

LabVIEW™ 2020 アップグレード ノート

このアップグレードノートは、Windows、macOS、および Linux 用の LabVIEW を LabVIEW 2020 にアップグレードする方法を説明します。アップグレードを行う前に、以下のトピックの情報について、このドキュメントを参照してください。

- LabVIEW をアップグレードする際の推奨プロセス
- LabVIEW の旧バージョンで保存された VI をロードする前に考慮すべき互換性の問題
- LabVIEW 2020 に新たに追加された機能および動作

目次

LabVIEW 2020 にアップグレードする.....	2
推奨アップグレードプロセスの概要.....	2
1.VI およびマシン構成のバックアップを作成する.....	2
2.VI の既存の動作をテストして記録する.....	4
3.LabVIEW、アドオン、およびデバイスドライバをインストールする.....	4
4.VI を変換し、動作の変化を検証する.....	5
アップグレードに関する一般的な問題のトラブルシューティング.....	7
アップグレードおよび互換性の問題.....	7
LabVIEW 2016 からアップグレードする.....	8
LabVIEW 2017 からアップグレードする.....	8
LabVIEW 2018 からアップグレードする.....	9
LabVIEW 2019 からアップグレードする.....	12
LabVIEW 2020 の機能および変更点.....	13
LabVIEW インタフェースを使用してコードの柔軟性を向上させる.....	13
LabVIEW Web サービスの改善点.....	15
アプリケーションビルダの改善点.....	16
環境の改善点.....	16
「共有ライブラリをインポート」ウィザードの改善点.....	19
ブロックダイアグラムの改善点.....	20
追加および変更された VI および関数.....	22
追加および変更されたプロパティ、メソッド、およびイベント.....	24
旧バージョンの LabVIEW の機能と変更点.....	26

LabVIEW 2020 にアップグレードする

小さなアプリケーションは、LabVIEW の新しいバージョンをインストールしてから VI をロードすることにより新しいバージョンにアップグレードできますが、NI では、アップグレード時の問題をできるだけ効率的に検出して修正できるよう、より綿密なアップグレードプロセスを推奨しています。



ヒント このプロセスは、重要な操作を制御または監視する、長時間のダウンタイムが許されない、複数のモジュール、ツールキット、ドライバを使用する、またはサポートされないバージョンの LabVIEW で保存された大規模な LabVIEW アプリケーションで特に効果があります。現在主要サポートの対象となっている LabVIEW のバージョンについては、NI Web サイト (ni.com/info) で Info Code に「[lifecycle](#)」と入力してください。

推奨アップグレードプロセスの概要

VI とマシン構成を
バックアップする

VI と開発マシンを保護することにより、必要な場合にはファイルを復元してアップグレードプロセスをやり直せるようにしておきます。

VI の既存の動作を
テストして記録する

前バージョンの LabVIEW で VI の標準的動作を把握しておくことにより、アップグレードによって生じる動作の変化をより正確に検知できます。

LabVIEW、アドオン、
デバイスドライバを
インストールする

すべての NI ソフトウェアを同時にアップグレードすると、アップグレード後の VI は、最新バージョンの LabVIEW 環境に必要なサブ VI、パレット、およびプロパティノードを検出できます。

VI を変換して、
動作の変化を検証する

アップグレード直後に VI を変換してテストすることにより、VI が予想どおりに動作することを確認し、動作が変化した場合でも、実行開始前に修正することができます。



メモ LabVIEW 5.1 以前からアップグレードするには、まず LabVIEW の中間バージョンにアップグレードする必要があります。LabVIEW の特定のレガシーバージョンからのアップグレードについては、NI Web サイト (ni.com/info) で Info Code に「[upgradeOld](#)」と入力してください。

1. VI およびマシン構成のバックアップを作成する

必要に応じて VI を以前の機能に復元してアップグレードプロセスを再度実行できるように、LabVIEW 2020 にアップグレードする前に、VI のコピーおよび開発マシンまたは本稼動マシンの構成 (可能な場合) を保護しておきます。

a. VI のバックアップを作成する

LabVIEW をアップグレードする前に VI をバックアップしておく、バックアップコピーの状態にすぐ戻すことができます。バックアップコピーがない場合、各 VI を旧バージョン用に保存しないと、アップグレードした VI を旧バージョンの LabVIEW で開けなくなります。

ソースコード管理に VI をサブミットすることにより、VI をバックアップできます。これにより、VI のアップグレードによる動作の変化に対応できない場合に、VI をこのバージョンに戻すことができます。

LabVIEW でソースコード管理プロバイダを使用する方法については、『LabVIEW ヘルプ』で **基本機能** → **プロジェクトとターゲットを操作する** → **概念** → **LabVIEW でソース管理を使用する** トピックを参照してください。

b. マシン構成のバックアップを作成する

新しいバージョンの LabVIEW をインストールすると共有ファイルが更新されるため、旧バージョンの VI の動作も影響を受けることがあります。しかし、それらの共有ファイルを更新後に前のバージョンに戻すことは非常に困難です。このため、サポートされていないバージョンの LabVIEW からアップグレードする場合、またはアプリケーションのダウンタイムコストが高い場合には特に、以下のいずれかの方法で、NI ソフトウェアの構成をバックアップすることを検討してください。

- マシン構成のバックアップイメージを作成する—アップグレードする前に、インストールされているソフトウェア、ユーザ設定、ファイルなどのマシンのディスク状態を、ディスクイメージングソフトウェアを使用して保存します。アップグレードした後でマシンを元の構成に戻すには、バックアップしたディスクイメージをデプロイします。
- テストマシンでアップグレードプロセスをテストする—テストマシン (通常は仮想マシン) を使用し、アップグレードプロセスをテストします。テストマシンでアップグレードを行うとバックアップイメージを作成するよりも時間はかかりますが、NI では、実際の工程を制御または監視するマシンのダウンタイムを回避または最小化する必要がある場合にはこのアプローチを強く推奨しています。アップグレードに起因するすべての問題をテストマシンで解決した後、テストマシンを本稼動マシンに置き換えるか、本稼動マシンでこのアップグレードプロセスを再実行します。



ヒント アップグレードした VI がテストマシンと開発マシンで異なる動作をする可能性を最小化するため、開発マシンの CPU、RAM、オペレーティングシステム、ソフトウェアのバージョンなどの装備とできるだけ一致したテストマシンを使用してください。

2.VIの既存の動作をテストして記録する

VIをアップグレードすると、LabVIEWの旧バージョンとLabVIEW 2020の間の違いによってVIの動作が変化する場合があります。両方のバージョンでVIをテストしてその結果を比較することで、アップグレードに起因する動作の変化を検知できます。このため、以下のいずれかのテストの現在の結果が手元にあることを確認してください。

- 壊れているVIを特定する一括コンパイルログ

特に複数の人がそのVIの開発に関わっていたり、それらのVIの中に最近コンパイルされていないVIがある可能性がある場合は、アップグレードする前にVIを一括コンパイルすることが有益です。この一括コンパイルログを生成するには、**一括コンパイルダイアログボックスで結果をログチェックボックスをオンにします**。VIの一括コンパイルの詳細については、『LabVIEW ヘルプ』で**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を一括コンパイルする**トピックを参照してください。

