LabVIEW™

バージョン 6.1

このアップグレードノートでは、Windows、Macintosh、UNIX 対応 LabVIEW をバージョン 6.1 にアップグレードするプロセスについて説明します。

インストール手順およびその他の重要情報については、『LabVIEW リリースノート』をご覧ください。

アップグレードノートについて

本書では、LabVIEW 6.1 へのアップグレード時に発生する可能性のある問題や新機能について説明します。

詳細については、

LabVIEW 6.1 の機能については、『LabVIEW ユーザマニュアル』および「LabVIEW ヘルプ」を参照してください。「LabVIEW ヘルプ」にアクセスするには、**ヘルプ→ヘルプを表示**を選択します。PDF 形式の『LabVIEW ユーザマニュアル』および他の LabVIEW マニュアルを参照するには、**ヘルプ→印刷版マニュアルを表示**を選択します。ハードコピー版の LabVIEW マニュアルは、LabVIEW 6.1 では改訂されません。最新の PDF 版 LabVIEW マニュアルについては「LabVIEW ライブラリ」を参照してください。PDF を表示するには、Adobe Acrobat Reader 4.0 以降がインストールされている必要があります。Acrobat Reader をダウンロードするには、アドビシステムズ社のウェブサイト `www.adobe.com` にアクセスしてください。

目次

322784B-01

2002 年 01 月版

アップグレード時の問題点

LabVIEW 5.*x* からアップグレードする場合は、「VI を変換する」、「Applib とツールセットをアップグレードする」、および「LabVIEW 5.x からアップグレードする」の各セクションをお読みください。

LabVIEW 4.*x* 以前のバージョンからアップグレードする場合は、「VI を変換する」、「Applib とツールセットをアップグレードする」、および 「LabVIEW 4.x からアップグレードする」の各セクションをお読みください。

VI を変換する

LabVIEW アプリケーションのアップグレードプロセスは自動化されています。LabVIEW 4.0 以降のバージョンで保存された VI を開くと、LabVIEW 6.1 によってその VI の変換とコンパイルが自動的に行われます。VI を閉じるときは、LabVIEW 6.1 で保存してください。そうしないと、その VI にアクセスするたびに変換処理が行われ、システムのメモリリソースが使用されます。

メモ LabVIEW 6.1 で保存した VI は、旧バージョンの LabVIEW ではロードできません。以前のバージョンの LabVIEW で実行できるようにするためには、**ファイル→オプション付き保存**を選択し、**前回用に保存**をクリックして VI を保存します。

VI の変換に必要なメモリ量は、VI やサブ VI のすべてが使用するディスク上のメモリ量を合計することによって予測できます。VI が VI ライブラリにある場合、その VI は圧縮されているため、VI ライブラリのサイズのおよそ 30 パーセント分を加算します。LabVIEW での変換プロセスには、最低でも、上記のメモリに 3 MB を足した量のメモリが必要です。

VI すべてを一度に変換するのに十分なメモリがコンピュータにない場合は、VI を段階的に変換します。変換する VI の階層を調べ、下位階層にあるサブ VI をロードして保存することから開始します。その後、徐々にその上位階層に進んでください。**ツール→上級→一括コンパイル**を選択して VI のディレクトリを変換することもできます。しかし、このオプションではディレクトリまたは VI ライブラリ内の VI はアルファベット順に変換されます。変換プロセスで上位の VI が先に処理される場合、**一括コンパイル**では、上位の VI を先に開いた場合とおよそ同じ量のメモリが必要になります。

ヘルプ→ LabVIEW についてを選択してメモリの使用状況を監視し、それまでに使用したメモリ量の合計を表示できます。

Applib とツールセットをアップグレードする

既存のツールセットの多くは、LabVIEW 6.1 で動作します。ただし、LabVIEW 6.1 で使用するには、VI を一括コンパイルする必要があります。VI の一括コンパイルの詳細については、このドキュメントの「VI を変換する」セクションを参照してください。LabVIEW 6.1 は、LabVIEW 4.0 以降向けに設計されたツールセットと互換性がありますが、いくつかの例外があります。

- **（LabVIEW 開発システム）LabVIEW アプリケーションビルダ：**
  LabVIEW アプリケーションビルダ 6.1 にアップグレードする必要があります。LabVIEW プロフェッショナル開発システム 6.1 には、アップグレード済みのアプリケーションビルダライブラリが含まれています。

