabVIEW 8.2.以降では、.libの拡張子を付けるようにアプリケーションまたは共有ライブラリの名前を変更することで、スタンドアプリケーション (EXE) または共有ライブラリ (DLL) の内容を参照することはできません。また、アプリケーションまたは共有ライブラリの外から VI へのパスを指定することで、スタンドアプリケーションまたは共有ライブラリの VI にアクセスすることもできません。アプリケーションと共有ライブラリを表示およびアクセスする詳細は、ナショナルインスツルメンツのウェブサイト ni.com/jp/info にアクセスして、info code として macintel と入力してください。

LabVIEW 7.xからアップグレードする

LabVIEW 7.xからLabVIEW 8.6にアップグレードする際、以下の互換性の問題が発生する場合があります。起こり得るその他のアップグレードの問題の詳細は、このドキュメントの「[LabVIEW 8.0からアップグレードする](#)」、「[LabVIEW 8.2からアップグレードする](#)」、「[LabVIEW 8.5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の詳細は、LabVIEW のバージョン 7.xから 8.5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 基本機能』マニュアルは、LabVIEW 8.6用に変更がありません。これらのドキュメントの PDF 版は、`labview\manuals` ディレクトリで参照できます。『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なため、更新された内容の詳細は、『LabVIEW ヘルプ』の**目次**タブの**基本機能**ブック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には、サポートされているプラットフォームで以下の変更点が含まれています。

- LabVIEW 7.1 以降では、Windows Me/98/95 はサポートされていません。LabVIEW 8.xでは、Windows NT はサポートされていません。
- LabVIEW 8.xでは、Mac OS X 10.2 以前のバージョンはサポートされていません。
- LabVIEW 8.xでは、Sun Solaris はサポートされていません。

システム要件

LabVIEW 7.x の RAM の最小要件は 128 MB ですが、ナショナルインストゥルメンツでは 256 MB を推奨します。LabVIEW 8.5 では、最低 256 MB 以上の RAM が必要とされていますが、ナショナルインストゥルメンツでは 1 GB 以上の 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、ナショナルインストゥルメンツでは 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須です。

ウィンドウ

LabVIEW 7.x では、Pentium III 以降または Celeron 600 MHz 以降のプロセッサが必要ですが、ナショナルインストゥルメンツでは Pentium 4 以降のプロセッサを推奨します。LabVIEW 8.x では、Pentium III または Celeron 866 MHz 以上のプロセッサが必要ですが、ナショナルインストゥルメンツでは 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 130 MB のディスク容量、完全インストールには 550 MB のディスク容量が必要です。LabVIEW 8.6 でインストールを完了するには 1.6 GB のディスク容量が必要です。

Mac OS

LabVIEW 7.x では、LabVIEW の最小インストールには最低 280 MB のディスク容量、完全インストールには 350 MB のディスク容量が必要です。LabVIEW 8.6 の最小インストールには 262 MB 以上のディスク容量、完全インストールには 828 MB 以上のディスク容量が必要です。

Linux

LabVIEW 7.x では、Pentium III 以降または Celeron 600 MHz 以降のプロセッサが必要ですが、ナショナルインストゥルメンツでは Pentium 4 以降のプロセッサを推奨します。LabVIEW 8.x では、Pentium III または Celeron 866 MHz 以上のプロセッサが必要ですが、ナショナルインストゥルメンツでは 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 200 MB のディスク容量、完全インストールには 300 MB のディスク容量が必要です。LabVIEW 8.6 の最小インストールには 365 MB 以上のディスク容量、完全インストールには 651 MB 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、ナショナルインストゥルメンツは 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 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。パレットセットは、カスタムパレットセットを使用せずに編集できます。LabVIEW 8.0 でのパレットの変更点の詳細は、ナショナルインストゥルメンツのウェブサイト ni.com/jp/info にアクセスして、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 および関数が LabVIEW 7.1 または 8.0 で変更されました。

.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。

最後に LabVIEW 7.x で保存された .NET VI をロードすると、アセンブリファイルがこの VI と同じディレクトリに存在する場合でも、または LabVIEW 7.x で **ツール→上級→.NET アセンブリリファレンス** を選択してアセンブリファイルを登録した場合、LabVIEW 8.x は VI が参照するアセンブリを検索するようにプロンプトする場合があります。

解析 VI のアルゴリズム

LabVIEW 7.1 以降では、解析 VI は BLAS/LAPACK アルゴリズムを使用します。これらの VI は、より正確な結果を生成するようになりました。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** が空または未配線で、単一信号または連結信号を **入力信号 B** に配線する場合、**付加信号** 出力は空になります。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、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** に配線する場合は、連結信号の各信号は次の信号を付加して、結果として 1 つの信号を生成します。

比較関数

LabVIEW 7.x 以前では、比較関数を使用してバリエーションデータを比較すると、LabVIEW は最初に 2 つのバリエーションの長さを比較して、次にバリエーションをビットごとに比較します。LabVIEW 8.x では、バリエーションの実際のタイプ情報をエンコードするタイプコードとバリエーションデータの比較を開始して、次に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、「内積 (Dot Product)」 VI は以下の式を使用して、 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 は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 のコネクタペーンは変更されました。LabVIEW 8.x では、「Easy Text Report」 VI を使用する、最後に LabVIEW 7.x 以前のバージョンで保存された VI を開く際、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、%b、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)」関数と使用すると、その入力を文字列に変換する前に浮動小数点入力が32ビット整数に丸め込まれます。

LabVIEW 8.xでは、これらの形式指定子の構文要素は、この関数が入力を文字列に変換する前に浮動小数点入力を64ビット整数に丸め込む原因となります。

「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ビット整数に強制変換して、1つの32ビット整数を作成します。LabVIEW 8.xでは、「数値結合」関数は32ビット整数入力を結合して、1つの64ビット整数を作成します。



メモ LabVIEW 7.xのVIをLabVIEW 8.xで開いた場合、LabVIEWは32ビット整数入力を16ビット整数に強制変換します。

数学 VI と行列

LabVIEW 8.xでは、数学 VI は行列データタイプをサポートしています。LabVIEW 7.xで作成されたVIをLabVIEW 8.xでロードし、そのVIに行列データタイプを使用できる関数に配線された数学VIがあり、2D配列を代用している場合、赤い7.xグリフがその関数上に表示されます。赤いグリフは、LabVIEWによって2D配列が行列データタイプで置換されたことを示します。

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 (Number To Hexadecimal String)」、「数値を8進数文字列に変換 (Number To Octal String)」、「数値を10進数文字列に変換 (Number To Decimal String)」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ビット整数に丸め込みます。

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ビット整数に丸め込みます。ただし、LabVIEW 8.xでLabVIEW 7.xのVIを開く場合は、LabVIEWは浮動小数点入力を32ビット整数に丸め込むことで、互換性と機能性を維持します。

「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、同じ名前前のVIがメモリに存在する場合、メモリ内のVIへのパスが指定したパスと一致しない場合でも、LabVIEWはメモリ内のVIへのリファレンスを返します。

LabVIEW 8.xでは、「VI リファレンスを開く」関数のVIパス入力が文字列の場合、VIパスがそのターゲット上のメモリにあるVIのVIの正規名に一致する場合のみ、LabVIEWはVIを開きます。VIパスがパスの場合、LabVIEWは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。LabVIEWが一致するパスでVIを検出しない場合、LabVIEWは指定されたパスでディスクからVIをロードしようとします。LabVIEW 8.5以降では、LabVIEWがファイルを検出できない場合、またはファイルのVI名がそのターゲット上のメモリにある別のVIと同じVIの正規名である場合、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 $\max|X|$ を0および $Y(i)=X(i)/\max|X|$ として返すか、または $Yij=Xij/\max|X|$ をNaNの配列として返します。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

LabVIEW 7.x以前では、「キーを読み取る (Read Key)」VIの文字列インスタンスを使用して、Shift-JISでエンコードされた日本語のマルチバイト文字を読み取ることができます。

1 または <Shift-JIS> を **マルチバイトエンコーディング** 入力に配線する必要があります。LabVIEW 8.xでは、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場合、「キーを読み取る」VIの文字列インスタンスはデフォルトでエンコードされたマルチバイト文字を読み取ります。

スケールVI

LabVIEW 7.1以前では、「スケール 1D (Scale 1D)」VIまたは「スケール 2D (Scale 2D)」VIの **X** 入力ゼロの配列の場合、このVIは **スケール** を 0、**オフセット** を 0、**Y=(X- オフセット)/ スケール** を NaN の配列として返します。LabVIEW 8.xでは、スケールVIの **X** 入力ゼロの配列の場合、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はエラー入力の入力がエラーを受信すると実行を試みません。LabVIEW 8.xでは、これらのVIはエラー入力の入力でエラーを受信した場合でも実行を試みます。ただし、LabVIEW 7.xのVIをLabVIEW 8.xで開くと、LabVIEWはLabVIEW 7.xの機能を維持するためにVIをアップデートします。

SMTP E メールVI

LabVIEW 7.x以前では、SMTP E メールVIの文字セット入力に値を書き込むことで、文字セットを指定できます。LabVIEW 8.xでは、SMTP 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ことを想定します。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には、現在文字セットまたは字訳パラメータは含まれていません。

「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、LabVIEWは同じ振幅の要素のシーケンスを変更しません。LabVIEW 8.xでは、メソッドを **振幅** に設定すると、LabVIEWは最初に同じ振幅の要素を実部に対してソートして、次に虚部に対してソートします。

