



IVI 計測器ドライバ プログラミングガイド LabVIEW 編

March 2019 Revision 2.2

目次

はじめに.....	3
LabVIEW で使用できる IVI 計測器ドライバ	3
使用できるインターフェース.....	4
スペシフィック・インターフェースを使用したプログラミング.....	5
スペシフィック・インターフェースを使用するための準備.....	5
LabVIEW Instrument Driver Import Wizard をインストールする	5
IVI Compliance Package をインストールする	5
IVI-C ドライバをインポートする	6
プログラムを構成する	13
関数を追加する.....	13
パラメータを設定する.....	16
プログラムを実行する.....	18
関数の解説.....	19
セッションの開始.....	19
チャンネル名の設定.....	22
セッションのクローズ	23
クラス・インターフェースを使用したプログラミング	24
プログラムを構成する	24
仮想インストルメントを作成する.....	24
関数を追加する.....	30
パラメータを設定する.....	33
関数の解説.....	35
セッションの開始.....	35
チャンネル名の設定.....	37
セッションのクローズ	38
計測器を交換する.....	39

はじめに

本ガイドでは、KikusuiPwr IVI 計測器ドライバ（KIKUSUI PWR-01 シリーズ直流電源）を使用する例を示します。

他社メーカーおよび他機種用の IVI 計測器ドライバでも、ほぼ同様の手順で使用できます。

本ガイドでは、LabVIEW 2018（64bit 版）を使用し、Windows10（x64）上で動作する 64bit（x64）プログラムを作成する場合を例に説明します。

LabVIEW で使用できる IVI 計測器ドライバ

LabVIEW では、IVI-C 計測器ドライバをインポートしてプログラミングすることができます。

IVI-C 計測器ドライバを使用することで、より簡単にプログラミングすることができます。

本ガイドでは、IVI-C 計測器ドライバを推奨し、IVI-C 計測器ドライバを使用してプログラミングする手順を例に説明しています。

Memo

当社の IVI 計測器ドライバには IVI-C ドライバと IVI-COM ドライバの両方が含まれています。IVI-COM 計測器ドライバを一般の COM コンポーネントとして直接利用することも可能です。

使用できるインターフェース

IVI 計測器ドライバは以下の 2 つのインターフェースに対応しています。

- **スペシフィック・インターフェース**

計測器ドライバに固有のインターフェースです。使用する計測器の機能を最大限に利用できます。

- **クラス・インターフェース**

IVI 仕様書で定義されている計測器クラスのインターフェースです。

インターチェンジャビリティ機能を利用できますが、機種固有の機能を使うことは制限されます。

本ガイドでは、それぞれのインターフェースを使用してプログラミングする方法について説明します。

Memo

- 計測器ドライバが所属する計測器クラスは、ドライバの `Readme.txt` に記載されています。Readme 文書は、[Start] ボタン > [Kikusui] > [KikusuiPwr IVI Driver 1.0.0 Documentation] メニューから確認できます。
- 計測器ドライバがどの計測器クラスにも属していない場合、クラス・インターフェースを利用できないため、インターチェンジャビリティ機能を利用するアプリケーションは作成できません。

スペシフィック・インターフェースを使用したプログラミング

スペシフィック・インターフェースを使用すると、計測器ドライバの機種固有機能を最大限に利用できます。

ここでは、スペシフィック・インターフェースを使用してプログラムする手順を説明します。

Memo

スペシフィック・インターフェースの場合、インターチェンジャビリティ機能は利用できません。インターチェンジャビリティ機能を利用する場合は、クラス・インターフェースを使用してください。(⇒ 24 ページ)

スペシフィック・インターフェースを使用するための準備

LabVIEW Instrument Driver Import Wizard をインストールする

LabVIEW で IVI-C 計測器ドライバを呼び出すコードを記述するためには、LabVIEW Instrument Driver Import Wizard を使用して IVI-C ドライバをインポートし、LabVIEW VI ライブラリ (LabVIEW IVI-C ラッパー) を生成します。

あらかじめ、National Instruments 社の Web サイトから、LabVIEW Instrument Driver Import Wizard をダウンロードしてインストールしてください。

IVI Compliance Package をインストールする

Import Wizard で生成された LabVIEW IVI-C ラッパーを機能させるには、ランタイム環境として IVI Compliance Package が必要です。