- **（LabVIEW 開発システム）LabVIEW プロフェッショナル G 開発ツールキット：**プロフェッショナル G 開発ツールキット 5.0 以降のバージョンをご使用の場合は、LabVIEW プロフェッショナル開発システム 6.1 にアップグレードする必要があります。プロフェッショナル G 開発ツールキット 5.1 のユーザに対しては無料でアップグレードします。LabVIEW プロフェッショナル開発システム 6.1 には新しいバージョンのプロフェッショナル G 開発ツールキット 6.1 が含まれています。
- **LabVIEW テスト / 検査システム（テストエグゼクティブ）：**LabVIEW テスト / 検査システム 5.1 以前のバージョンをご使用の場合は、LabVIEW 6.1 で使用できるようにこれらの VI を一括コンパイルする必要があります。VI の一括コンパイルの詳細については、このドキュメントの「VI を変換する」セクションを参照してください。

旧バージョンの LabVIEW をアップグレードする

以下のセクションでは、LabVIEW のバージョンによって異なるアップグレードおよび互換性の問題点について説明します。

LabVIEW 6.0 からのアップグレード

このセクションでは、LabVIEW 6.0 から LabVIEW 6.1 にアップグレードする場合に発生する可能性のある問題点について説明します。

強制ドットとタイプ定義

ワイヤには、タイプ定義の情報が含まれるため、ブロックダイアグラム上でより多くの強制ドットが見られることがあります。タイプ定義制御器を、タイプ定義端子以外の VI や関数端子に配線すると、強制ドットが表示されます。タイプ定義である出力端子をタイプ定義でない表示器に配線したときにも、強制ドットは表示されます。そのような強制ドットは、VI でのタイプ定義の使用が一貫していないことを示します。

この場合、強制ドットはランタイム時の性能に影響しません。

メモ 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。

Control Online Help 関数

Control Online Help 関数の**ヘルプファイルへのパス**入力が必要です。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、入力に配線することができます。コンパイル済みのヘルプファイル名のみを配線した場合、LabVIEW は `labview\help` ディレクトリ内でそのファイルを検索します。

技術サポートのフォーム

LabVIEW インストーラでは、techsup.llb はインストールされません。インストール、構成、およびアプリケーションに関する問題や質問については、ni.com/support/ja の技術サポートのセクションを参照してください。

LabVIEW 5.*x* からアップグレードする

このセクションでは、LabVIEW 5.*x* から LabVIEW 6.1 にアップグレードする場合に発生する可能性のある問題点について説明します。

データログファイルを変換する

旧バージョンの LabVIEW で作成したデータログファイルを開くと、LabVIEW 6.1 では、そのファイルを LabVIEW 6.1 フォーマットに変換するように指示されます。変換を選択すると、LabVIEW によって、データログファイルが新しいフォーマットに変換されたデータと置き換えられます。ファイルを変換しないとエラーが返され、ファイルは開きません。

データログファイルを開いたときに自動的に変換するには、LabVIEW 環境設定ファイルに次の行を追加します。

```
silentDatalogConvert=True
```

(Macintosh) 次の行を追加します。

```
silentDatalogConvert:True
```

(UNIX) 次の行を追加します。

```
labview.silentDatalogConvert:True
```

データログファイルを開いたときに自動的に変換されないようにするには、環境設定を False に設定します。

LabVIEW 5.*x* サーバと LabVIEW 6.1 クライアント間の互換性の問題

LabVIEW 6.1 クライアントを LabVIEW 5.*x* アプリケーションの VI サーバに接続しようとするとエラーになります。これは、LabVIEW 5.*x* のアプリケーションでは LabVIEW 6 VI サーバプロトコルの新しい機能を認識できないためです。

LabVIEW 5.*x* クライアントから LabVIEW 6.1 アプリケーションの VI サーバに接続することも可能です。

UDP 関数

関数→通信→ UDP パレットにある UDP 関数を使用します。UDP VI は、互換性のある VI として vi.lib_oldvers_oldvers.llb に入っています。