「単位ベクトル」VI

LabVIEW 7.x以前では、「単位ベクトル (Unit Vector)」VIは、以下の公式を使用して入力ベクトルのノルムを計算します。

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

LabVIEW 8.xでは、「単位ベクトル」VIは、以下の公式を使用して入力ベクトルのノルムを計算します。

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

ここで、 X は入力ベクトル、 $\|X\|$ はノルム、 γ 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、それぞれヘルプ、ツール、およびファイルメニューに表示されます。LabVIEW 8.x 以降でこれらのディレクトリに配置する VI は、LabVIEW 8.0 以降ではこれらの VI はプライベートアプリケーションインスタンスで開かれるため、LabVIEW 8.x で正常に動作しない場合があります。

labview¥vi.lib¥Utility¥allVIsInMemory.llb にある VIMemory Get VIs in Memory VI を使用して、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 のリストを生成します。labview¥vi.lib¥ Utility¥allVIsInMemory.llb にある Get User Application Reference VI を使用して、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します。アプリケーションインスタンスの詳細は、『LabVIEW ヘルプ』を参照してください。

廃止された VI および関数

LabVIEW 8.x では、以下の VI と関数はサポートされていません。

- LabVIEW 7. 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)」関数は**関数**パレットにありません。「シンプルエラー処理 (Simple Error Handler)」VI または「一般エラー処理 (General Error Handler)」VI を代用してください。
- LabVIEW 8.x は、「カイ 2 乗分布 (Chi Square Distribution)」、「F 分布 (F Distribution)」、「正規分布 (Normal Distribution)」、「T 分布 (T Distribution)」VI をサポートしていません。各 VI に対して、「連続 CDF (Continuous CDF)」VI のカイ 2 乗、F、正規、ステューデント t インスタンスを代用してください。
- 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)」、「逆 T 分布 (Inv T Distribution)」VI をサポートしていません。各 VI に対して、「連続 CDF」VI のカイ 2 乗、F、正規、ステューデント t インスタンスを代用してください。
- LabVIEW 8.x では、「1D 線形評価 (1D Linear Evaluation)」VI と「2D 線形評価 (2D Linear Evaluation)」VI は**関数**パレットにありません。「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 と「2D 多項式評価 (2D Polynomial Evaluation)」VI は、**関数**パレットにありません。「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 と「1D 極座標から直交座標へ変換 (1D Polar to Rectangular)」VI は**関数**パレットにありません。「直交座標を極座標に変換 (Re/Im To Polar)」関数と「極座標を直交座標に変換 (Polar To Re/Im)」関数を代用してください。

- LabVIEW 8.xでは、「高調波アナライザ (Harmonic Analyzer)」VIは**関数**パレットにありません。「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、**THD** または **コンポーネントレベル** 出力を測定します。
- LabVIEW 8.xでは、「ネットワーク関数 (平均) (Network Functions (avg))」VIは**関数**パレットにありません。「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は**関数**パレットにありません。「遷移計測 (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** (「遷移計測」VIの**遷移後**出力に相当)、または**プレシュート** (「遷移計測」VIの**遷移前**出力に相当する) 出力、「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、あるいは「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)」関数は**関数**パレットにありません。「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は**関数**パレットにありません。「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 リソース名定数とIVI 論理名定数は**関数**パレットにありません。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、エラーリング定数は**関数**パレットにありません。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、LabVIEW 7.xの**サウンド**パレットから利用可能なサウンドVIは、**関数**パレットにありません。LabVIEW 8.xでは、サウンドVIを代用してください。LabVIEW 7.xで出荷されるサンプルは、LabVIEW 8.xと出荷されません。

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は**関数**パレットにありません。「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は**関数**パレットにありません。「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ています。

- 「ファイルサイズを取得 (Get File Size)」関数を使用して、ファイルのサイズを決定します。
- 「ファイルダイアログ (File Dialog)」Express VIを使用して、ファイルダイアログボックスの開始パス、ファイルパターン、ファイルまたはディレクトリのデフォルト名を指定します。

- 「Refnum をパスに変換 (Refnum to Path)」関数を使用して、リファレンスをパスに変換します。
- 「バイナリファイルに書き込む (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)」関数は**関数**パレットにありません。「バイナリファイルから読み取る (Read from Binary File)」および「バイナリファイルに書き込む (Write to Binary File)」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 は**関数**パレットにありません。「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**パレットにありません。「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**パレットにありません。「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**パレットにありません。「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**パレットにありません。「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」関数の場合と同様に、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)」関数は**関数**パレットにありません。「ファイル位置を取得 (Get File Position)」と「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)」関数は**関数**パレットにありません。「タイプと作成者を取得 (Get Type and Creator)」と「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)」関数は**関数**パレットにありません。「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)」関数は**関数**パレットにありません。「ファイルから一行読み取り (Read Lines From File)」VI は**関数**パレットにありませんが、互換性の理由により LabVIEW と出荷されます。

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**パレットにありません。「バイナリファイルから読み取る (Read from Binary File)」と「バイナリファイルに書き込む (Write to Binary File)」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とメソッドの動作は、それらのプロパティとメソッドが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」の名前に変更されました。LabVIEW 7.1には、フロントパネルが既に開いている場合、エラーを返さない異なる「FPを開く」メソッドが含まれています。LabVIEW 7.1の「FPを開く」メソッドは、LabVIEW 8.xで「フロントパネル:開く」の名前に変更されました。LabVIEW 7.0の「FPを開く」メソッドを使用するVIがある場合、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に設定する場合、LabVIEWはVIの実行が停止された後リファレンスを自動的に破棄します。VIを実行メソッドがエラーを生成すると、リファレンスは自動的に閉じられません。LabVIEW 8.0では、メソッドがエラーを返すと、LabVIEWはリファレンスを自動的に破棄します。この動作は、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、VIが壊れる原因となる場合があります。LabVIEW 8.2以降では、この動作はLabVIEW 7.1での動作に戻りました。

キーダウンイベントとキー繰り返しイベント

現在、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>キーと数値キーパッドの<Enter>キーに対して異なる値が含まれます。LabVIEW 7.x以前では、<Enter>キーまたは<Return>キーがこれらのイベントの1つを生成すると、LabVIEWはVKeyデータフィールドに<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データフィールドにそれぞれ<Enter>または<Return>を返します。

(Mac OS) LabVIEW 8.xは、ショートカットメニューに対して<Control>-クリックのみを受け入れ、<Command>-クリックのキーの組み合わせは受け入れません。イベントストラクチャで動作をエミュレートする場合、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下の項目より下の行に設定する場合、LabVIEWはその行を最後に表示されている項目に固定します。LabVIEW 8.xでは、リストボックスの表示されている項目数によって、このプロパティに配線できる行数は制限されません。

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 は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によって、VI はメモリ内に維持されません。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。「VI リファレンスを開く (Open VI Reference)」関数を使用して、リファレンスを明示的に閉じるまでメモリに残る VI へのリファレンスを取得します。

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フロントパネルオブジェクトと同じテキスト表示で文字列を返します。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；高さプロパティとアクティブセルプロパティ：セルサイズ：幅プロパティは、ツリー制御器内の各行に LabVIEW 8.xよりも 1 ピクセル少ない値を返します。たとえば、それぞれ 70 ピクセルと 16 ピクセルとしてツリー制御器の高さと幅を返すプロパティノードを含む LabVIEW 7.x の VI を LabVIEW 8.x でロードする場合、高さと幅を決定するのに配置する新規プロパティノードのすべては 69 ピクセルと 15 ピクセルを返します。

VI 文字列メソッド

VI 文字列：インポートメソッドを使用する場合、エクスポート VI 文字列メソッドを使用して前回のバージョンの LabVIEW からエクスポートする文字列は、LabVIEW 8.x でプロパティをインポートしない場合があります。

廃止されたプロパティ、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、複数列リストボックス、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対して、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を代用してください。

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対して、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。それぞれテキストカラー、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。LabVIEW 8.xでこのプロパティを使用する場合、プロパティは左上隅のペーンのみにも適用されます。ペーンカラープロパティをペーンクラスを代用してください。

サブパネルプロパティ

LabVIEW 8.xでは、サブパネルのサブVIのペーンを使用して、サブパネル制御器に対してスクロールバーの可視性の構成とサブパネル制御器のフロントパネルのスケールを行います。

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。LabVIEW 8.xでこのプロパティを使用する場合、プロパティは左上隅のペーンのみにも適用されます。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。LabVIEW 8.xでこのプロパティを使用する場合、プロパティは左上隅のペーンのみにも適用されます。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も適用されます。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では、**イベントを編集**ダイアログボックスのVIクラスのマウスダウン、マウスダウン?、マウス移動、またはマウスアップイベントは選択できません。ペーンクラスのマウスダウン、マウスダウン?、マウス移動、マウスアップイベントを代用してください。

旧バージョンのアプリケーションビルダからの変更点

アプリケーションビルダは**プロジェクトエクスプローラ**ウィンドウに統合されます。

LabVIEW ベースパッケージまたは開発システムをご使用の場合は、ナショナルインスツルメンツのウェブサイト ni.com/jp/info で Info Code に rd1v21 と入力する方法で、アプリケーションビルダを別途購入していただけます。