- 各VIが意図した機能を正しく実行しているかどうかを検証するユニットテスト
- プロジェクトまたはサブVIのグループがまとめて予期した動作をしているかどうかを検証する統合テスト
- VIがデスクトップやFPGAターゲットなどのターゲットにデプロイされたときに予想どおりに動作するかどうかを検証するデプロイメントテスト
- CPU使用率、メモリ使用、コード実行速度のベンチマーク測定する性能テスト

パフォーマンスおよびメモリをプロファイルウィンドウを使用すると、VIの平均実行速度を計算することができます。

- VIが予期しないデータを正しく処理できるかどうかを検証するストレステスト



メモ テスト結果をもとにVIを変更した場合は、この先に進む前に新しいバージョンのVIをバックアップします。

VIテストの詳細については、『LabVIEW ヘルプ』で**基本機能→アプリケーション開発と設計ガイドライン→概念→大規模アプリケーションを開発する→開発モデルの段階→アプリケーションをテストする**トピックを参照してください。

3.LabVIEW、アドオン、およびデバイスドライバをインストールする

a. LabVIEW (モジュール、ツールキット、およびドライバを含む) をインストールする
新しいバージョンのLabVIEWにアップグレードした場合、新しい開発システムだけでなく、その新しいバージョンと互換性のあるモジュール、ツールキット、およびドライバをインストールする必要があります。

b. user.lib ファイルをコピーする

旧バージョンの LabVIEW で作成したカスタム制御器と VI を LabVIEW 2020 で使用するには、旧バージョンの LabVIEW の labview\user.lib ディレクトリのファイルを LabVIEW 2020 の labview\user.lib ディレクトリにコピーします。

c. VI パッケージを再インストールする

JKI VI Package Manager (VIPM) を使用して LabVIEW の前のバージョンに VI パッケージをインストールした場合は、VIPM ソフトウェアを使用して LabVIEW 2020 にすべてのパッケージを再インストールしてください。

4.VI を変換し、動作の変化を検証する

LabVIEW 2020 で VI を一括コンパイルすると、VI は LabVIEW の最新バージョンに変換され、壊れた VI を特定するために役立つエラーログが作成されます。この情報を「[アップグレードおよび互換性の問題](#)」と併せて使用することにより、LabVIEW の最新バージョンに関連する動作変化を特定して修正できます。



メモ NI では、LabVIEW でソース管理を使用して VI をバックアップし、変更をトラッキングすることを推奨しています。これにより、VI のアップグレードによる動作の変化に対応できない場合に、VI を前のバージョンに戻すことができます。

a. VI を新しいバージョンの LabVIEW で一括コンパイルする

一括コンパイルされた VI は、LabVIEW 2020 に変換されて保存されます。しかし、各 VI またはプロジェクトに対して **ファイル→旧バージョン用に保存** を選択しない限り、VI を旧バージョンの LabVIEW で開けなくなります。このため、新しいバージョンの LabVIEW に変換したい VI のみを一括コンパイルしてください。アップグレードにより発生した問題を特定するには、**一括コンパイルダイアログボックスの結果をログ** チェックボックスをオンにして一括コンパイルログを作成します。



メモ FPGA またはリアルタイムリソースが含まれている VI を一括コンパイルすると、**一括コンパイルダイアログボックス**に、これらの VI は実行不可能な VI であるとレポートされる場合があります。エラーを確認するには、FPGA または RT ターゲットの下にある VI を、プロジェクト内で必要な FPGA リソースまたはリアルタイムリソースとともに開きます。

VI の一括コンパイルの詳細については、『LabVIEW ヘルプ』で以下のトピックを参照してください。

- **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 を一括コンパイルする**
- **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→一括コンパイルの一般的なステータスメッセージ**

b. 壊れた VI を修正する

旧バージョンの LabVIEW と LabVIEW 2020 との違いにより、変更された機能が使用されている VI が壊れる場合があります。壊れた VI を LabVIEW 2020 で特定して、修正するには、以下の手順を実行してください。

1. アップグレード中に壊れた VI を特定するには、前のステップで作成した一括コンパイルエラーログと、VI の既存の動作を確認するために作成したログを比較します。
2. LabVIEW への更新により各 VI が壊れたか確認するには、「[アップグレードおよび互換性の問題](#)」参照してください。

c. 動作の変化を特定して修正する

NI では、LabVIEW のバージョン間における VI の動作の違いを最小限に抑えるために多大な努力を払っていますが、VI の改善やバグ修正の結果、VI の動作が変化することがあります。LabVIEW の新しいバージョンで VI の動作が変更するかどうかを素早く確認するには、以下のツールを使用します。

- VI Analyzer Upgrade Test を実行する一大規模な VI セットの場合、これらのテストにより、アップグレードに起因する多くの動作変化を効率的に特定できます。これらのテストを取得して使用するには、以下の手順に従います。
 1. 使用している旧バージョン以降の LabVIEW のすべてのバージョンの VI Analyzer Upgrade Test をダウンロードします。これらのテストをダウンロードするには、NI Web サイト (ni.com/info) で Info Code に「[analyzevi](#)」を入力してください。
 2. **ツール→VI アナライザ→VI を解析**を選択して、新規の VI アナライザタスクを開始することにより、テストを開いて、実行します。プロジェクト全体をまとめて解析するには、個々の VI からではなく、**プロジェクトエクスプローラ**ウィンドウでこのメニュー項目を選択します。
 3. テストに失敗した場合は、テストに対応する LabVIEW のバージョンの「[アップグレードおよび互換性の問題](#)」を参照して修正します。
- アップグレードドキュメントを読む
 - [アップグレードおよび互換性の問題](#)—VI を壊したり、VI の動作に影響を与える可能性のある変更が記載されています。前のバージョンから始めて、LabVIEW の各バージョンのサブセクションを参照します。



ヒント 「アップグレードおよび互換性の問題」トピックで言及されている廃止されたオブジェクトやその他のオブジェクトを手早く見つけるには、アップグレードした VI を開き、**編集→検索して置換**を選択します。

- LabVIEW 2020 既知の問題リスト—LabVIEW 2020 のリリース前およびリリース中に発見されたバグが記載されています。このリストを参照するには、NI Web サイト (ni.com/info) で Info Code に「[lv2020ki](#)」を入力してください。

「アップグレード—動作の変化」と「アップグレード—移行」セクションがある場合は、アップグレードした VI の動作に影響するバグの回避方法についてそれらを参照してください。

- モジュールおよびツールキットのドキュメント — LabVIEW FPGA および LabVIEW Real-Time モジュールなど、一部のモジュールおよびツールキットに特有のアップグレード問題が記載されています。
- 各ドライバの Readme ファイル — 各ドライバに特有のアップグレード問題が記載されています。各ドライバの Readme を表示するには、ni.com/manuals を参照してください。



ヒント 動作が変化した原因が LabVIEW のアップデートではなく、ドライバのアップデートによるものであるかどうかを確認するには、LabVIEW 2020 をインストールした後に、その VI を旧バージョンの LabVIEW でテストしてください。

- ユーザ独自のテストを実行する — 旧バージョンの LabVIEW で行ったものと同じテストを LabVIEW 2020 の VI に対して行い、結果を比較します。新しい動作を特定した場合は、その変化の原因を診断するために、アップグレードドキュメントを参照します。

アップグレードに関する一般的な問題のトラブルシューティング

以下のアップグレードに関する問題の解決法の詳細については、`labview\manuals` ディレクトリにインストールされている『`troubleshooting_guide.html`』ドキュメントを参照してください。