LabVIEW 4.*x* からアップグレードする

このセクションでは、LabVIEW 4.*x* から LabVIEW 6.1 にアップグレードする場合に発生する可能性のある問題点について説明します。

LabVIEW 4.*x* から（LabVIEW 4.*x* へ）ブールデータを変換する

ブールデータの格納形式は、LabVIEW 4.*x* と LabVIEW 5.*x* の間で変更されました。LabVIEW 4.*x* では、データが配列ではない場合、ブールデータは 2 バイトで格納されます。データが配列の場合、LabVIEW 4.*x* では各ブール要素は 1 ビットで格納されます。LabVIEW 6.1 では、それが配列かどうかに関わらず、1 つのブール値は 1 バイトで格納されます。この変更により、ブロックダイアグラム上でブール値の配列に対応した関数が増え、これらの配列の動作が数値の配列の動作と一貫性のあるものになります。新しいブール値のデータ形式は、コードインタフェースノード (CIN) でのデータ操作に影響しますが、LabVIEW 6.1 には既存の CIN との互換性があります。

1 つまたは複数のブール値を含んでいるバイナリデータを LabVIEW 4.*x* のファイルに書き込む場合、その形式は LabVIEW 6.1 で同じデータを書き込む場合とは異なります。LabVIEW 6.1 には、LabVIEW 4.*x* で書き込まれたバイナリデータを読み取り、LabVIEW 4.*x* で読み取ることのできるバイナリデータを書き込むメカニズムが備わっています。Write File、Read File、Type Cast、Flatten To String、および Unflatten From String の 5 つの関数には、**4.*x* データの変換**ショートカットメニュー項目があります。関数を右クリックしてこのメニュー項目を選択すると、この関数は、バイナリデータを LabVIEW 4.*x* で書き込まれたデータのように扱うことができます。LabVIEW 4.*x* 用にフォーマットされたデータを作成するには、Write File、Flatten to String、または Type Cast 関数を使用します。LabVIEW 4.*x* 用にフォーマットされたデータを読み込むには、Read File、Unflatten From String、または Type Cast 関数を使用します。**4.*x* データの変換**ショートカットメニュー項目を選択すると、LabVIEW 6.1 ではその関数上に赤い `4.x` を描画して、データを LabVIEW 4.*x* 形式に、あるいは LabVIEW 4.*x* 形式から変換中であることを示します。データを変換しないようにするには、**4.*x* データの変換**ショートカットメニュー項目をもう一度選択してチェックマークを外します。

ブール値が含まれているデータファイルが複数ある場合、これらのファイルを開き、LabVIEW 6.1 で認識できる新しいデータファイルにデータを書き込む VI を作成できます。

LabVIEW 6.1 で、LabVIEW 4.*x* 以前のバージョンで保存した VI をロードすると、Write File、Read File、Type Cast、Flatten To String、および Unflatten From String の各関数上に **4.*x* データの変換**属性が自動的に設定されます。これらの関数は、以前と同じように機能します。VI で LabVIEW 6.1 のブールデータ形式を使用する必要がある場合は、各関数で **4.*x* データの変換**ショートカットメニューを選択します。通常、VI で、

旧バージョンの LabVIEW で書き込まれたブールデータを含むファイルを操作する必要がない場合や、旧バージョンの LabVIEW で実行している VI とブールデータをやり取りする必要がない場合は、LabVIEW 6.1 ブールデータ形式を使用してください。今後の LabVIEW バージョンでは、今までのブールデータ形式はサポートされない可能性があります。

データログファイルを変換する

アップグレード時のデータログファイルの変換方法の詳細については、このドキュメントの「データログファイルを変換する」セクションを参照してください。

VI Control VI

VI Control VI は、デフォルトのパレットビューではなく、互換性 VI として `vi.lib\utility\victl.llb` に入っています。したがって、VI Control VI の代わりに、**関数→アプリケーション制御**パレットにある、Open VI Reference、Call By Reference、Property Node、Invoke Node の各 VI サーバ関数を使用します。

VI Control VI から返されるエラーコードには、LabVIEW 6.1 で変更されたものもあります。旧バージョンの LabVIEW では、VI Control VI はエラーコード 7 および 1000 を返しました。LabVIEW 6.1 の VI Control VI は、1004 と 1003 というコードを返します。LabVIEW 4.*x* で作成された VI がエラーコード 7 および 1000 かどうかをチェックする場合、その VI を LabVIEW 6.1 で実行するには VI に変更を加える必要があります。

DDE VI

(Windows) DDE VI は、デフォルトのパレットビューにはありませんが、互換性 VI として `vi.lib\platform\dde.llb` に入っています。