プロジェクトエクスプローラウィンドウで**ビルド仕様**を使用して、ビルド仕様を作成したり、スタンドアロンアプリケーション (EXE)、共有ライブラリ (DLL)、ZIP ファイルを作成します。**(Windows) ビルド仕様**でビルド仕様を作成してインストーラを作成することもできます。ビルド仕様は旧バージョンのアプリケーションビルダの .bld ファイルと同等ですが、このファイルは現在別のファイルではなく LabVIEW プロジェクトの一部です。



メモ アプリケーションビルダはプロジェクト内で使用する必要があります。

.bld ファイルは、新規プロジェクト内でビルド仕様に変換できます。**ツール→ビルドスクリプトを交換**を選択して、変換する .bld ファイルを参照して選択します。LabVIEW はこのファイルを使用して、ソースファイルとビルド仕様を含むプロジェクトを作成します。

アプリケーション項目のタグ

以下のリストは、該当する機能が廃止されたか、別の機能と統合されたために LabVIEW から削除されたアプリケーション項目タグを示します。

- APP_SAVE_WITH_OPTIONS
- APP_UPDATE_VXI
- APP_DSC_TOOLBAR
- APP_DSC_TAGEDITOR
- APP_DSC_TAGMONITOR
- APP_DSC_HTV
- APP_DSC_ENGINE
- APP_DSC_SECURITY
- APP_DSC_LOGOUT
- APP_DSC_CPWD
- APP_DSC_USERINFO
- APP_DSC_USEREDITOR
- APP_DSC_ADVANCED
- APP_DSC_STARTUP
- APP_DSC_SRVBW
- APP_DSC_IST
- APP_DSC_IMAGENAV
- APP_DSC_OPTIONS
- APP_SRC_CODE_CTRL
- APP_BUILD_STANDALONE_APP
- APP_EDIT_VI_LIBRARY

- APP_DN_ASSEMBLY_REFS
- APP_SHOW_CLIPBOARD
- APP_VIEW_PRINTED_MANUALS
- APP_RT_ENGINE_INFO
- APP_SWITCH_EXEC_TARGET
- APP_REALTIME

旧バージョンの LabVIEW で保存されたランタイムメニュー（.rtm）ファイルを使用し、ファイルに削除されたタグが含まれている場合、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際、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

ナショナルインストルメンツは、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、それらを信号処理および数学 VI で置換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 に対して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のメモリ内にあるすべての項目に対する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、このセクションは明白でない項目のそれぞれに対し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

ツール→上級→文字列をインポートを選択するか、または VI 文字列：インポートメソッドを使用して、VI 文字列ファイルをインポートする際、LabVIEW 8.xは <GROUPER></GROUPER> の新規のタグのセットを検索します。このタグセットは、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、従来の LabVIEW のバージョンで保存された VI 文字列ファイルはインポートできません。

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がリストされます。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がリストされます。また、LabVIEW 7.1 以前では、LabVIEW がそのプライベートデータの <LBLABEL> セクションにリストする 1 つのフォントのみリストボックスに含むことができます。LabVIEW 8.xでは、LabVIEW がそのプライベートデータの <CELL_FONTS> セクションにリストする複数のフォントをリストボックスに含むことができます。

LabVIEW 7.1 以前では、複数リストボックスの文字列はそのデフォルトデータ内にリストされます。ただし、複数リストボックスのデフォルトデータは整数または整数の配列です。LabVIEW 8.xでは、複数リストボックスの文字列はプライベートデータ内にリストされます。

LabVIEW 7.1 以前では、ツリー制御器の文字列とフォントのいずれもエクスポートできません。LabVIEW 8.xでは、ツリー制御器とフォントの両方をエクスポートできる他、リストボックスと複数リストボックスと同じ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にプライベートデータまたはデフォルトデータに対して2つ以上のタグを含むことはできません。また、LabVIEW 8.xでは、ネストされたレベルごとに項目がインデントされます。

以下の手順に従って、VI 文字列ファイルを LabVIEW 8.x形式に変換します。

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. **ツール→上級→文字列をエクスポート**を選択して、LabVIEW 8.x形式で VI 文字列ファイルを保存します。

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は、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は、LabVIEW 7.xで書き込まれたタイプデスクリプタの読み取りと LabVIEW 7.xで読み取ることが可能なタイプデスクリプタを書き込む機能を提供します。「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が LabVIEW 7.xで書き込まれたかのように入力データを処理します。**7.x データの変換** ショートカットメニュー項目を選択して、データ文字列出力が配線されている場合、LabVIEW 8.xは関数上に赤い 7.x グリフを表示して、データを LabVIEW 7.x形式に変換中か LabVIEW 7.x形式から変換中であることを示します。データが変換されることを回避するには、再度 **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て、チェックマークをオフにします。

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、LabVIEW 8.xは「文字列に平坦化 (Flatten To String)」関数の **7.x データの変換** 属性を自動的に設定します。関数は、LabVIEW 7.x以前の場合と同様に動作し続けます。VI で LabVIEW 8.xのタイプデスクリプタ形式を使用する場合、「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、チェックボックスをオフにします。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中の VI でデータを送受信しない場合は、LabVIEW 8.xのタイプデスクリプタ形式を使用します。LabVIEW の将来のバージョンでは、以前のタイプデスクリプタ形式のサポートは中止される可能性があります。

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。従来の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、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の詳細は、ナショナルインスツルメンツのウェブサイト ni.com/jp/info にアクセスして、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保存されたりバージョン履歴が失われます。従来のバージョンのファイルは、新規プロバイダに転送できません。

以下の手順に従って、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します。

1. 従来のバージョンの LabVIEW では、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は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は、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は、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. 他社のソース管理プロバイダが使用できるように LabVIEW を構成します。
他社のソース管理プロバイダと動作するように LabVIEW を構成する詳細は、『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**ブック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→操作方法→LabVIEW プロジェクトを作成する**ブック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の最小値の -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の最大値の 32,767 となります。

ケースストラクチャに配線されている定数

LabVIEW 7.x 以前では、定数をケースストラクチャに配線して、実行されないケースにそのサブ VI を配置することで、メモリにサブ VI を維持することができます。たとえば、ケースストラクチャに TRUE 定数を配線して、サブ VI をケースストラクチャの FALSE ケースに配置すると、LabVIEW は呼び出し VI とともにサブ VI をロードします。LabVIEW 8.x は、実行されないコードをすべて削除します。したがって、ケースストラクチャに定数が配線されている従来のバージョンの LabVIEW で保存された VI を LabVIEW 8.x でロードする場合、LabVIEW は定数を非表示の制御器に変更して、従来のバージョンの LabVIEW の動作を維持します。

オペレーティングシステムのメッセージを遅延する

LabVIEW 7.x では、.LabVIEW は NET および ActiveX イベントを処理するためのコールバック VI を実行中、オペレーティングシステムのメッセージを処理します。LabVIEW 8.x では、LabVIEW はコールバック VI の実行が停止するか、ユーザがモーダルダイアログボックスをロードするまで、オペレーティングシステムのメッセージの処理を延期します。

この遅延によってコールバック VI が割り込みなしで実行されることが可能になり、デッドロック状態の原因となり得る、LabVIEW が別のイベント内でイベントを生成することが防止されます。

コールバック VI から非モーダルダイアログボックスに同期的呼び出しは行えません。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が 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オペレーティングシステムのメッセージを処理します。LabVIEW 8.x では、DLL 関数への呼び出しが終わるか、モーダルダイアログボックスを DLL からロードするまで、LabVIEW は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されることが可能になり、デッドロック状態の原因となり得る、LabVIEW が DLL 関数の実行中に同じ DLL の呼び出しを回避します。ユーザからのキーボードメッセージなどのオペレーティングシステムのメッセージの遅延は、共有ライブラリ関数の実行中に同じ共有ライブラリファイルの呼び出しを回避するために役立ちます。たとえば、ユーザがボタンを押したために共有ライブラリ関数が呼び出された場合、この関数が完了するまではユーザが再度同じボタンを押すことを禁止する必要があります。

このデフォルトの動作を使用する場合、DLL が実行中、非モーダルダイアログボックスに同期的呼び出しは行えません。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が 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。**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**を選択して、**カテゴリ**リストから**上級**を選択して、**共有ライブラリでオペレーティングシステムメッセージを遅延**チェックボックスをオフにして、DLL 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と書き込みの機能を使用できます (ドキュメント化されていません) LabVIEW 8.x では、これらのメソッドは存在しません。ドキュメント化されていないこれらの機能を活用するユーティリティは機能しません。また、そのため VI から Macintosh リソースの読み取りまたは書き込みは行えません。

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 はプログラマ的に中止できません。LabVIEW 8.x では、中断メソッドを使用して、これらのダイアログボックスを表示している VI をプログラマ的に中止できます。

プロパティおよびインポートノード

LabVIEW 7.x でカーソル凡例から暗示的にリンクされたプロパティノードまたはインポートノードを作成した場合、その VI を LabVIEW 8.x で VI を開く際、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 以前でのバージョンで labview.lib にリンクする共有ライブラリ (DLL) を作成した場合、LabVIEW 8.x は共有ライブラリを labviewv.lib にリンクします。labviewv.lib に共有ライブラリをリンクする詳細は、『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**オプションに余白値にセンチメートルが使用されます。LabVIEW 8.xの**余白**オプションには、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 8.6にアップグレードする際、以下の互換性の問題が発生する場合があります。起こり得るその他のアップグレードの問題の詳細は、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](#)」、「[LabVIEW 8.2からアップグレードする](#)」、「[LabVIEW 8.5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の詳細は、LabVIEWのバージョン6.xから8.6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、**10**コンポーネントに対して、波形のデータタイプは倍精度浮動小数点数ではなく、タイムスタンプのデータタイプを使用します。LabVIEW 6.xでデータタイプの情報を含めずに、波形データタイプとしてデータを保存する場合、LabVIEW 7.x以降でそのデータを回収しようとする、エラーが発生する場合があります。