- 見つからないモジュールまたはツールキットの機能を探す
- 見つからないサブ VI、パレット、プロパティノードを探す
- 旧バージョンの LabVIEW で作成した VI を LabVIEW 2020 で開けない理由を特定する
- インストールされている NI ソフトウェアのバージョンを特定する
- VI を旧バージョンの LabVIEW に復元する

アップグレードおよび互換性の問題

VI を壊したり、動作を変更したりする可能性のある、LabVIEW のバージョンごとの変更については、以下のセクションを参照してください。

LabVIEW の新しいバージョンの既知の問題、その他の互換性に関する問題、LabVIEW 2020 の最新の追加機能の詳細については、`labview` ディレクトリにある『`readme.html`』ファイルを参照してください。

LabVIEW 2016 からアップグレードする

LabVIEW 2016 から LabVIEW 2020 へアップグレードすると、以下の互換性問題が発生する可能性があります。

アクターフレームワーク VI の動作の変更

LabVIEW 2016 以前では、ネストされたアクターは、起動前初期化メソッドのエラーが原因で起動できないと、エラーを返し、そのエラーを含んだ最終 Ack メッセージを発呼者アクターに送信します。LabVIEW 2017 以降では、ネストされたアクターは、最終 Ack メッセージを送信せずにエラーを返します。

LabVIEW 2017 からアップグレードする

LabVIEW 2017 から LabVIEW 2020 へアップグレードすると、以下の互換性問題が発生する可能性があります。

LabVIEW ランタイムエンジンの下位互換性

LabVIEW 2017 以降では、LabVIEW ランタイムエンジンの下位互換性がサポートされています。旧バージョンの LabVIEW で作成されたバイナリおよび VI を、バイナリを再度ビルドしたり、VI を現在のバージョンの LabVIEW で一括コンパイルしたりせずにロードおよび実行できます。たとえば、LabVIEW 2017 以降のバージョンでは、LabVIEW 2017 でビルドされたバイナリおよび VI を再コンパイルせずにロードできます。この改善の対象となるのは、スタンドアロンアプリケーション (EXE)、共有ライブラリ (DLL)、およびバックプロジェクトライブラリです。

バイナリで下位互換性を有効にするには、ビルド仕様に基づいて、特定のダイアログボックスの「上級」ページで以下のチェックボックスをオンにします。

ビルド仕様	ダイアログボックス	チェックボックス
スタンドアロンアプリケーション (EXE)	アプリケーションプロパティ	今後の LabVIEW ランタイムバージョンでこのアプリケーションの実行を許可する
バックプロジェクトライブラリ	バックライブラリプロパティ	今後の LabVIEW バージョンでこのバックライブラリのロードを許可する
共有ライブラリ (DLL)	共有ライブラリプロパティ	今後の LabVIEW バージョンでこの共有ライブラリのロードを許可する

LabVIEW 2017 以降で作成したビルド仕様では、これらのオプションはデフォルトで有効になっています。ビルド仕様を特定のバージョンの LabVIEW に関連付ける場合は、これらのオプションを無効にします。これらのオプションを無効にすると、パフォーマンスプロファイルへの変更を防ぐことができる上、コンパイラのアップグレードに

よる予期しない問題の発生を回避できます。リアルタイムアプリケーションでは、ダイアログボックスにこれらのオプションは表示されませんが、この機能はデフォルトで有効になっています。

レポート生成 VI の動作の変更

LabVIEW 2018 では、レポート生成 VI は、標準レポート形式のレポートの生成をサポートしなくなりました。レポートは、HTML、Word、または Excel 形式でのみ生成できます。この動作変更により、次の VI は廃止予定となりました。

- 簡易 VI パネルまたはドキュメントを印刷—この VI は廃止予定となりました。代わりに、「VI パネルまたはドキュメントを印刷」VI を使用してください。
- 簡易テキストレポート—この VI は廃止予定となりました。代わりに、「簡易テキストレポートを作成」VI を使用してください。
- レポートタイプ取得—この VI は廃止予定となりました。代わりに、「レポートタイプ」VI を使用してください。
- 新規レポート—この VI は廃止予定となりました。代わりに、「レポートを作成」VI を使用してください。
- レポートタブ幅設定—この VI は廃止予定となりました。

廃止予定の VI、関数、およびノード

LabVIEW 2018 以降では、「数値を列挙体に変換」VI はサポートされません。代わりに、「タイプに強制変換」関数を使用してください。

LabVIEW 2018 からアップグレードする

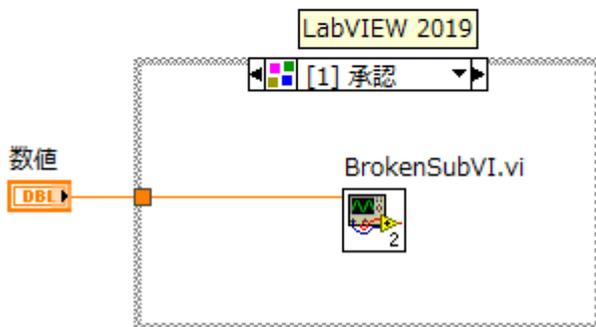
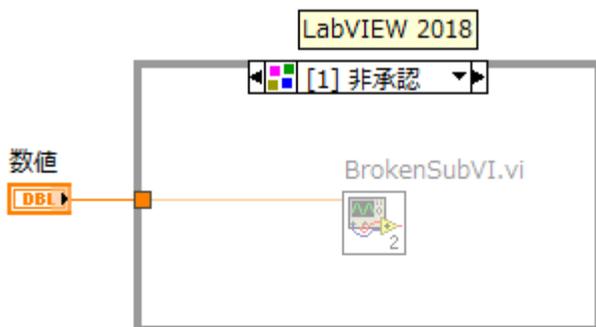
LabVIEW 2018 から LabVIEW 2020 へアップグレードすると、以下の互換性問題が発生する可能性があります。

- 「パッケージプロパティ」ダイアログボックスの**パッケージ属性**ページが**パッケージ**ページと名前変更されました。
- 「配列、行列&クラスタ」パレットが「データコンテナ」と名前変更されました。
- データタイプ解析 VI のトップレベル端子名が、大文字から小文字に変更されました。クラスタ要素など、サブレベルの端子名は変更されていません。

タイプ特化ストラクチャの動作の変更

LabVIEW 2019 では、タイプ特化ストラクチャがサブダイアグラムを受け入れるか拒否するかを決定する際の構文エラーのチェック動作が変更されました。LabVIEW 2018 以前では、タイプ特化ストラクチャは、ストラクチャ内のエラー（壊れたワイヤなど）およびサブ VI や他の依存項目からのエラーを構文エラーとして認識します。

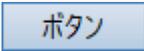
LabVIEW 2019 以降では、タイプ特化ストラクチャは、ストラクチャ内のエラーのみを構文エラーとして認識します。



タイプ特化ストラクチャの詳細については、『LabVIEW ヘルプ』で **VI と関数のリファレンス→プログラミング VI および関数→ストラクチャ→タイプ特化ストラクチャ** トピックを参照してください。

クラシック、システム、または NXG スタイルでブール制御器および表示器を作成する際の動作の変更

LabVIEW 2019 では、VI がクラシック、システム、または NXG スタイルを使用して制御器および表示器を作成するように構成されている場合に、ブール端子から制御器および表示器を作成する際の動作が変更されました。次の表は、LabVIEW 2018 以前と LabVIEW 2019 以降で作成されたブール制御器および表示器の外観を比較したものです。