LabVIEW 3.*x* 以前のバージョンからアップグレードする

LabVIEW 3.*x* 以前からのアップグレードについては、ナショナルインスツルメンツのウェブサイト（`ni.com/jp`）を参照してください。

LabVIEW 6.1 の機能

LabVIEW 6.1 の機能については、『LabVIEW ユーザマニュアル』および「LabVIEW ヘルプ」を参照してください。

フロントパネルをリモートで表示し制御する

組み込まれている LabVIEW ウェブサーバに接続することにより、LabVIEW 内またはウェブブラウザ内からリモートでフロントパネルを表示したり制御したりすることができます。クライアントからフロントパネ

ルをリモートで開くと、ウェブサーバはフロントパネルをクライアントに送りますが、ブロックダイアグラムやすべてのサブ VI はサーバコンピュータ上に残ります。VI がクライアント上で実行しているのと同じように、フロントパネルを操作することができますが、ブロックダイアグラムはサーバ上で実行します。フロントパネル全体をパブリッシュしたり、リモートアプリケーションを安全に、すばやく、簡単に制御するには、この機能を使用します。

イベントドリブンプログラミング

アプリケーションでイベントを処理するには、イベントストラクチャを使用します。ケースストラクチャと同様、イベントストラクチャに複数のイベントケースを追加することができます。そして、1 つまたは複数のイベントを処理するようにそれらのイベントケースを構成できます。イベントが発生すると、LabVIEW は対応するイベントケースを実行します。イベントを使用して、ユーザが値を変更したり、フロントパネルを閉じたり、アプリケーションを終了したときなどを検出することができます。イベントストラクチャを使用してイベント特定のコードを実行することにより、ユーザが行ったアクションを検出するためにブロックダイアグラムがフロントパネルをポーリングする必要が少なくなります。それにより、処理時間が短縮され、ブロックダイアグラムが簡素化されます。

自動ツール選択

フロントパネルやブロックダイアグラム上のオブジェクトにカーソルを移動すると、LabVIEW は**ツール**パレットの中から対応するツールを自動的に選択します。自動ツール選択を切り替えるには、**ツール**パレットの**自動ツール選択**ボタンをクリックするか、<Shift-Tab> キーを押します。自動ツール選択を無効にするには、ツールパレットからツールを手動で選択するか、<Tab> キーを押してパレットの次のツールに移動します。

新規または変更されたショートカットキーを以下に示します。

- 自動ツール選択を有効にしたり無効にするには、<Shift-Tab> キーを押します。
- 自動ツール選択が有効になっている場合に、オブジェクト上にカーソルを移動して <Ctrl> キーを押すと、次に利用する可能性の高いツールが選択されます。**(Macintosh)** <Option> キーを押します。**(UNIX)** <Meta> キーを押します。
- 自動ツール選択が有効になっているときに、一時的にスクロールツールに切り替えるには、フロントパネルまたはブロックダイアグラムの空いている領域にカーソルを移動して <Shift-Ctrl> キーを押します。**(Macintosh)** <Shift-Option> キーを押します。**(UNIX)** <Meta-Shift> キーを押します。
- 自動ツール選択が有効になっていて他のツールが選択されているときに、一時的に位置決めツールに切り替えるには、<Shift> キーを押します。

- 自動ツール選択が有効になっているときに、ラベルのテキストやブロックダイアグラムの定数の値を編集するには、ラベルまたは定数をダブルクリックします。
- ワイヤを設定した最後のポイントを取り消すには、<Shift> キーを押してブロックダイアグラム上の任意の場所をクリックします。
- ノードの角をドラッグして、ノードのサイズを変更することはできません。ノードに端子を追加するには、上または下の枠線をドラッグします。
- フロントパネルまたはブロックダイアグラムにフリーラベルを配置するには、<Ctrl> キーを押して任意の空き領域をダブルクリックします。**(Macintosh)** <Option> キーを押します。**(UNIX)** <Meta> キーを押します。

サンプル VI を検索する

「サンプルの検索」はサンプルの検索ツールに代わりました。**ヘルプ→サンプルの検索**を選択するか、LabVIEW の起動時に表示される最初のダイアログボックスで**サンプルの検索**ボタンをクリックすると、表示することができます。このツールを使用すると、インストールされているサンプル VI を機能別またはディレクトリ構造別に参照したり、VI の説明を参照したりすることができます。また、サンプルをキーワードで検索することもできます。