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はファイルの古い波形データタイプ形式を新規の波形データ形式に変換します。このVIは、変換を受け入れるようにプロンプトす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はこの変換処理を実行することはできず、代わりにエラーを返します。LabVIEW 6.xからLabVIEW 7.x以降に波形データを移行する方法の詳細は、[ナショナルインスツルメンツのウェブサイト](#) ni.com/jp/info にアクセスして、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。VISA VIと関数を使用してシリアルデバイスと通信するVIを作成します。

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の通信を行うために `serpdrv` ドライバを使用しません。LabVIEWには、VISAに基づいた互換性があるVIが含まれます。新しいアプリケーションでは、VISAとシリアルVIおよび関数を使用して、シリアルデバイスを制御します。シリアルVIを含む、従来のバージョンのLabVIEWで作成されたVIのすべては、LabVIEW 7.1以降で動作し続けます。

ポート番号のポートへの割り当てを構成する場合、割り当てはそれらのポートに指定する必要があります。 `labview¥vi.lib¥Instr¥_sersup.11b`にある「シリアルエイリアス設定 (set serial alias ports)」VIを使用して、シリアルポートの割り当てを指定します。VIの**VISA エイリアス**入力に文字列配列を配線して、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ます。たとえば、VIエイリアスの `MySerialPort` に割り当てるようにポート0を構成する場合、**VISA エイリアス**入力配列の最初の要素として `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¥smplser1.11b` 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 を配線する場合、または自動指標付けが有効の状態では For ループに空の配列を入力として配線する場合、For ループはデフォルトデータを生成します。ループは実行されずに、自動指標付けが無効の状態の出カトンネルのすべてにトンネルのデータタイプに対してデフォルト値が含まれます。

リモートフロントパネルライセンス

LabVIEW 開発システムとアプリケーションビルダには、リモートでフロントパネルを表示および制御することを 1 つのクライアントに許可す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、フロントパネルを表示および制御することを 5 つのクライアントに許可す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して、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 以前のバージョンよりも多くの VI 実行用のスレッドを割り当てます。この変更により、呼び出す DLL が実際は再入可能でない場合、「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場合があります。「ライブラリ関数呼び出しノード」および再入可能の詳細は、『LabVIEW ヘルプ』を参照してください。

LabVIEW がスレッドを割り当てる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。また、**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して、VI の再入可能を無効にすることができます。

スレッド割り当ての詳細は、『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 計測器ドライブライブラリ CD は含まれていません。計測器ドライバは、ナショナルインスツルメンツ計測器ドライバネットワーク (ni.com/jp/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含まれています。

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、「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はありません。

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)』で記載されているガイドラインにより厳密に準拠しています。また、現在カロリーの単位はカロリー（熱）、馬力は馬力（電気）と表記されます。これらの単位の略語は変更されていません。以下の表は、LabVIEW 6.1 と 7.x 以降の単位変換ファクタの変更の詳細です。

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	16.5 ft	5.029210 m
スラグ	32.174 lb	14.59390 kg
統一された原子質量 (u)	1.66057e-27 kg	1.66053873e-27 kg

パネルアップデートを延期プロパティ

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、LabVIEW は保留中の変更があるフロントパネルオブジェクトを再描画することを待機します。LabVIEW 7.0 以降では、このプロパティを TRUE に設定すると、LabVIEW は保留中の変更があるフロントパネルオブジェクトを再描画して、次にフロントパネルを更新するためのすべての新規の要求を延期します。場合によっては、この変更は、LabVIEW がフロントパネルの変更された要素を描画するのに余分な時間がかかる原因となります場合があります。

数値制御器のデータ範囲

LabVIEW 6.1 以前では、一部の数値制御器には、デフォルトで最小値が 0.00、最大値が 0.00、増分値が 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、タイプ定義の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あることに気が付いたかもしれません。タイプ定義端子でない VI または関数の端子にタイプ定義を配線すると、強制ドットが表示されます。また、タイプ定義の出力端子をタイプ定義でない表示器に配線した場合も強制ドットは表示されます。これらの強制ドットは、VI でタイプ定義を一貫して使用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パフォーマンスに影響しません。

「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、ユーザが新規ファイル名を入力できる場合、「ファイルダイアログ (File Dialog)」関数が表示するファイルダイアログボックスに**保存**というボタンラベルが表示されます。そうでない場合、ボタンラベルは**開く**と表示されます。LabVIEW 8.xでは、ボタンラベルを変更しない場合を除いて、「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ラベルは**OK**と表示されます。「ファイルダイアログ」Express VI の**ボタンラベル**入力を使用して、ボタンラベルを変更します。既存の VI で「ファイルダイアログ」Express VI を使用する場合、VI の動作を再確認して、デフォルトラベルの**OK**は VI の機能に対して適切であることを確認します。

「オンラインヘルプを制御」関数

現在、「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**入力は必須の入力です。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入力に配線できます。コンパイル済みのヘルプファイルのみ配線した場合、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、LabVIEW が VI をロードし、VI サーバを使用して VI をロードする際、フロントパネルを表示するように VI を構成する場合、LabVIEW はフロントパネルを表示しません。フロントパネル: 開くメソッドを使用して、プログラマ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、LabVIEW はテンプレートから VI のインスタンスを作成します。テンプレートがメモリにある場合は、LabVIEW はテンプレートへのリファレンスを開きます。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すでにあるテンプレートへのリファレンスを作成する場合、オプションパラメータに 0x02 を指定しない限り、関数はエラーを返します。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場合、そのテンプレートがすでにメモリにある場合でも、LabVIEW はテンプレートから VI のインスタンスを作成します。

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、指数を表す演算子は ** (x**y など) です。^ 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 ファイル形式では現在すべての名前に対して大文字と小文字を区別が必要です。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、名前の文字の大文字と小文字が異なることなく、使用する名前は IVI Configuration Store ファイルで定義された名前と完全に一致する必要があります。

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 8.6 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/jp/info で info code として ext8h9 と入力して参照できます。このサイトは、旧バージョンの『LabVIEW アップグレードノート』の情報、および LabVIEW 8.6 で動作させるために行う VI の変換情報を提供しています。

LabVIEW 8.6 の機能および変更点

LabVIEW 8.6 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）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→LabVIEW ヘルプを検索**を選択します。

既知の問題、修正されたバグの部分的なリスト、その他の互換性に関する問題、LabVIEW 8.6 における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 をインストールする

(Windows) LabVIEW 8.6 では、LabVIEW をインストールして LabVIEW プラットフォーム DVD からモジュールとツールキットを選択できます。詳細については、このドキュメントの「**モジュール、ツールキット、計測器ドライバをアップグレードする**」のセクション、または『LabVIEW リリースノート』の「LabVIEW 8.6 をインストールする」セクションを参照してください。

LabVIEW のドキュメント

『LabVIEW クイックリファレンスカード』はキーボードショートカット、オンラインリソース、ドキュメントの使用に関するガイドが簡潔に記載されています。

新しいサンプル VI

NI サンプルファインダの**参照**タブで **LabVIEW 8.x** フォルダの新しいサンプルフォルダを参照し、LabVIEW 8.x に新しく追加されたサンプルの説明を表示してサンプルを起動します。

ブロックダイアグラムの改善点

LabVIEW 8.6 には、以下のブロックダイアグラムでの改善点とブロックダイアグラムに関連する改善点が含まれています。

Express VI を作成および編集する

ツール→上級→Express VI を作成 / 編集を選択して、**Express VI を作成 / 編集**ダイアログボックスを使用して Express VI を作成または編集します。Express VI は、既存の VI から作成するか、別の Express VI から作成するか、あるいは空の VI から作成することができます。LabVIEW の以前のバージョンでは、この機能は LabVIEW Express VI 開発ツールキットがインストールされている場合のみ使用できます。

LabVIEW で Express VI を作成する詳細については、『LabVIEW ヘルプ』の**目次**タブにある**基本機能→ブロックダイアグラムを作成する→Express VI** ブックを参照してください。

ブレイクポイントマネージャウィンドウを使用してブレイクポイントを管理する

ブロックダイアグラムで1つのオブジェクトを右クリックして、ショートカットメニューから**ブレイクポイント→ブレイクポイント設置**を選択してブレイクポイントを作成します。ブレイクポイントを無効にし、物理的にブレイクポイントを削除せずにブレイクポイントで VI が一時停止することを回避するには、ブレイクポイントがあるオブジェクトを右クリックして、ショートカットメニューから**ブレイクポイント→ブレイクポイントを無効にする**を選択します。無効なブレイクポイントがあるオブジェクトを右クリックし、ショートカットメニューから**ブレイクポイント→ブレイクポイントを有効にする**を選択すると、無効にしたブレイクポイントを有効にできます。ブレイクポイントを削除するには、オブジェクトを右クリックし、ショートカットメニューから**ブレイクポイント→ブレイクポイントを解除**を選択します。

ブレイクポイントマネージャウィンドウを使用して、VI でブレイクポイントを管理します。ブレイクポイントマネージャウィンドウを使用すると、VI 階層ごとまたは一度にブレイクポイントを無効、有効、または解除することができます。ブレイクポイントマネージャウィンドウでブレイクポイントをダブルクリックして、ブロックダイアグラムで特定のブレイクポイントの位置を検出します。