スタイル	種類	LabVIEW 2018 以前	LabVIEW 2019 以降
クラシック	制御器		<input checked="" type="checkbox"/> OFF/ON
システム	制御器/表示器	<input type="radio"/> OFF/ON	<input checked="" type="checkbox"/> OFF/ON
NXG	制御器		<input checked="" type="checkbox"/> Off/On



メモ 制御器および表示器を作成する際のスタイルを構成するには、**ファイル** → **VI プロパティ** を選択し、**カテゴリ**プルダウンメニューから**編集オプション**を選択した後、**制御器/表示器を作成時の制御器スタイル**リストから適切なスタイルを選択します。

この動作の変更は、次の方法を使用して作成されるブール値の制御器と表示器に適用されます。

- 制御器を作成または表示器を作成メソッドを使用して作成する。
- ブール端子を右クリックし、ショートカットメニューから**作成** → **制御器**または**作成** → **表示器**を選択して作成する。

「データ値リファレンス読み取り/書き込み」境界ノードによる自動エラー処理への動作変更

In Place 要素ストラクチャに 1 組のデータ値リファレンス読み取り/書き込み境界ノードを配置した場合、左右両方の境界ノードに**エラー出力**端子があります。LabVIEW 2018 では、VI で自動エラー処理が有効になっている時にエラーが発生すると、LabVIEW は各未配線の**エラー出力**端子にエラーダイアログボックスを表示します。LabVIEW 2019 以降では、未配線の**エラー出力**端子の数に関わらず、LabVIEW は 1 つのエラーダイアログボックスのみを表示します。

「データ値リファレンス要素読み取り/書き込み」境界ノードの詳細については、『LabVIEW ヘルプ』で **VI と関数のリファレンス** → **プログラミング VI** および **関数** → **ストラクチャ** → **In Place 要素ストラクチャ** → **データ値リファレンス要素読み取り/書き込み境界ノード**トピックを参照してください。

パスワード保護されたライブラリでのコミュニティメンバーの呼び出しへの動作変更
LabVIEW 2018 では、フレンド VI がパスワードで保護されているライブラリ内のコミュニティメンバーを呼び出す場合、フレンド VI を初めて編集または実行するときに

ライブラリのパスワードを入力する必要があります。LabVIEW 2019 以降では、ライブラリのパスワードを入力せずにフレンド VI を編集または実行することができます。

定数、制御器、および表示器でのテキストのオーバーフローを通知する

LabVIEW 2019 では、デフォルトで、文字列、数値、タイムスタンプ、テキストリングと列挙体、およびコンボボックスで、表示テキストが収まりきらずに切れている場合、テキストがフェードアウトして矢印が表示されます。LabVIEW 2018 以前でテキストオーバーフローを有効にするには、制御器、表示器、クラスタまたはクラスタ配列内の定数を右クリックし、**表示項目** → **テキストオーバーフロー** を選択します。

データタイプ解析 VI の端子名の変更

データタイプ解析 VI のトップレベル端子名が、大文字から小文字に変更されました。クラスタ要素など、サブレベルの端子名は変更されていません。

データタイプ解析 VI の詳細については、『LabVIEW ヘルプ』で **VI と関数のリファレンス** → **プログラミング VI および関数** → **クラスタ、クラス、バリエーション VI および関数** → **バリエーション VI および関数** → **データタイプ解析 VI** ブックを参照してください。

LabVIEW 2019 からアップグレードする

LabVIEW 2019 から LabVIEW 2020 へアップグレードする際に、以下の互換性問題が発生する可能性があります。

MathScript 機能のサポートの終了

LabVIEW MathScript は、新規の設計には推奨されません。詳細および推奨の代替方法については、ni.com/migratemathscript を参照してください。

リモートフロントパネルでの SSL 暗号化のサポートの終了

LabVIEW 2020 では、**オプションダイアログボックスの Web サーバ** ページの **リモートパネルサーバ** セクションで **SSL チェックボックス** をオンにしても、SSL (Secure Sockets Layer) はサポートされなくなりました。リモートフロントパネルサーバにセキュア接続しようとする、LabVIEW はエラーを返します。

廃止予定の VI、関数、およびノード

LabVIEW 2020 以降では、「MD5 チェックサムファイル」VI はサポートされません。「ファイルチェックサム」VI を使用してください。

「親メソッド呼び出し」ノードの名前変更

「親メソッド呼び出し」ノードの名前が「親クラスメソッド呼び出し」に変更されました。

LabVIEW 2020 の機能および変更点

意見交換アイコン  は、ni.com の [NI Idea Exchange](#) ディスカッションフォーラム (英語) での製品に関する提案を元に開発された新機能を示しています。

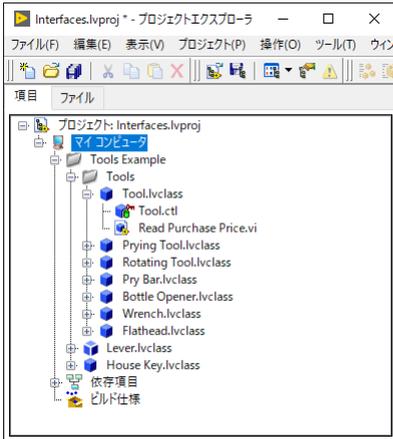
LabVIEW の各バージョン特有のアップグレードおよび互換性の問題については、「[アップグレードおよび互換性の問題](#)」を参照してください。既知の問題、修正されたバグの部分的なリスト、その他の互換性に関する問題、LabVIEW 2020 における追加機能については、`labview\readme` ディレクトリにある『`readme.html`』ファイルを参照してください。

LabVIEW インタフェースを使用してコードの柔軟性を向上させる

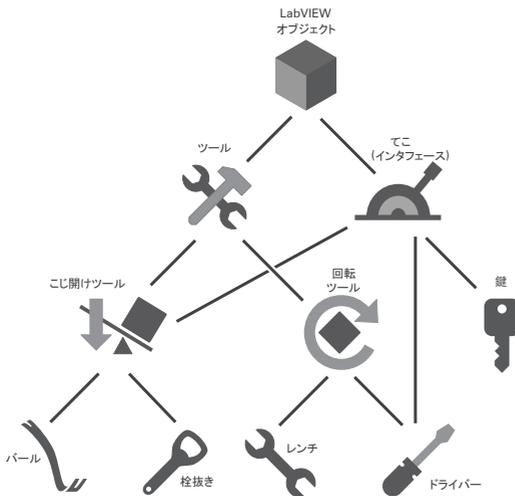
LabVIEW 2020 では、インタフェースが導入されました。インタフェースはプライベートデータ制御のないクラスと考えることができますが、そのわずかな違いにより、インタフェースは、ソフトウェアアーキテクチャの中でクラスとはまったく異なる目的に使用することができます。具体的には、インタフェースでは複数の継承が可能です。

インタフェースは、オブジェクトがどのように役割を実行するかは定義せずに、そのオブジェクトが満たすことができる役割を宣言します。クラスは、インタフェースから継承することにより、そのオブジェクトが特定の役割を果たすことを宣言します。また、その動作の実行方法を指定する責任を負います。複数のインタフェースを継承するクラスは、そのオブジェクトをさまざまな役割を必要とする複数のソフトウェアモジュールに渡すことができます。