LabVIEW データタイプを XML に変換する

関数→上級→データ操作パレットにある Flatten to XML 関数を使用して、LabVIEW データタイプを LabVIEW XML スキーマに基づいて XML 形式に変換します。XML 形式のデータタイプを LabVIEW データに変換するには、同じパレットにある Unflatten from XML 関数を使用します。

メモ LabVIEW では、そのような変換に、`labview\help\LVXMLSchema.xsd` で説明されている定義済みの XML スキーマを使用します。

カラーピッカーの機能向上

カラーピッカーには、ユーザ定義カラーとシステムカラー各種、および最近使用したカラーの履歴が含まれていて、アプリケーション内の VI 間で統一のとれたカラースキームを維持することができます。フロントパネルの外観を VI を動作するコンピュータのシステムカラーに合わせるには、システムカラーを使用します。カラーピッカーの機能が向上したため、**ウィンドウの外観をカスタマイズする**ダイアログボックスの**パネルにシステムカラーを使用**オプションは削除されています。その代わりに、フロントパネルオブジェクトに色をつけるには、カラーピッカーのシステムカラーを使用します。

カラーピッカーの中の色の上にカーソルを置くと、その色の RGB 値が表示されます。透明（**T**）ボックスは、カラーピッカーの右上にあります。

VI を HTML ファイルまたはコンパイル済みのヘルプファイルにリンクする

VI から HTML ファイルやコンパイル済みのヘルプファイルにリンクすることができます。それを実行するには、**ファイル→ VI プロパティ**を選択して、プルダウンメニューから**ドキュメント**を選択します。**ヘルプのパス**に、.htm、.html、HTML Help（.chm）、または WinHelp（.hlp）ファイルのファイル名を入力できます。**ヘルプのパス**に .chm ファイル名が含まれている場合、**ヘルプタグ**は索引のキーワードか、HTML ヘルププロジェクトの個々の HTML ファイルのファイル名になります。

また、Control Online Help 関数を使用して、VI を .htm ファイルや .html ファイルにリンクしたり、索引のキーワードや HTML ファイル名を使用して HTML ヘルプファイルにリンクすることができます。

ケーススストラクチャの機能向上

すべてのケースに対してケーススストラクチャの出力トンネルを配線する必要がなくなりました。出力トンネルを右クリックしてショートカットメニューから**未配線の場合はデフォルトを使用**を選択すると、未配線のすべてのトンネルのトンネルデータタイプに対しデフォルト値を使用することができます。

アプリケーションビルダの機能向上

本バージョンの LabVIEW で向上したアプリケーションビルダ機能については、『アプリケーションビルダリリースノート』を参照してください。

タブ制御器の機能向上

タブ制御器のタブを右クリックして、**上級**ショートカットメニューで次のオプションのいずれかを選択します。

- **カスタマイズする：**制御器エディタを使用してカスタムタブ制御器を作成する場合にこのオプションを選択します。このオプションを選択してカスタマイズできるのはタブ制御器のみで、タブ制御器ページ上のオブジェクトはカスタマイズできません。
- **複数色を許可：**タブ制御器の各ページで異なる色を使用する場合にこのオプションを選択します。
- **タブレイアウト：**タブ制御器の各ページのタブにテキストや画像を使用する場合にこのオプションを使用します。タブの画像の追加や削除にもこのショートカットメニューを使用します。
- **タブの位置：**タブ制御器上でタブの位置を変更する場合にこのオプションを選択します。
- **(Windows)** タブ制御器ページに ActiveX コンテナを配置することができます。

タブ制御器をプログラム的に構成するには、次のプロパティを使用します。

- **ページラベル：**このプロパティを使用して、タブ制御器ページのラベルを読み取ります。
- **タブ キャプション：**このプロパティを使用して、タブ制御器ページのキャプションのテキストを取得したり設定したりします。
- **独自のラベル：**このプロパティを使用して、タブ制御器ページのキャプションがページのラベルと連動しないようにします。このプロパティを TRUE に設定すると、タブキャプションプロパティを使用してページキャプションを変更することができます。
- **複数色を許可**および**色：**これらのプロパティを使用して、タブ制御器と個々のページの色を設定します。
- **ページの装飾体**および**ページのオブジェクト：**これらのプロパティを使用して、タブ制御器ページのすべての装飾体へのリファレンスを取得したり、各ページのすべての制御器、表示器、および装飾体へのリファレンスを取得したりします。
- **説明**および**ヒントラベル：**これらのプロパティを使用して、各ページの説明とヒント情報を変更します。