表示→ブレイクポイントマネージャを選択するか、またはオブジェクトを右クリックしてショートカットメニューから**ブレイクポイント→ブレイクポイントマネージャ**を選択して、**ブレイクポイントマネージャ**ウィンドウを表示します。ブレイクポイントを個々に選択するか、または**すべて選択**ボタンをクリックしてすべてのブレイクポイントを選択します。**有効**または**無効**ボタンをクリックして、すべての選択されたブレイクポイントを同時に有効または無効にします。

タイミングストラクチャでクローンを共有する再入可能 VI の事前の割り当てを許可する

タイミングループまたはタイミングシーケンスストラクチャの内側でクローンを共有する再入可能 VI を事前に割り当てるか、そのクローンを共有するように LabVIEW を構成できます。たとえば、タイミングループまたはタイミングストラクチャの内側にクローンを共有する再入可能 VI を配置する場合、ストラクチャの内側で呼び出す再入可能 VI のインスタンスに対してクローンを事前に割り当てるようにストラクチャを設定することができます。ストラクチャの外側で呼び出す再入可能 VI のインスタンスは、クローンを共有し続けます。デフォルトは、タイミングストラクチャにクローン VI を事前に割り当てます。クローンを共有する再入可能 VI 間でメモリ使用を節約するには、必要に応じてタイミングストラクチャでクローン VI を割り当てるように構成します。タイミングループまたはタイミングシーケンスストラクチャ内で VI の割り当てを設定するには、ストラクチャを右クリックして、ショートカットメニューから**共有クローン割り当て**を選択して、以下のオプションから 1 つを選択します。

- **自動**—LabVIEW はストラクチャの内容に基づいてストラクチャを**事前割り当て**または**必要に応じて割り当て**として設定します。LabVIEW は、LabVIEW が選択する設定を示すテキストでショートカットメニュー項目を追加します。たとえば、LabVIEW が**事前割り当て**を選択した場合、メニュー項目は**自動 (事前割り当て)**になります。
- **事前割り当て**—LabVIEW はストラクチャの内側での再入可能 VI の呼び出しごとにクローン VI を作成します。タイミングループまたはタイミングシーケンスストラクチャに特定のアイコンが表示されて、LabVIEW がストラクチャ内の再入可能 VI のすべてに対してクローンを事前に割り当てていることを示します。
- **必要に応じて割り当て**—LabVIEW はストラクチャの内側で再入可能 VI を呼び出すまでクローン VI を作成しません。タイミングループまたはタイミングシーケンスストラクチャに特定のアイコンが表示されて、LabVIEW がストラクチャ内の再入可能 VI のすべてに対してクローンを共有していることを示します。

複数のケースを含むストラクチャでトンネルを自動的に配線する

LabVIEW は、複数のストラクチャのケースの入力と出力トンネルを配線することができます。既存の出力トンネルを右クリックして、**入力トンネルへのリンク**ショートカットメニューから以下のオプションのうち 1 つを選択し、自動配線用に入出力トンネルを構成します。

- **作成 & 未配線のケースを配線**—LabVIEW は配線していない既存のケースのすべてでトンネルを自動的に配線します。また、LabVIEW は作成する新規ケースのすべてのトンネルも配線します。
- **作成有** LabVIEW は、作成する新規ケースのすべてでトンネルを自動的に配線します。既存のケースは未配線のまま残ります。

上記のオプションの1つを選択すると、LabVIEWは出力と入力のトンネル間でリンクを関連付けるか、作成します。両方のトンネルに白い三角形のグリフが表示され、LabVIEWがトンネルをリンクしたことを示します。出力トンネルへのリンクを右クリックして、**入力トンネルへのリンク→検索**またはショートカットメニューから**入力トンネルへのリンク→クリア**を選択して、個別に入力トンネルへのリンクを検索するか、またはトンネル間のリンクを削除します。



メモ スタックシーケンスストラクチャでは自動配線用にトンネルを構成することはできません。

その他のブロックダイアグラムの改善点

LabVIEW 8.6では、その他のブロックダイアグラムに対して以下のとおり改善点があります。

- LabVIEW 8.6では、すべての比較関数がLabVIEWクラスと一緒に動作するようになりました。
- LabVIEW 8.6は、共有ライブラリにビルドするVIのコネクタペーンでブール制御器と表示器用にプロトタイプ情報を生成します。LabVIEWで共有ライブラリを呼び出す場合、ブールは符号なし8ビット整数として処理されます。制御器ラベルがブールの場合、エラーが返されます。ブール (Boolean) は大文字と小文字を区別しません。

フロントパネルの改善点

LabVIEW 8.6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。

3D グラフの改善点

(Windows) LabVIEW 8.5以前と出荷される3D曲面グラフ、3Dパラメトリックグラフ、3D曲線グラフはそれぞれActiveX 3D曲面グラフ、ActiveX 3Dパラメトリックグラフ、ActiveX 3D曲線グラフに名前が変更されました。ActiveX 3Dグラフでは、ActiveXテクノロジと3D表現を処理するVIを使用します。

LabVIEW 8.6には、すべてのプラットフォームで使用可能な3DグラフのXControlバージョンが含まれています。3D曲面、3Dパラメトリック、3Dライングラフを使用して、3Dプロットで3Dデータを表示します。**3Dグラフプロパティ**ダイアログボックスを使用して、3DグラフXControlの外観をカスタマイズします。一般的なグラフの外観の構成、x、y、zプロットのフォーマット、グラフ軸のフォーマット、値のペアの追加、カーソルの追加を行うことができます。

フロントパネルのその他の改善点

プライベートデータ制御器にはXControlやタイプ指定XControl refnumは含まれません。

環境の改善点

LabVIEW 8.6では、LabVIEW環境に以下の改善点が含まれています。

エラーリストウィンドウの改善点

LabVIEW 8.6では、LabVIEWクラスのエラーに対する改善点があります。これは、このクラスのエラーで、**エラーリスト**ウィンドウにリストされる壊れていないクラス項目の数が少なくなる可能性があるとしてレポートしています。

パフォーマンスの改善点

VIのファイルサイズが小さくなりました。

VI 階層ウィンドウの改善点

LabVIEW 8.6 には、VI 階層ウィンドウの以下の改善点が含まれています。

- **VI 階層**ウィンドウには、LabVIEW クラスとダイナミックメンバー VI、XControl、共有ライブラリ、MathScript ノードから参照する .m ファイル、Express VI、スタティック VI リファレンスが表示されます。
- VI の呼び出しチェーンをハイライトするには、**VI 階層**ウィンドウで、一時停止 VI の VI アイコンを右クリックしてショートカットメニューから**呼び出しチェーンを表示**を選択します。呼び出しチェーンは、現在の VI からトップレベル VI への発呼者のチェーンです。階層接続をハイライト表示すると、その接続は太字の赤い線で表示されます。

VI 階層ウィンドウの詳細には、『LabVIEW ヘルプ』の目次タブにある**基本機能→VI およびサブ VI を作成する→操作手順→サブ VI を作成する→VI 階層**ブックを参照してください。

ダイアログボックスの改善点

LabVIEW 8.6 には、ダイアログボックスでの以下の改善点が含まれています。

警告ダイアログボックスでの改善点

警告ダイアログボックスは、**ロード時に発生した警告の履歴と旧バージョン用保存で発生した警告**ダイアログボックスに変更されました。概要ダイアログボックスの**詳細を表示**ボタンをクリックして、**ロード / 保存時の警告のリスト**ダイアログボックスが開いて、各警告カテゴリの詳細情報を参照します。また、**表示メニュー**から**ロード / 保存時の警告のリスト**ダイアログボックスを選択して、**ロード / 保存時の警告のリスト**ダイアログボックスを開くこともできます。

ロード時に発生した警告の履歴ダイアログボックスは、ロードしようとしたトップレベル項目の警告カテゴリのリストを報告します。

旧バージョン用保存で発生した警告ダイアログボックスは、保存しようとしているトップレベルの項目の警告カテゴリのリストを報告します。

アプリケーションビルダダイアログボックスでの変更点

LabVIEW 8.6 には、LabVIEW アプリケーションビルダで変更された以下の変更点が含まれています。

- **インストーラプロパティ**ダイアログボックスの**追加のインストーラ**ページでは、**インストーラソースの場所**テキストボックスにソースの位置のみが表示されます。このダイアログボックスではインストーラの場所は変更できなくなりました。インストーラソースを自動的に検索できない場合、ビルド処理で後から手動でソースを検索するようにプロンプトが表示されます。
- 追加のインストーラまたはコンポーネントを含むインストーラを作成すると、LabVIEW が選択したインストーラまたはインストーラの依存項目のいずれも検索できない場合、**必要なディストリビューション**ダイアログボックスでプロンプトが表示される場合があります。
- **(Windows)** ウェブサービスとして LabVIEW プロジェクト内で VI の作成とデプロイを行うことができます。**ウェブサービスプロパティ**ダイアログボックスを表示するには、**プロジェクトエクスプローラ**ウィンドウで**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**新規→ウェブサービス (RESTful)**を選択します。LabVIEW ウェブサービスは、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可能です。