以下のプロジェクトには、Tool (ツール) クラスとさまざまなツール子クラスが含まれています。また、Lever (てこ) インタフェースも含まれています。Tools クラスにはクラス内のデータを定義する制御器がありますが、インタフェースにはプライベートデータがないため、Lever (てこ) インタフェースには制御器がありません。「クラス」は、面の枠が塗りつぶされた立方体  として表示されます。「インタフェース」は、面の枠が白線の立方体  として表示されます。インタフェースとクラスのファイル拡張子は、どちらも `.lvclass` です。



以下の図は、「ツール」クラス、「鍵」クラス、「てこ」インタフェースの間の継承関係を示しています。「ツール」クラスと「てこ」インタフェースは、どちらも LabVIEW オブジェクトを継承します。「こじ開けツール」クラスと「ドライバー」クラスは、それぞれ独自のメソッドと親クラスのメソッドをもつ他、「てこ」としても使用できるため、「てこ」インタフェースからも継承します。これらのクラスは、「てこ」インタフェースのメソッドを継承します。「てこ」はクラスではなくインタフェースなので、この多重継承は問題ありません。互いに関連のない(こじ開けツール、ドライバー、鍵) という 3 つのクラスは、LabVIEW オブジェクト以外に Lever (てこ) という共通の祖先をもちます。



インタフェースは、以下の方法で作成できます。

- **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、ショートカットメニューから **新規** → **インタフェース** を選択する。
- **ファイル** → **新規** を選択し、**新規作成**リストから **他のファイル** → **インタフェース** を選択する。
- LabVIEW クラス:インタフェースを作成メソッドを使用する。

インタフェースの新しいプロパティおよびメソッドの詳細については、「追加および変更されたプロパティ、メソッド、およびイベント」セクションを参照してください。

インタフェースを使用するサンプルについては、以下のプロジェクトを参照してください。

- `labview\examples\Object-Oriented Programming\Basic Interfaces\Basic Interfaces.lvproj`
- `labview\examples\Object-Oriented Programming\Actors and Interfaces\Actors and Interfaces.lvproj`

LabVIEW Web サービスの改善点

LabVIEW 2020 では、LabVIEW Web サービスを開発するための機能に加え、Web サービスを NI Web サーバにパブリッシュするための機能を提供します。アプリケーション Web サーバへのパブリッシュも引き続きサポートされます。

NI Web サーバは、LabVIEW Web サービスや NI が作成した SystemLink サービスなどのユーザが作成したサービスをホストできる堅牢な Web サーバです。NI Web サーバは、Web アプリケーションを一般的な Web セキュリティの脅威から保護し、エンタープライズ級のデータサービスに高度なスケーラビリティを提供してデバイス管理を可能にします。



メモ NI Web サーバは Windows (64 ビット) のみをサポートします。アプリケーション Web サーバは、Windows (32 ビットおよび 64 ビット) および RT ターゲットの両方をサポートします。

Web サービスは、スタンドアロンアプリケーション、パッケージ、またはパッケージインストーラを通じて NI Web サーバにパブリッシュできます。NI Web サーバで SSL (Secure Sockets Layer) 暗号化を有効にするか、各ユーザ役割に異なる権限を割り当てることにより、Web クライアントと LabVIEW Web サービス間のセキュアな通信を確立できます。

Web クライアントへのデフォルトの応答形式が、XML から JSON 文字列に変更されました。Web ブラウザで Web サービスからのデータを表示し、相互作用するユーザインタフェースを作成する場合は、WebVI を Web サービスに統合してください。

WebVI の詳細については、『[LabVIEW NXG Web Module Manual](#)』を参照してください。

LabVIEW Web サービスの開発、ホスティング、パブリッシュの詳細については、『LabVIEW ヘルプ』の **基本機能** → **データを転送する** → **ネットワークを介してデータを**

転送する → Web クライアントから LabVIEW アプリケーションと通信する → 概要:
LabVIEW アプリケーションとの Web ベースの通信トピックを参照してください。

アプリケーションビルダの改善点

LabVIEW 2020 では、LabVIEW アプリケーションビルダおよびビルド仕様が以下のように改善されました。

(Windows) パッケージを修復する

インストールが正しく実行されない場合、NI パッケージマネージャでパッケージを修復できます。LabVIEW では、NI パッケージマネージャがパッケージを修復するときに実行するカスタム修復動作を構成できます。カスタム修復動作を作成するには、**パッケージプロパティ** ダイアログボックスの「**上級**」ページで、**カスタム動作表**を右クリックし、**修復動作を追加**を選択します。以下のオプションから選択できます。

- **修復前**—現在のパッケージを修復する前に実行する動作を指定します。
- **修復後**—現在のパッケージを修復した後に実行する動作を指定します。
- **すべての修復後**—すべてのパッケージを修復した後に実行する動作を指定します。

アプリケーションビルダのその他の改善点

「**パッケージプロパティ**」ダイアログボックスの**フィード**ページには、以下のオプションが新たに追加されました。

- **フィードにすべての依存項目を含める**—パッケージのすべての依存項目をフィードに追加するか、パッケージのみをフィードに追加するかを指定します。
- **パッケージバージョンをフィードの場所に追加する**—フィードにパッケージバージョンを含めるかどうかを指定します。LabVIEW は、フィードの場所にパッケージバージョン付きのサブフォルダを追加します。
- **SystemLink フィードにすべての依存項目を含める**—パッケージのすべての依存項目を SystemLink フィードにパブリッシュするか、パッケージのみを SystemLink フィードに追加するかを指定します。

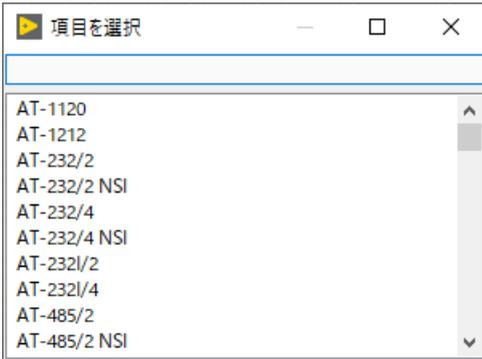
これらのオプションは Windows でのみ利用可能です。

環境の改善点

LabVIEW 2020 では、LabVIEW 環境が以下のように改善されました。

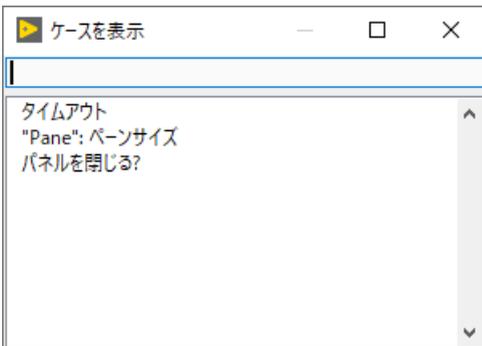
リスト項目の選択における改善

LabVIEW 2020 では、**項目を選択**ダイアログボックスを使用して、リング制御器、列挙体、または I/O 制御器または定数のリスト項目値を変更できます。フロントパネル上の制御器/定数、またはブロックダイアグラム上の定数を右クリックして、**項目を選択**を選択します。このダイアログボックスは、**項目を選択**ショートカットメニュー項目に代わるものです。