キューおよびノーティファイアを使用する

Notification VI と Queue VI の代わりに、新たに Notifier Operations 関数と Queue Operations 関数が**上級→同期**パレットに追加されました。以前のバージョンの VI は、文字列データのみ受け付けていました。新しい関数は多形性で、どのようなタイプのデータも受け入れます。また、新しい関数は、使いやすく、実行速度もやや速くなり、構築されたアプリケーションでの消費メモリが少なくてすみます。Notifier Operations 関数と Queue Operations 関数のサンプルについては、`examples\general\notifier.llb` および `examples\general\queue.llb` を参照してください。

ポイントバイポイント VI

ポイントバイポイント VI は、LabVIEW 開発システムおよびプロフェッショナル開発システムに含まれています。これらの VI は**関数→解析→ポイントバイポイント**パレットにあります。これらの VI を使用すると、データを 1 ポイントずつ効率的に都合よく処理することができます。この VI は、各データサンプルが利用可能になると結果を生成し、その間フィルタのような動作をします。ポイントバイポイント VI は、連続するデータポイントを集録している間に実際のデータ解析を行います。これらの VI の詳細については、「LabVIEW ライブラリ」にある『LabVIEW 入門　ポイントバイポイント VI』の PDF を参照してください。

波形測定 VI

関数→解析→波形測定パレットにある Amplitude and Levels、Cycle Average and RMS、Pulse Measurements、および Transition Measurements の各 VI は新しい VI です。これらの VI は、波形の時間領域形状の特徴を指定するために使用します。これらの VI に関する詳細は、「LabVIEW ヘルプ」を参照してください。

フロントパネルの動作を監視する

ユーザがボタンをクリックしたり、ノブを回したり、データを入力したりして、ユーザがフロントパネルオブジェクトの値を変更した後にのみブロックダイアグラムが実行するように設定するには、**関数→時間 & ダイアログ**パレットにある Wait For Front Panel Activity 関数を使用します。この関数を使用すると、フロントパネル上でのユーザによる操作が検出されたときにブロックダイアグラムが実行されます。この関数は、Occurrence 関数に似ています。

この関数を使用することにより、フロントパネルオブジェクトの値に変更があったかどうかを確認するために、頻繁にフロントパネルをポーリングする必要がなくなります。

メモ この関数を使用して、マウスのクリックやキー操作などのフロントパネルイベントをプログラム的に処理することはできません。フロントパネルのイベントをプログラム的に処理するには、イベントストラクチャを使用します。

ワイヤレスデバイスとともに LabVIEW を使用する

(Windows) 別々のコンピュータ上で実行している VI 間にワイヤレス通信リンクを確立するには、**関数→通信→ IrDA** パレットにある IrDA 関数を使用します。赤外線によってデータを送信するプロトコルである IrDA テクノロジーを使用して、リモートコンピュータにデータを送ったりリモートコンピュータからデータを読み取ったりする VI を作成することができます。このテクノロジーを利用するには、IrDA デバイスがクライアントコンピュータとサーバコンピュータに接続されている必要があります。

メモ IrDA テクノロジーは、携帯型コンピュータでよく使用されています。ただし、現在のところ、PalmOS や Wndows CE などのハンドヘルドコンピュータで使用されているオペレーティングシステムでは、LabVIEW は利用できません。IrDA テクノロジーおよび LabVIEW での使用方法の詳細については、アプリケーションノートの「Using LabVIEW with Wireless Devices」を参照してください。

ActiveX オートメーションのカスタムインタフェースのサポート

LabVIEW を使って ActiveX サーバからプロパティやメソッドにアクセスする ActiveX クライアントを作成している場合は、サーバにより公開されるカスタムインタフェースにアクセスできます。ActiveX サーバを作成

する際には、それらのカスタムインタフェースのプロパティおよびメソッドのパラメータがオートメーション（IDispatch）データタイプであることを確認してください。カスタムインタフェースへのアクセスに関する詳細は、サーバ開発プログラミング環境のマニュアルを参照してください。