シェア変数プロパティダイアログボックスの改善点

LabVIEW 8.6 では、**シェア変数プロパティ**ダイアログボックスに対して以下のとおり改善点があります。

- **シェア変数プロパティ**ダイアログボックスの**変数**ページで、**ソースにバインド**の名前が**有効**に変更されました。**有効**チェックボックスをオンにすると、シェア変数のアクセスタイプを指定できます。
- **変数**ページにある**バッファ**を使用と**1つの書き込み**オプションは、**シェア変数プロパティ**ダイアログボックスの**ネットワーク**ページに移動しました。
- **シェア変数プロパティ**ダイアログボックスの**説明**ページは、DSC モジュールがインストールされていなくても表示されるようになりました。

オプションダイアログボックスの改善点

LabVIEW 8.6 では、**オプション**ダイアログボックスの**ウェブサーバ: 構成**ページに対して以下の改善点があります。

- LabVIEW ウェブサーバに複数のインタフェースを構成できるようになりました。**サーバラベル**オプションはサーバインタフェースの名前をカスタマイズできるようにし、**サーバラベル (ポート)** リストボックスはウェブサーバ用に現在構成されているすべてのインタフェースを表示するようになりました。
- **リスナの IP アドレス**は選択したウェブサーバインタフェースの IP アドレスを指定します。このプルダウンメニューにはローカルマシン上のすべての有効なアドレスが含まれます。このフィールドは、LabVIEW Real-Time モジュールターゲットの使用時には自動的に生成されません。
- **アクセスを許可**オプションは選択したウェブサーバインタフェースの有効な機能を構成します。**アクセスを許可**セクションでは、リモートフロントパネル、スクリプト、ウェブサービス、スタティックコンテンツ、スナップショットをアクティブ化できます。

また、LabVIEW 8.6 には、**オプション**ダイアログボックスの新規の**ウェブサーバ: セキュリティ**ページが設けられています。このページで LabVIEW ウェブサービスのセキュリティを構成できます。

ダイアログボックスのその他の改善点

LabVIEW 8.6 では、その他のダイアログボックスに対して以下のとおり改善点があります。

- **競合解決**ダイアログボックスには、LabVIEW がデプロイメント中に検出する競合を解決するオプションがあります。たとえば、このダイアログボックスはオフにされたターゲットに VI をデプロイすると表示されます。競合の解決が可能な場合は、**競合解決列**にプルダウンメニューが表示されます。リストでオプションを選択し、競合を解決します。また、このダイアログボックスには、**競合**リストで競合がある前または次の項目をハイライトする**前へ**ボタンと**次へ**ボタンがあります。
- **数値プロパティ**ダイアログボックスと**数値定数プロパティ**ダイアログボックスの**データタイプ**ページ、および**数値ノードプロパティ**ダイアログボックスの**出力構成**ページには、LabVIEW に固定小数点数のオーバーフロー状態を含めるかどうかを設定する**オーバーフローステータスを含む**チェックボックスがあります。
- **見つからない項目を検索**ダイアログボックスでは、LabVIEW で検索できないディスク上の項目を参照するプロジェクト内のすべての項目を検索します。**プロジェクトエクスプローラ**ウィンドウでプロジェクトルートをクリックして、ショートカットメニューから**見つからない項目を検索**を選択して、このダイアログボックスを表示します。

パレットの改善点

LabVIEW 8.6 には、パレットでの以下の改善点が含まれています。

制御器と関数パレットセットをプログラマ的に編集する

また、パレットの編集プロパティを使用して、パレットセットをプログラマ的に作成および編集することもできます。多くのパレットを編集する、新規のパレットを作成する、またパレットの外観を確認する場合、「パレット編集 (Palette Editing)」VI を使用します。

制御器と関数パレットセットをプログラマ的に編集するサンプルについては、`labview\examples\Palette` API ディレクトリを参照してください。

パレットをプログラマ的に編集する詳細は、『LabVIEW ヘルプ』の目次タブで **VI と関数のリファレンス→プログラミング VI と関数→アプリケーション制御 VI と関数→パレット編集 VI** ブックを参照してください。

その他のパレットの改善点

LabVIEW 8.6 には、パレットで以下のその他の改善点が含まれています。

- XML パレット上に表示された VI と関数は、**LabVIEW スキーマ**パレットで使用できます。LabVIEW スキーマパレットは、XML パレット上のサブパレットです。
- 更新の際、事前にインストールされた LabVIEW のバージョンでパレット形式を復元しようとします。

環境のその他の改善点

LabVIEW 8.6 には、環境で以下のその他の改善点が含まれています。

- コマンドラインウィンドウに以下のコマンドを入力することにより、LabVIEW ビルドインストーラのサイレントインストールをアンインストールし、該当ファイルをインストールできるようにします。

```
uninst.exe /qb /xproduct name
```

 ここで、「*product name*」はアンインストールする製品名です。

入力する製品名が製品名と完全に一致しない場合、アンインストールは失敗を通知するプロンプトが表示されない状態で失敗します。uninst.exe ファイルは `C:\Program Files\National Instruments\Shared\NIUninstaller` ディレクトリにあります。 `C:\Program Files\National Instruments\Shared\NIUninstaller` ディレクトリが動作するディレクトリでない場合、uninst.exe ファイルに対するディスク上の完全パスを入力します。

- スタンドアロンアプリケーションまたは共有ライブラリのバージョン情報をプログラマ的に回収するには、`labview\vi.lib\Platform` ディレクトリにある `fileVersionInfo.llb` を使用します。VI は関数パレット上から利用できません。
- 操作メニューの**ブレイクポイント**オプションはなくなりました。代わりに**表示→ブレイクポイントマネージャ**を選択して、VI 階層でブレイクポイントを管理します。
- プロジェクトライブラリがフォルダ内にある場合、実行するソース管理の操作はプロジェクトライブラリ内の項目まで含まれるようになりました。
- `_goopsup.llb` は `labview\vi.lib\Utility` ディレクトリに移動しました。

新しく追加または変更された VI、関数、ノードでの改善点

LabVIEW 8.6 には、以下の新しく追加された VI と関数および変更された VI と関数が含まれています。VI、関数、ノードの詳細は、『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 8.6 には、以下の新しく追加された VI と関数および変更された VI と関数が含まれています。

上級ノーティファイア待機 VI および関数

上級ノーティファイア待機パレットには、以下の新しい関数があります。

- 複数からのノーティフィケーションを待機 (ノーティファイア履歴付き)
- ノーティフィケーションを待機 (ノーティファイア履歴付き)

アプリケーション制御 VI および関数

アプリケーション制御パレットには、以下の新しい VI がある新規の**パレットの編集**パレットがあります。

- パレットを読み取る
- パレットを更新する
- パレットに書き込む

比較関数

比較パレットには、以下の新しい関数があります。

- 固定小数点のオーバーフロー?

コネクティビティ VI および関数 (Windows)

コネクティビティパレットには、以下の新しい VI がある新規の**ウェブサービス**パレットがあります。



メモ 以下の VI は、LabVIEW 開発システムおよび LabVIEW プロフェッショナル開発システムでのみ使用できます。

- セッションが存在するかどうかをチェック
- セッションを作成
- セッション変数を削除
- セッションを破棄
- 出力を排出
- セッション ID クッキーを取得
- すべてのフォームデータを読み取り
- すべての要求変数を読み取り
- すべてのセッション変数を読み取り
- フォームデータを読み取り
- ポストデータを読み取り
- 要求変数を読み取り
- セッション変数を読み取り
- ESP テンプレートをレンダリング
- ESP 変数を設定
- HTTP ヘッダを設定
- HTTP リダイレクトを設定
- HTTP 応答コードを設定
- HTTP 応答 MIME タイプを設定

- 応答を書き込み
- セッション変数を書き込み

ウェブサービスパレットには、以下の新しいVIがある新規の**ユーティリティ**パレットがあります。

- HTTP URL をエスケープ
- HTTP URL をアンエスケープ

固定小数点関数

固定小数点パレットには、以下の新しい関数があります。

- 固定小数点のオーバーフローステータスをクリア
- 固定小数点のオーバーフローステータスを含める
- 固定小数点のオーバーフローステータスを削除

ヘルパー VI

ヘルパーパレットには、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に標準搭載されている「センサマッピング (Sensor Mapping)」Express VI があります。「センサマッピング (Sensor Mapping)」Express VI を使用して、3D モデルで実環境のデータをマップします。

キュー操作関数

キュー操作パレットには、以下の新しい関数があります。

- キューに要素を追加 (ロッキー)

セマフォ VI

セマフォパレットには、以下の新しいVIがあります。

- セマフォリファレンス取得
- セマフォリファレンス解放

XML VI および関数

XMLパレットには、以下の新しいVIがある新規の**XML パーサー**パレットがあります。

- XML を閉じる
- 最初の子ノードを取得 (非テキスト)
- 次の兄弟ノードを取得 (非テキスト)
- ノードテキスト内容を取得
- インボークノード (XML)
- XML をロード
- 新規 XML
- プロパティノード (XML)
- XML を保存

VISA 上級 VI および関数

VISA 上級パレットには、以下の新しいノードがあります。

- VISA プロパティノード

変更された VI、関数、ノード

以下の VI、関数、ノードは、LabVIEW 8.6 で変更されました。

上級ファイル VI および関数

上級ファイルパレットには、以下の変更された関数があります。

「ボリューム情報を取得 (Get Volume Info)」関数には、ディスク上の最小物理的ストレージ単位のサイズを示す**セクタサイズ (バイト)** 出力があります。バッファなしでファイルを開く場合、ファイル内のデータ量を所定のセクタサイズの倍にする必要があります。「ファイルを開く / 作成 / 置換 (Open/Create/Replace File)」関数の**バッファを無効**入力を TRUE に設定する場合、バッファを無効にできます。