サブダイアグラム間の移動における改善

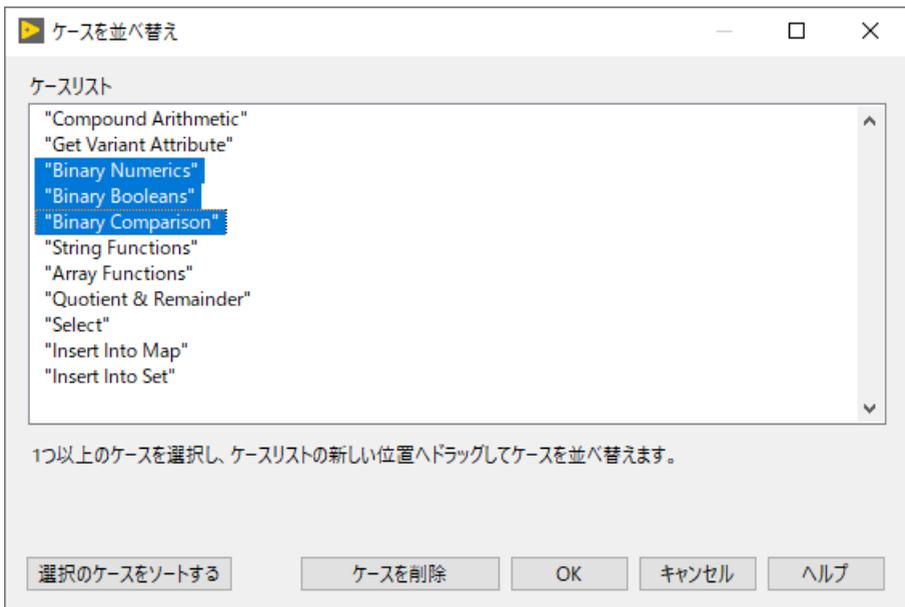
LabVIEW 2020 では、**ケースを表示**ダイアログボックスを使用して、ケース、イベント、または条件無効ストラクチャの表示フレームを変更できるようになりました。ストラクチャの枠線を右クリックし、**ケースを表示**を選択します。このダイアログボックスは、**ケースを表示**ショートカットメニュー項目に代わるものです。



ケースまたはサブダイアグラムの並べ替えにおける改善

ケースを並べ替えダイアログボックスの設計が変更され、新しく以下の機能が追加されました。

- 複数のケースを並べ替える
- 1つまたは複数のケースを削除する
- ケースリスト内の選択したケースのみを並べ替える
- ダイアログボックスのサイズを変更する



(NI ディスカッションフォーラムメンバー Intaris の提案による)

バリエーションデータタイプのデータをクリアする

🗨️ バリエーションデータタイプのデータをクリアするには、バリエーション制御器または表示器を右クリックして、**データ操作** → **データをクリア** を選択します。

(NI ディスカッションフォーラムメンバー attenbach の提案による)

エラー表示の改善

LabVIEW 2020 では、VI またはライブラリにエラーはないが、依存関係が壊れている場合のエラーの表示方法が変更されました。LabVIEW 2019 以前では、VI またはライブラリの直接の依存関係のエラーは「**エラーリスト**」ウィンドウに表示されます。エラーをダブルクリックすると、直接の依存関係が開きます。LabVIEW 2020 では、「**エラーリスト**」ウィンドウに「**根本原因: 依存項目が壊れています**」というエラーが表示されます。エラーをダブルクリックすると、エラーの原因である壊れた依存関係が開きます。

クラスタサイズダイアログボックスを起動する

🗨️ 右クリックしてメニューから選択する方法に加えて、「配列からクラスタに変換」関数をダブルクリックしても**クラスタサイズ**ダイアログボックスを開くことができるようになりました。

(NI ディスカッションフォーラムメンバー PalanivelThiruvankadam の提案による)

リング制御器の最後の項目を表示する

VIの実行中にユーザが項目を選択できないように、リング制御器の最後の項目を非表示にできるようになりました。最後の項目を表示するか、非表示にするかを切り替えるには、リング制御器を右クリックして、**実行時に最後のアイテムを表示/非表示**を選択します。また、**最後の値を非表示**プロパティを使用して、リング制御器の最後の項目の表示/非表示をプログラマ的に切り替えることもできます。

ダイアログボックスの改善点

LabVIEW 2020 では、LabVIEW Web サービスの設定を構成するための以下のダイアログボックスが追加および更新されました。

- **Web サービスプロパティ**ダイアログボックスでは、以下の点が変更されました。
 - **NI Web サーバ権限** (新規) — このページで、Web サービスの権限を作成および定義します。
 - **HTTP メソッド VI 設定** — **NI Web サーバ**タブ (新規) で、HTTP メソッド VI の呼び出しに必要な権限を構成します。アプリケーション Web サーバーに関連付けられた権限を設定する**セキュリティ**タブの名前が、**アプリケーション Web サーバ**に変更されました。
- **パッケージプロパティ**ダイアログボックスに、**Web サービス**が追加されました。このページを使用して、利用可能な Web サービスをパッケージに含めるように構成することができます。
- **アプリケーションプロパティ**ダイアログボックスの **Web サービス**ページに、**NI Web サーバ**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ことで、NI Web サーバを、スタンドアロンアプリケーションに含まれている Web サービスをホスティングする Web サーバとして指定することができるようになりました。
- LabVIEW 2020 では、NI Web Server または Application WebServer のどちらを使用するかに応じて、Web サービスへのアクセスを構成できるように、**オプション**ダイアログボックスの **Web サーバ**ページのオプションの構成を変更しました。

クラスプロパティダイアログボックスの**継承**ページでは、以下の点が変更されました。

- **親インタフェース**セクション (新規) — クラスが継承するインタフェースを定義します。
- **継承階層**セクションの名前が**親クラス階層**に変更されました。
- **継承の変更**オプションの名前が、**親クラスを変更**に変更されました。

「共有ライブラリをインポート」ウィザードの改善点

「共有ライブラリをインポート」ウィザードの **VI および制御器を構成**ページで、以下の新規オプションを使用して、複数のパラメータおよび関数に設定を適用できるようになりました。

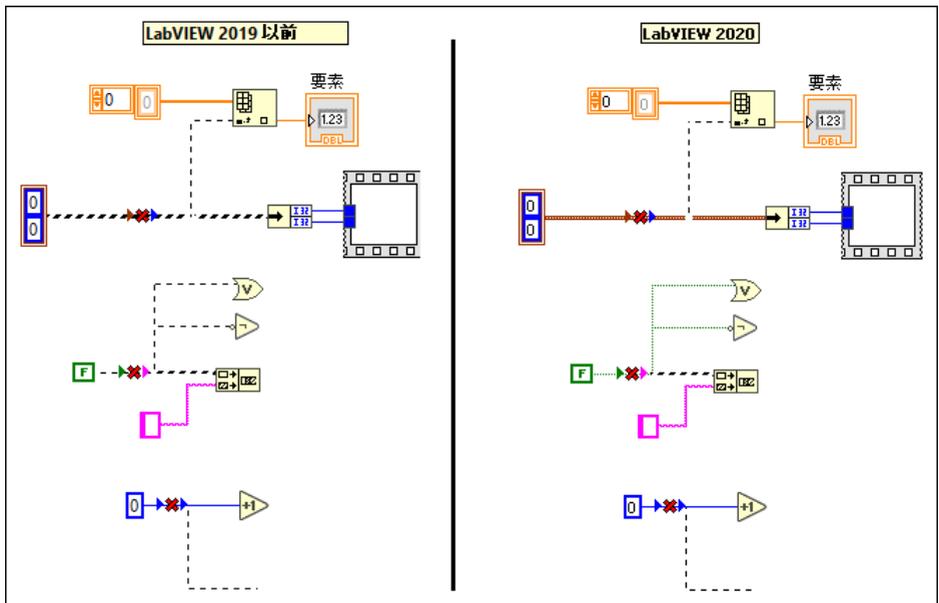
- **マッチするすべてのパラメータに適用**—現在のパラメータのタイプ設定が、ヘッダファイルの宣言に一致するすべての関数パラメータに適用されます。
- **すべてに適用**—現在の関数の呼び出し規約が共有ライブラリのすべての関数に適用されます。