表および複数列リストボックスでの行と列の挿入と削除

表に行や列を挿入するには、新しい行か列を挿入する部分を右クリックして、ショートカットメニューから**データ操作→前に行を挿入**または**前に列を挿入**を選択します。行または列を削除するには、その行または列を右クリックして、ショートカットメニューから**データ操作→行を削除**または**列を削除**を選択します。

複数列リストボックスに行や列を挿入するには、新しい行か列を挿入する部分を右クリックして、ショートカットメニューから**前に行を挿入**または**前に列を挿入**を選択します。行または列を削除するには、その行または列を右クリックして、ショートカットメニューから**行を削除**または**列を削除**を選択します。

グラフィックス機能の向上

LabVIEW では、フロントパネル、ブロックダイアグラム、ブール制御器および画像リング制御器における動画 GIF ファイルをサポートしています。システムがサポートしているグラフィック形式に対して使用するのと同じメソッドで、動画 GIF ファイルを LabVIEW にインポートします。LabVIEW はまた、透明の動画を含め、MNG、動画 MNG、PNG、および動画 PNG グラフィック形式をサポートします。

Run VI メソッド

Run VI メソッドに **Auto Dispose Ref** という新しいパラメータが追加されました。デフォルトは FALSE です。**Auto Dispose Ref** が TRUE の場合、ターゲット VI はリファレンスをメイン VI から切り離します。ターゲット VI の実行が終了すると、LabVIEW はパラレルデータスペースとともにリファレンスを自動的に破棄します。Run VI メソッドの使用例については、`examples\viserver\runvi.llb` を参照してください。

Open VI Reference 関数

Open VI Reference 関数の**オプション**パラメータには、新しいオプションビット **0x08 - Prepare for Reentrant Run** があります。この新しいオプションは、ターゲット VI を保留モードにして、ターゲット VI が再入可能な場合のみパラレルデータスペースを割り当てます。この新しいオプションの使用例については、`examples\viserver\runvi.llb` を参照してください。

旧バージョンで保存する

LabVIEW 6.1 では、**オプション付き保存**を選択して、VI を LabVIEW 6.0 用に保存することができます。VI を LabVIEW 5.*x* 用に保存する場合は、VI を LabVIEW 6.0 で開いてから**ファイル→オプション 付き保存**を選択する必要があります。

ソースコード管理ツールの機能向上

ソースコード管理プルダウンメニューに、一般的なソースコード管理操作が追加されました。LabVIEW は、メニュー項目を起動する VI で各操作を実行します。旧バージョンのダイアログボックスは 1 つのインタフェースに統合され、そこでファイルをチェックインしたり、プロジェクトを編集したり、ローカル構成を変更したりすることができます。

LabVIEW は、Perforce と Microsoft Visual SourceSafe という 2 つのサードパーティソースコード管理システムをサポートしています。Solaris 対応 ClearCase は本バージョンではサポートしていません。

Perforce や Microsoft Visual SourceSafe などのサードパーティ SCC プロバイダを利用する際は、VI を LLB に残しておくことができます。ただし、すべてのソースコード管理操作は、VI 自体ではなく、VI を含む LLB 上で実行されます。

テキストファイルでカスタムエラーコードを定義する

ナショナルインスツルメンツでは、General Error Handler VI を使用して、カスタムエラーコードを 5000 から 9999 までの範囲で定義することをお勧めします。ただし、カスタムエラーコードをその範囲で定義するには、XML ベースのテキストファイルを `labview\user.lib\errors` ディレクトリ内で作成したり、テキストファイルにエラーコードとメッセージを追加したりすることができます。いくつかの VI で同じカスタムエラーコードを使用したい場合や、アプリケーションまたは共有ライブラリとともにカスタムエラーコードを配布したい場合に、この方法を使用します。アプリケーションや共有ライブラリとともにカスタムエラーコードを配布する場合は、エラーコードのテキストファイルを配布する必要があります。

波形データタイプを含む共有ライブラリ（DLL）を作成する

波形データタイプを含む共有ライブラリ（DLL）を作成することができます。また、ライブラリ関数の呼び出しノードで共有ライブラリの波形データタイプにアクセスすることができます。

Windows 2000/Me/98 のアニメーションメニューのサポート

(Windows) LabVIEW は、メニューのフェード効果やポップアップウィンドウなど、Windows 2000/Me/98 のアニメーションメニュー機能をサポートしています。LabVIEW でこの機能を利用するには、この機能がコンピュータ上で有効になっている必要があります。Windows 98 では、こ