比較関数

比較パレットには、以下の変更された関数があります。

- **範囲内と強制**—固定小数点データタイプをサポートします。
- **最大最小**—固定小数点データタイプをサポートします。

ファイル I/O VI および関数

ファイル I/O パレットには、以下の変更された関数があります。

「ファイルを開く / 作成 / 置換 (Open/Create/Replace File)」関数には、LabVIEW がバッファなしでファイルを開く、作成する、または置換することができる**バッファを無効**入力があります。

バッファを無効にして、特定の状態でデータ転送の速度を向上します。たとえば、大量のデータをストリームするか、RAID (Redundant Array of Independent Disks) でデータを格納する場合、バッファを無効にして、LabVIEW でディスクに直接データを書き込むようにします。

LabVIEW スキーマ

LabVIEW スキーマパレットには、以下の変更された関数があります。

- **XML に平坦化**—LabVIEW クラスおよび固定小数点データタイプをサポートします。
- **XML から非平坦化**—LabVIEW クラスおよび固定小数点データタイプをサポートします。

数値関数

数値パレットには、以下の変更された関数があります。

- **商**—固定小数点データタイプをサポートします。
- **逆数**—固定小数点データタイプをサポートします。
- **平方根**—固定小数点データタイプをサポートします。

レポート生成 VI

レポート生成パレットには、LabVIEW Report Generation (レポート生成) ツールキットに特有の入力と出力を持つ VI があります。VI は、Report Generation ツールキットをインストールする場合を除いて、これらの入力と出力を無視します。また、**レポート生成**パレットには、追加で以下の変更された VI があります。

- **水平ラインをレポートに追加**—幅入力のデフォルト値はピクセルからパーセンテージに変更されています。
- **テキストをレポートに追加**—LabVIEW が数値を文字に変換する際に使用する数値のフォーマットを指定する**形式文字列**を含みます。

- **VI ブロックダイアグラムをレポートに追加**—すべてのインスタンスは、レポートの画像のアライメントを設定する**アライメント**入力を含みます。
- **VI 階層をレポートに追加**—すべてのインスタンスは、レポートの画像のアライメントを設定する**アライメント**入力を含みます。
- **VI アイコンをレポートに追加**—すべてのインスタンスは、レポートの画像のアライメントを設定する**アライメント**入力を含みます。
- **VI の制御器リストをレポートに追加**—すべてのインスタンスは、レポートの画像のアライメントを設定する**アライメント**入力を含みます。
- **VI のサブVI リストをレポートに追加**—すべてのインスタンスは、レポートの画像のアライメントを設定する**アライメント**入力を含みます。
- **新規レポートライン**—VI がレポートに追加するラインの数を指定する**ライン数**入力を含みます。
- **レポートをファイルに保存**—VI が既存のファイル名を指定するかどうかを確認するダイアログボックスを表示するかどうかを決定する**置換のプロンプト?**入力を含みます。

スペクトル解析 VI(ベースパッケージにはありません)

スペクトル解析パレットには、以下の変更された VI があります。

「STFT スペクトログラム (STFT Spectrograms)」VI には、**周波数ビン**の構成を指定する**時間 - 周波数構成**入力があり、**STFT スペクトログラム {X}** の列数を決定します。また、この VI には、時間 / 周波数共同領域が時間領域のエネルギーに等しくなるように、**STFT スペクトログラム {X}** をスケールするかどうかを指定する**エネルギー節約?**入力が含まれます。

ライブラリ関数呼び出しノード

また、**ライブラリ関数呼び出し**ダイアログボックスには、以下の変更点が追加されています。

- **パラメータタブのデータタイププルダウンメニュー**には、**符号付きポインタサイズ整数**と**符号なしポインタサイズ整数**の数値データタイプがあります。これらの2つのデータタイプを使用する場合、ライブラリ関数呼び出しノードは実行されている特定のオペレーティングシステムに適合して、適切なサイズのデータをライブラリ関数に返します。64 ビットプラットフォームでは、LabVIEW はこれらの数値データタイプを 64 ビット整数タイプに変換します。32 ビットプラットフォームでは、LabVIEW はこれらの数値データタイプを 32 ビット整数タイプに変換します。
- **ライブラリ関数呼び出しノード**は、**ライブラリ名またはパス**のリファレンスを 32 ビットと 64 ビット Windows のプラットフォーム間に依存しないようにするワイルドカード(*)の使用をサポートします。拡張子の左側にある1つのインスタンスの*は、32 ビット Windows プラットフォームでは 32、64 ビット Windows プラットフォームでは 64 に変換されます。たとえば、myLibrary*.dll は myLibrary32.dll または myLibrary64.dll に変換されます。2つの*は、ライブラリ名は以前プラットフォームを参照しなかったが、新しいライブラリ名はプラットフォームを参照する状態を対処します。たとえば、myLibrary**.dll は 32 ビットプラットフォームでは myLibrary.dll、64 ビットプラットフォームでは myLibrary_64.dll に変換されます。

シェア変数ノード

シェア変数ノードには以下の変更があります。

- データを読み取るためにネットワークで共有されたシェア変数ノードの**ミリ秒タイムアウト**制御器および**タイムアウト**表示器を有効にするオプションが含まれます。
- ブロックダイアグラムで大量のシェア変数を使用して作業すると、LabVIEW 8.6 ではシェア変数のパフォーマンスの改善点が示されます。
- (LabVIEW Real-Time モジュールのインストール時にサポートされたターゲットで使用可能な) 時間トリガ変数は、標準のシェア変数とは別に表示されます。時間でトリガされた変数作成するには、サポートされたターゲットを右クリックし、ショートカットメニューから**新規→時間でトリガ変数**を選択します。

VI、関数、ノードでのその他の変更点

LabVIEW 8.6 には、VI で以下のその他の変更点があります。

「複素数をソート (Sort Complex Numbers)」VI の**メソッド**入力には名前変更された値が含まれます。**振幅**は**振幅**、**実部**、**虚部**に名前変更されています。また、**メソッド**入力には新しい値の**振幅**、**位相角**が含まれています。**振幅**、**実部**、**虚部**は最初の振幅に対する要素をソートします。振幅が同じ要素は、実数、さらに虚数に従ってソートされます。**振幅**、**位相角**は最初の振幅に対する要素をソートします。振幅が同じ要素は、位相角に従ってソートされます。位相角の範囲は $-\pi \sim \pi$ です。

新しいクラス、プロパティ、メソッド、イベント

LabVIEW 8.6 には、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。新規のクラス、プロパティ、メソッド、イベントのリストについては、『LabVIEW ヘルプ』の目次タブの**LabVIEW 8.6 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**トピックを参照してください。

XML パーサーのプロパティとメソッド

XML パーサ VI および関数と一緒に使用できる新しいプロパティとメソッドのリストについては、『LabVIEW ヘルプ』の目次タブの**プロパティとメソッドのリファレンス→XML パーサー**ブックを参照してください。

3DPC_SurfacePlot プロパティ

3D グラフで使用できる新規のプロパティリストについては、『LabVIEW ヘルプ』の目次タブの**プロパティとメソッドのリファレンス→3DPC_SurfacePlot** ブックを参照してください。

ブロックダイアグラムを自動的にクリーンアップ

編集→ダイアグラムをクリーンアップを選択して、ブロックダイアグラム上のすべての既存のワイヤを自動的に経路設定して、オブジェクトを再配置して、より整理されたレイアウトを生成します。クリーンアップオプションを構成するには、**ツール→オプション**を選択して**オプション**ダイアログボックスを表示して、**カテゴリ**リストから**ブロックダイアグラム：クリーンアップ**を選択します。制御器を含むダイアグラムの左側に制御器、右側に表示器を自動的に移動させて、ブロックダイアグラムオブジェクトとワイヤ間に指定した数のピクセルを配置して、ブロックダイアグラムのレイアウトを縮小するように LabVIEW を構成できます。

クイックドロップを使用してオブジェクトを配置

クイックドロップダイアログボックスは、名前でブロックダイアグラムまたはフロントパネルのオブジェクトを検索し、**制御器**または**関数**パレットで参照せずに、そのオブジェクトをブロックダイアグラムまたはフロントパネルに配置します。(Windows および Linux) <Ctrl-スペース> キーを押すか、**表示→クイックドロップ**を選択して、**クイックドロップ**ダイアログボックスを表示します。(中国語のキーボードでは、<Ctrl-Shift-Space> キーを押します。)(Mac OS) このダイアログボックスを開くには <Command-Shift-スペース> キーを押します。

クイックドロップダイアログボックスを使用する詳細については、『LabVIEW ヘルプ』の **目次**タブにある**基本機能→LabVIEW 環境→操作手順→オブジェクトを検索する**ブックを参照してください。

複数のオブジェクトのプロパティを編集

フロントパネルまたはブロックダイアグラム上の複数のオブジェクトを選択して、オブジェクトが共有するプロパティのどれでも編集することができます。複数のオブジェクトを選択するには、位置決めツールを使用して、編集するすべてのオブジェクトの周りを囲むように選択四角形をドラッグするか、各オブジェクトをクリックしながら <Shift> キーを押します。選択されたオブジェクトを右クリックして、ショートカットメニューから**プロパティ**を選択して、**プロパティ**ダイアログボックスを表示します。**プロパティ**ダイアログボックスは、選択するオブジェクトが共有するタブとプロパティを表示するのみです。同様のオブジェクトを選択して、タブとプロパティを表示します。共通するプロパティを共有しないオブジェクトを選択する場合、**プロパティ**ダイアログボックスはタブまたはプロパティを表示しません。