ブロックダイアグラムの改善点

LabVIEW 2020 では、ブロックダイアグラムおよびそれに関連する機能が以下のように改善されました。

不良ワイヤの分岐をクリーンアップする

🗨️ LabVIEW の以前のバージョンでは、1つの分岐ワイヤが破損すると、ワイヤ全体が破損したように表示され、不良の分岐ワイヤを削除するとワイヤ全体が削除されました。LabVIEW 2020 では、不良の分岐ワイヤのみが破損していると表示されるようになりました。編集 → **不良ワイヤを削除** を選択するか、または<Ctrl-B>を押して不良ワイヤを削除すると、破損している分岐ワイヤのみが削除されるようになりました。すべての分岐ワイヤが破損している場合は、ワイヤ全体が削除されます。



(NI ディスカッションフォーラムメンバー altenbach の提案による)

イベントデータノードを非表示にする

🗨️ イベントストラクチャでは、すべてのデータ項目が未使用の場合に、イベントデータノードの表示/非表示を切り替えられるようになりました。ストラクチャの枠線を右クリックして、**表示項目 → このケースのイベントデータノード** を選択することで、イベントデータノードの表示/非表示を切り替えることができます。

(NI ディスカッションフォーラムメンバー Broken_Arrow の提案による。)

ループで反復端子を非表示にする

🗨️ For ループおよび While ループでは、反復端子の表示するか、非表示にするかを選択できるようになりました。反復端子の表示/非表示を切り替えるには、ループ枠を右クリックして、**反復端子**を選択します。反復端子に配線されている場合は、反復端子を非表示にすることはできません。

(NI ディスカッションフォーラムメンバー somebody_that_i_used_to_k の提案による。)

配線されている入力が 1 つのみの場合にワイヤの位置をスワップする

🗨️ LabVIEW の以前のバージョンでは、関数に 2 つの入力があり、両方の入力が配線されている場合にワイヤ位置をスワップできました。LabVIEW 2020 では、配線されている入力が 1 つのみの場合でも関数のワイヤ位置をスワップできるようになりました。ワイヤの位置をスワップするには、<Ctrl>キーを押しながら、配線されている入力をクリックします。

(NI ディスカッションフォーラムメンバー attenbach および tst の提案による)

アイコンにテキストを配置する

🗨️ LabVIEW 2020 では、**クイックドロップ**のキーボードショートカット<Ctrl-K>を使用してテキストを含むアイコンをすばやく作成できるようになりました。<Ctrl-Space>を押して<Ctrl-K>を押すと、VI のファイル名をテキストとしてアイコンに追加することができます。長すぎるテキストは切り捨てられます。

VI アイコンを右クリックして、**アイコンを VI 名に設定**を選択しても、同じことができます。

テキストは、<Ctrl-Space>を押して、アイコンに表示するテキストを入力してから<Ctrl-K>を押すことで変更できます。

(NI ディスカッションフォーラムメンバー tst の提案による)

クイックドロップでオブジェクトの詳細ヘルプを表示する

🗨️ **詳細ヘルプ**ウィンドウに、**クイックドロップ**ダイアログボックスで選択したオブジェクトの詳細ヘルプが表示されるようになりました。

(NI ディスカッションフォーラムメンバー elset191 の提案による)

クイックドロップから挿入されたオブジェクトの配線端子の改善

🗨️ **クイックドロップ**を使用して選択したワイヤに追加するオブジェクトのデフォルト配線が改善されました。**クイックドロップ**キーボードショートカット<Ctrl>を使用す

ると、オブジェクトは最も適切な端子に配線されます。以下の関数への配線が改善されました。

- 和、差、積、商などのバイナリ数値関数
- Or、Exclusive Or、Not And、Not Or、Not Exclusive Or、Implies などのバイナリブール関数
- 等しい?、等しくない?、大きい?、小さい?、以上?、以下?などのバイナリ比較関数
- 複合演算
- バリエーション属性を取得
- パターンで一致、文字列の検索と置換、文字列の検索と分割などの文字列関数
- 配列から削除、配列要素挿入などの配列関数
- 商&余り
- 選択
- マップに挿入
- セットに挿入

(NI ディスカッションフォーラムメンバー D*の提案による)

追加および変更された VI および関数

LabVIEW 2020 には、以下の VI および関数が新たに追加されました。VI、関数、およびノードについては、『LabVIEW ヘルプ』の「**VI と関数のリファレンス**」ブックを参照してください。

追加された VI および関数

Web サービス VI

Web サービスパレットの構成が変更され、以下のサブパレットが追加されました。

- アプリケーション Web サーバ—アプリケーション Web サーバにデプロイされる Web サービスをサポートする VI が含まれています。これには、組み込みサーバーページ (ESP) スクリプトを構成する VI、データ転送を暗号/復号化する VI、アプリケーション Web サーバで HTTP セッションを管理する VI が含まれています。
- NI Web サーバ—新しい「NI Web サーバの認証詳細を取得」VI が含まれています。この VI は、NI Web サーバの認証詳細を返します。

WebSockets VI

データ通信パレットには、Web ユーザインタフェースとの間でデータをストリーミングするための WebSockets VI をインストールするリンクが含まれています。

JKI VI Package Manager (VIPM) から WebSockets ツールキットをインストールするに

は、「データ通信」パレットで「**WebSockets アドオンをインストール**」をクリックします。**データ通信** → **WebSockets** パレットに WebSockets VI が表示されます。

(WebSockets アドオンの作成者である Sam Sharp 氏に感謝いたします。)

複数エラー VI

「ダイアログ&ユーザインタフェース」パレットに、「複数エラー」サブパレットが新たに追加されました。複数エラー VI を使用して、エラークラスタを異なる形式に変換したり、エラークラスタの属性を操作したりすることができます。

TLS (Transport Layer Security) 関数

「TCP」パレットに「TLS (Transport Layer Security)」パレットが追加されました。TLS プロトコルは、TCP (転送制御プロトコル) に似たインタフェースを提供し、TCP の上で暗号化され、認証されます。LabVIEW TLS は、TCP 関数の再利用をサポートします。LabVIEW は、TLS バージョン 1.2 をサポートしています。TLS 関数は、FPGA ターゲットをサポートしていません。

その他の新しい VI および関数

LabVIEW 2020 には、以下のその他の VI および関数が新たに追加されました。

- 以下の VI は、メッセージダイジェストアルゴリズムを計算できます。
 - 「上級ファイル」パレットの新しい「ファイルチェックサム」は、ファイルの内容のメッセージダイジェストを計算します。
 - 「Data Manipulation」パレットの新しい「Byte Array Checksum」VI は、バイト配列のメッセージダイジェストを計算します。

LabVIEW では、以下のメッセージダイジェストアルゴリズムを計算できます。

- SHA-256
- SHA-224
- SHA-512
- SHA-384
- SHA-512/256
- SHA-512/224
- SHA3-224
- SHA3-256
- SHA3-384
- SHA3-512
- メモリ管理パレットに新たに追加された「メモリステータスを取得」VI は、LabVIEW プロセスが使用している物理メモリ量を返します。この VI は、システム全体のメモリ使用量のステータスも返します。