の機能はデフォルトで無効になっています。Windows 2000/Me では、この機能は有効になっています。LabVIEW でこの機能を無効にするには、**ツール→オプション**を選択し、上部にあるプルダウンメニューから**その他**を選択して、**メニューアニメーションを無効**チェックボックスにチェックマークを付けます。ご使用のコンピュータでウィンドウのアニメーションを有効にする方法については、Windows 2000/Me/98 のマニュアルを参照してください。

フロントパネル端子の外観

フロントパネル端子に黒い矢印が表示されて、その端子が制御器か表示器かを示します。端子が制御器の場合は矢印が右側に表示され、表示器の場合は矢印が左側に表示されます。メモリ使用を最適化するためブロックダイアグラムの強制ドットを監視して、黒い矢印が強制ドットのデフォルトのグレーの色と競合している場合、**ツール→オプション**を選択し、次に**色**ページを選択して、強制ドットのデフォルトの色を変更します。

LabVIEW 6.1 のその他の機能

- LabVIEW 6.1 は、Windows XP をサポートしています。Windows XP のサポート情報については、ナショナルインスツルメンツのウェブサイト（`ni.com/info`）にアクセスの上、info コード `winxp` を入力してください。
- **(UNIX)** マウスホイールをサポートするシステムでは、LabVIEW もこの機能をサポートしています。
- **関数→グラフィック & サウンド→画像プロット**パレットにある Radar Plot VI と Draw Legend VI は、新しく追加された VI です。
- **関数→グラフィック & サウンド→画像関数**パレットにある Color to RGB VI と RGB to Color VI は、新しく追加された VI です。
- **関数→解析→波形測定**パレットにある FFT Spectrum (Mag-Phase)、FFT Spectrum (Real-Im)、FFT Power Spectrum、FFT Power Spectral Density、Averaged DC-RMS、Basic Averaged DC-RMS、Extract Single Tone Information、Harmonic Distortion Analyzer、および SINAD Analyzer VI は、多形性 VI で、シングルチャネルまたはマルチチャネルデータを受け入れます。**関数→解析→波形監視**パレットにある Limit Testing VI は多形性 VI で、時間データまたは周波数データを受け入れます。
- VI クラスにある Run-Time Menu Path プロパティを使用して、ランタイムメニュー（`.rtm`）ファイルへのパスをプログラムで指定します。複数言語のアプリケーションを開発していて、各言語のメニューをプログラムで切り替えたい場合に、この機能は便利です。
- **First Call** 関数は、サブ VI またはブロックダイアグラムの一部が初めて実行していることを示します。
- アルゴリズムの改善により、高速フーリエ変換アルゴリズムを使用する VI はすべて実行速度が大幅に向上しています。

- Write JPEG File、Read JPEG File、Write PNG File、および Read PNG File VI は、外部の共有ライブラリを呼び出さなくなりました。その代わりに、LabVIEW で MNG ライブラリを呼び出します。
- **データタイプ**出力を使用する Enum Registry Values VI は、Enum Registry Values Simple VI に代わりました。新しい VI では、他のレジストリ VI に対応するように、**シンプルデータタイプ**出力を使用しています。Enum Registry Values VI は本バージョンでも `vi.lib` に含まれていますが、パレットには表示されません。
- グラフやチャートの目盛のサイズが変更されると、グラフまたはチャートの他の要素も移動したりサイズが変わったりします。そのような動作を無効にしてプロット領域のサイズが固定されるようにするには、グラフやチャートを右クリックしてショートカットメニューから**上級→自動スケール調整**を選択します。この動作を無効にすると、目盛が切れたり重なったりすることがあります。
- グラフカーソルをグラフの枠の外まで移動させると、グラフはカーソルの方向にスクロールします。この動作を無効にするには、グラフを右クリックしてショートカットメニューから**上級→カーソルスクロールグラフ**を選択します。この動作を無効にすると、カーソルをグラフの枠の外までドラッグしても目盛は更新されません。
- DataSocket ステータス表示器を非表示にするには、フロントパネル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**項目を表示→ DataSocket LED** を選択します。また、Datasocket：LED 表示プロパティを使用して、表示器を非表示にすることもできます。

322784B-01 0102