LabVIEW ウェブサービス (Windows、ベースパッケージにはありません)

LabVIEW 8.6 ではウェブサービスとして VI をパブリッシュできます。ウェブサービスはサーバーの標準メソッドを提供し、すべての HTTP クライアントがアクセスできるアプリケーションをデプロイします。LabVIEW ウェブサービスの機能は、ほとんどの主なプラットフォームとプログラミング言語を介してクライアントをサポートし、LabVIEW でネットワークを介してウェブアプリケーションを簡単に実装およびデプロイできます。

オプションダイアログボックスの**ウェブサーバ**構成ページを介してウェブサービス機能をアクティブ化できます。ウェブサービスをビルドしてデプロイするには、**プロジェクトエクスプローラ**ウィンドウで**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**新規→ウェブサービス (RESTful)**を選択して**ウェブサービスプロパティ**ダイアログボックスを表示します。LabVIEW ウェブサービスは、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可能です。

LabVIEW でウェブサービスを使用する詳細については、『LabVIEW ヘルプ』の **目次**タブにある**基本機能→LabVIEW ウェブサービス**ブックを参照してください。

NI 分散システムマネージャ

(Windows) LabVIEW 8.6 には NI 分散システムマネージャが含まれます。これは、シェア変数を管理する統一されたツールとして変数マネージャの代替になります。このシステムマネージャは、変数、プロセス、I/O サーバー、ウェブサービスを作成します。また、このシステムマネージャはシェア変数エンジンと対話してセキュリティとエイリアスを管理します。

LabVIEW でウェブサービスを使用する詳細については、『LabVIEW ヘルプ』の **目次**タブにある**基本機能→LabVIEW ウェブサービス→シェア変数を使用してライブデータを共有する**トピックを参照してください。

LabVIEW MathScript での改善点 (ベースパッケージにはありません)

LabVIEW 8.6 には、MathScript での以下の改善点と変更点が追加されています。



メモ ツール→ **MathScript ウィンドウ** を選択して、**LabVIEW MathScript ウィンドウ** を表示します。

新しい MathScript 関数

LabVIEW 8.6 には、以下の新しく追加された MathScript 関数が含まれています。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。

MathScript クラス	関数
上級	sphbesselh, sphbesselj, sphbessely
推定	csaps, lsqcurvefit
オーディオ	auread
コマンド	keyboard
フィルタ設計	gaussfir, yulewalk
フィルタ実装	cconv
積分	quad
ODE	odeset, radau5
プロット	imagesc, xlim, ylim, zlim
スペクトル解析	buffer, cpsd, mscohere, pburg, pcov, peig, periodogram, pmcov, pmusic, pwelch, pyulear, rooteig, rootmusic, specgram, spectrogram, tfestimate
偏微分方程式	pdeelliptic
サポート	warning, xlsread, xlswrite
波形生成	stepfun
ウィンドウ	taylorwin

Mac OS と Linux での LabVIEW MathScript

LabVIEW MathScript ウィンドウと MathScript ノードは Mac OS と Linux で使用できません。ただし、一部の機能はこれらのオペレーティングシステムでは使用できません。以下の表は、Mac OS または Linux で使用できない MathScript 関数を示します。x は、関数そのオペレーティングシステムで使用できないことを示します。



メモ getfileproperty 関数、load 関数、save 関数、setfileproperty 関数は Mac OS と Linux で使用できます。ただし、これらの関数はバイナリ計測ファイル (.tdm または .tdms) とは使用できません。

MathScript 関数	Mac OS	Linux
aich	x	
aiwf	x	
aoch	x	
aowf	x	
calllib	x	x
dioread	x	
diowrite	x	
dos	x	x
ginput	x	x
gtext	x	x
libfunctionsview	x	x
libisloaded	x	x
loadlibrary	x	x
pause	x	x
system	x	x
unloadlibrary	x	x
waitforbuttonpress	x	x

MathScript ノードのデバッグでの改善点

LabVIEW 8.6 には、MathScript ノードでの以下の改善点と変更点が追加されています。

- 実行のハイライト、シングルステップ、ブレイクポイントを使用して、MathScript ノードのスクリプトをデバッグします。
- MathScript の左側のグレーの領域は、以下を表示します。
 - スクリプトの行の横の赤いエラーのグリフにはエラーが含まれます。
 - 警告グリフ
 - ブレイクポイント
- エラーリストウィンドウの MathScript ノードエラーを選択して、**エラーを表示** ボタンをクリックすると、LabVIEW はエラーを含むスクリプトの行をハイライトします。

MathScript ノードのスキプットのハイライト

スキプットのハイライトはカラーを使用して、MathScript ノードのスキプットの異なる部分を区別します。これらのカラーは、スキプットを読みやすくして、エラーを含む、または予期せぬデータを返すスキプットをデバッグする操作を簡素化します。たとえば、スキプットのハイライトは、ユーザ定義の関数または変数が組み込み式 MathScript 関数をオーバーライドする際に参照できるようにします。最後に LabVIEW の従来のバージョンで保存されたカスタムカラーを使用する MathScript ノードを除いて、スキプットのハイライトはデフォルトでオンに設定されます。

特定の MathScript ノード用にスキプットのハイライトを有効または無効にするには、MathScript ノードの内側を右クリックして、ショートカットメニューから**スキプットのハイライトを有効**または**スキプットのハイライトを無効**を選択します。すべての MathScript スキプットノードに対してスキプットのを有効または無効にして、スキプットのハイライトが有効の MathScript ノードのスキプットのハイライトカラーをカスタマイズするには、**オプション**ダイアログボックスの **MathScript: スキプットのハイライト** ページを使用します。

MathScript でのその他の改善点と変更点

LabVIEW 8.6 には、MathScript で以下の改善点と変更点があります。

- **VI 階層** ウィンドウには、MathScript ノードから参照する .m ファイルが表示されます。
- `delete` 関数には、プロットプロジェクトを削除して **LabVIEW MathScript ウィンドウ** で **変数リスト** からオブジェクトを削除できる **obj** 入力が含まれます。
- `filter` 関数には、フーリエ変換の代わりに直接コンボリューションを使用するように LabVIEW を設定する **'direct'** 入力が含まれます。
- LabVIEW MathScript は、パフォーマンス向上のために **LabVIEW MathScript ウィンドウ** と MathScript ノードの両方をより速くコンパイルします。
- VI に MathScript ノードがある場合、LabVIEW は VI の保存または実行時にすべての取り消し情報を削除します。

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

固定小数点数のオーバーフローを管理する

オーバーフロー状態は、固定小数点数に対して演算操作を行った場合、または「固定小数点に変換 (To Fixed-Point)」関数によって数値データを固定小数点データに変換した場合に発生することがあります。オーバーフローが発生したかどうかを決めるには、オーバーフローステータスを含めるように固定小数点数を構成します。LabVIEW は、固定小数点数でオーバーフローステータスを含める場合、固定小数点がオーバーフローした操作の結果であるかどうかをトラックするための追加の格納スペース割り当てます。

オーバーフローステータスを含む固定小数点数を構成すると、固定小数点制御器、定数、表示器でオーバーフローステータス LED が表示されます。この LED は、固定小数点数のオーバーフローステータスが TRUE の場合に点灯します。また、「固定小数点のオーバーフロー ? (Fixed-Point Overflow?)」関数で固定小数点値のオーバーフローステータスを決めることもできます。固定小数点関数は固定小数点値のオーバーフローステータスを操作します。

固定小数点数の詳細については、『LabVIEW ヘルプ』の**目次**タブにある**基本機能**→**ブロックダイアグラムを作成する**→**概念**→**数値データ**ブックを参照してください。

LLB を統合する (ベースパッケージにはありません)

LabVIEW プロフェッショナル開発システムでは、**統合する LLB を選択**ダイアログボックスで、LLB とその元の LLB の 2 つのリビジョン間での差異を統合および解決します。

ツール→統合→LLB を統合を選択して、**統合する LLB を選択**ダイアログボックスを表示します。**ベース LLB** フィールドで元の LLB を指定します。**統合 LLB 1** と **統合 LLB 2** フィールドで 2 つの修正された LLB を指定します。**統合** ボタンをクリックして、選択された LLB を統合して、**LLB を統合**ダイアログボックスを開きます。LLB のリビジョン間の差異を解決して、**閉じる** と **保存** をクリックして、統合された LLB を保存します。

LLB の統合の詳細は、『LabVIEW ヘルプ』の **目次** タブにある **基本機能→開発ガイドライン (英語)→操作手順→VI と LLB を統合する→LLB を統合する** トピックを参照してください。

National Instruments, NI, ni.com, および LabVIEW は National Instruments Corporation (米国ナショナルインスツルメンツ社) の商標です。National Instruments の商標の詳細については、ni.com/legal の「Terms of Use」セクションを参照してください。MATLAB® は The MathWorks, Inc. (米国マサワークス社) の登録商標です。本文中に記載されたその他の製品名および企業名は、それぞれの企業の商標または商号です。National Instruments の製品を保護する特許については、ソフトウェアに含まれている特許情報 (ヘルプ→特許情報)、メディアに含まれている 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。