- その他の文字列パレットに新しく追加された「NI GUID を作成」VI は、GUID (グローバル意識別子) 文字列を生成します。
- 数値パレットに新しく追加された「タイプの範囲制限」VI は、入力データタイプの最大値と最小値を返します。
- Conversion パレットに新しく追加された「列挙体から列挙体配列」VI は、列挙体の値の数と同じ数の要素をもつ列挙体の配列を作成します。
- データタイプ解析パレットに新しく追加された「LabVIEW クラスの親とメンバー VI の情報を取得」VI は、バリエーションに格納されている LabVIEW クラスまたはインタフェースの親とメンバー VI の情報を取得します。

変更された VI および関数

LabVIEW 2020 では、以下の VI および関数に変更されました。

- パス作成関数の**名前または相対パス**入力のデフォルトデータタイプは文字列からパスに変更されます。この入力から制御器または定数を作成すると、データタイプはパスになります。それでも、この入力に文字列を渡すこともできます。
- 「より特定のクラスに変換」および「より一般的なクラスに変換」関数は、インターフェースのサポートを追加します。これらの関数を使用して、ワイヤのクラスタイプまたはインタフェースタイプを別のクラスタイプまたはインタフェースタイプに型変換できます。
- 「同一または派生タイプをアサート」VI は、インターフェースのサポートを追加します。この VI は、入力データが入力クラスまたはインタフェースの派生と同じでないか、入力クラスの派生クラスでない場合に、発呼者 VI を壊します。
- 「親メソッド呼び出し」ノードは「親クラスメソッド呼び出し」に名前が変更されました。クラスメソッドの最も近い先祖の実装を呼び出すには、このノードを使用します。
- 「TDMS ファイルビューア」VI および「TDMS ファイルビューア」ダイアログボックスが、より直感的なデザインに変更されました。複数のタブを切り替えなくても、(.tdms) ファイルのデータを表示したり、視覚化の設定を変更することが、すべて 1 つのページで行えるようになりました。
- 旧バージョンの LabVIEW では、「エラー結合 Merge Errors」関数は、常に、入力エラークラスタ内の最初のエラーまたは警告を返しました。LabVIEW 2020 では、この関数を右クリックして、**すべてのエラーを保持**を選択することで、すべてのエラー/警告を結合したエラークラスタを返すように構成できるようになりました。

追加および変更されたプロパティ、メソッド、およびイベント

LabVIEW 2020 では、以下のプロパティ、メソッドおよびイベントが追加または変更されました。

新規のプロパティ、メソッド、およびイベント

LabVIEW 2020 では、以下のプロパティ、メソッドおよびイベントが追加されました。

- LV クラスライブラリクラスに以下のプロパティおよびメソッドが追加されました。
 - 親インタフェース
 - 親ライブラリ
 - インタフェース
 - 親インタフェースを書き込み
 - 親ライブラリパスを書き込み
 - 親インタフェースを追加
 - 親インタフェースを削除
- アプリケーションクラスに以下のプロパティおよびメソッドが追加されました。
 - パレット:制御器名
 - パレット:関数名
 - LabVIEW クラス:インタフェースを作成
- VI クラスに以下のプロパティおよびメソッドが追加されました。
 - 実行:ワイヤ値を保持
 - 制御 VI 変更を適用
- G オブジェクトクラスに以下のプロパティが追加されました。
 - グループ化済み
 - グループメンバーリファレンス()
 - ロック済み
- 配列クラスと配列定数クラスに以下のプロパティが追加されました。
 - 指標境界
- パスクラスに以下のプロパティが追加されました。
 - 参照ボタン境界
- リングクラスに以下のプロパティが追加されました。
 - 最後の値を非表示
 - 特色
- ツリー制御クラスに以下のプロパティが追加されました。
 - フォーカス項目
- 複数列リストボックスクラスに以下のプロパティが追加されました。
 - 選択したセル
 - バルクセル:フォント色
 - バルクセル:背景色

- プロジェクトクラスに以下のメソッドが追加されました。
 - 項目 ID から項目を取得
- 数値クラス、数値定数クラス、上書き可能なパラメータ端子クラスに、数値制御器と定数の固定小数点動作を構成するためのプロパティと、簡単な計算の固定小数点出力構成用のプロパティが追加されました。
- リングクラス、列挙体クラス、コンボボックスクラスに、以下のイベントが追加されました。
 - 操作メニュー起動
 - 操作メニュー破棄
- 制御器クラスに以下のイベントが追加されました。
 - ショートカットメニュー破棄

変更されたメソッド

LabVIEW 2020 では、以下のメソッドが変更されました。

- LV クラスライブラリ:実装あり?メソッドの**親クラスを考慮**パラメータの名前が**親を考慮**に変更されました。クラスが親インタフェースから同じ名前の複数のメソッドを継承し、独自のオーバーライドを提供しない場合、このメソッドはエラーを返します。
- ライブラリ:ソーススコープ設定および伝播メソッドに、新しく**プロンプトをスキップ?**パラメータが追加されました。項目がダイナミックディスペッチ VI である場合に、アクセススコープ設定をすべてのオーバーライド VI に適用する前に確認するかどうかを指定します。

旧バージョンの LabVIEW の機能と変更点

お使いのバージョン以降にリリースされた LabVIEW の各バージョンの新機能を特定するには、それらのバージョンのアップグレードノートを参照してください。これらのドキュメントにアクセスするには、NI Web サイト (ni.com/info) で以下のリストから適切な LabVIEW バージョン用の Info Code を入力してください。

- LabVIEW 2016 アップグレードノート—[upnote16jp](#)
- LabVIEW 2017 アップグレードノート—[upnote17jp](#)
- LabVIEW 2018 アップグレードノート—[upnote18jp](#)
- LabVIEW 2019 アップグレードノート—[upnote19ja](#)

情報は事前の通知なしに変更されることがあります。NI の商標の詳細については、ni.com/trademarks の NI Trademarks and Logo Guidelines (英語) を参照してください。本書中に記載されたその他の製品名及び企業名は、それぞれの企業の商標又は商号です。NI の製品及び技術を保護する特許については、ソフトウェアで参照できる特許情報 (ヘルプ>特許)、メディアに含まれている patents.txt ファイル、又は ni.com/patents からアクセスできる National Instruments Patent Notice のうち、該当するリソースから参照してください。エンドユーザ使用許諾契約 (EULA) 及び他社製品の法的注意事項はご使用の NI 製品の Readme ファイルにあります。NI の輸出関連法規遵守に対する方針については、また必要な HTS コード、ECCN (Export Control Classification Number)、その他の輸出入に関する情報の取得方法については、「輸出関連法規の遵守に関する情報」(ni.com/legal/ja/export-compliance) を参照してください。NI は、本書に記載の情報の正確性について、一切の明示又は黙示の保証を行わず、技術的な誤りについて一切の責任を負いません。米国政府のお客様へ: 本書に含まれているデータは、民間企業の費用により作成されており、民間機関用の連邦調達規則 52.227-14 と軍事機関用の国防省連邦調達規則補足 252.227-7014 および 252.227-7015 に基づく限定権利及び制約付データ権利の条項の適用を受けます。

© 1998–2020 National Instruments. All rights reserved.

371780T-0112 2020年3月16日