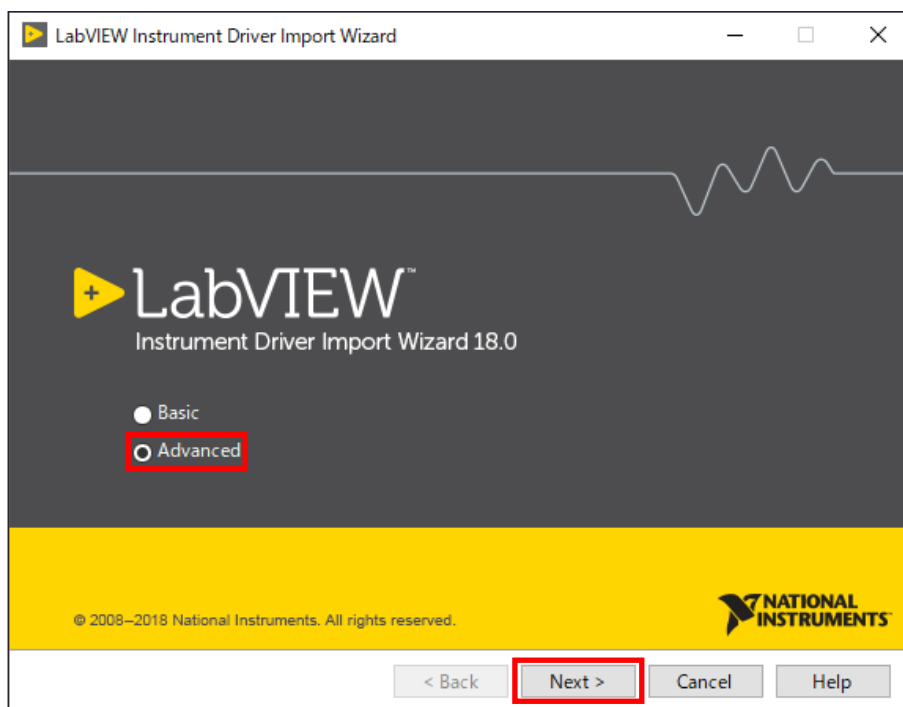
あらかじめ、National Instruments 社の Web サイトから、IVI Compliance Package をダウンロードしてインストールしてください。

IVI-C ドライバをインポートする

スペシフィック・インターフェースを使用してプログラムするには、IVI-C ドライバを LabVIEW 環境に対応した形式に変換して、インポートする必要があります。

IVI-C ドライバのインポートには、LabVIEW Instrument Driver Import Wizard (⇒ 5 ページ) を使用します。

- 1 **Instrument Driver Import Wizard** がインストールされている状態で **LabVIEW** を起動します。
- 2 **[Tools] > [Instrumentation] > [Import LabWindows/CVI Instrument Driver]** の順にメニューを選択します。
- 3 **[Advanced]** を選択して、**[Next >]** をクリックします。



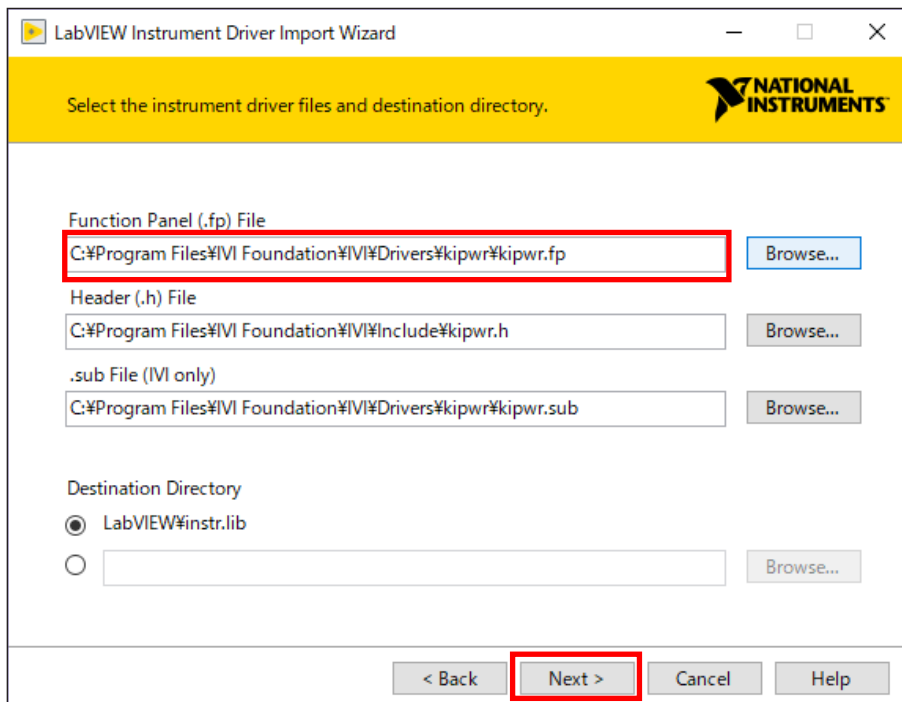
4 [Function Panel (.fp) File] に IVI 計測器ドライバの .fp (ファンクション・パネル) ファイルを指定し、[Next >] をクリックします。

この例では、KikusuiPwr IVI 計測器ドライバを使用するとして kipwr.fp を指定しています。

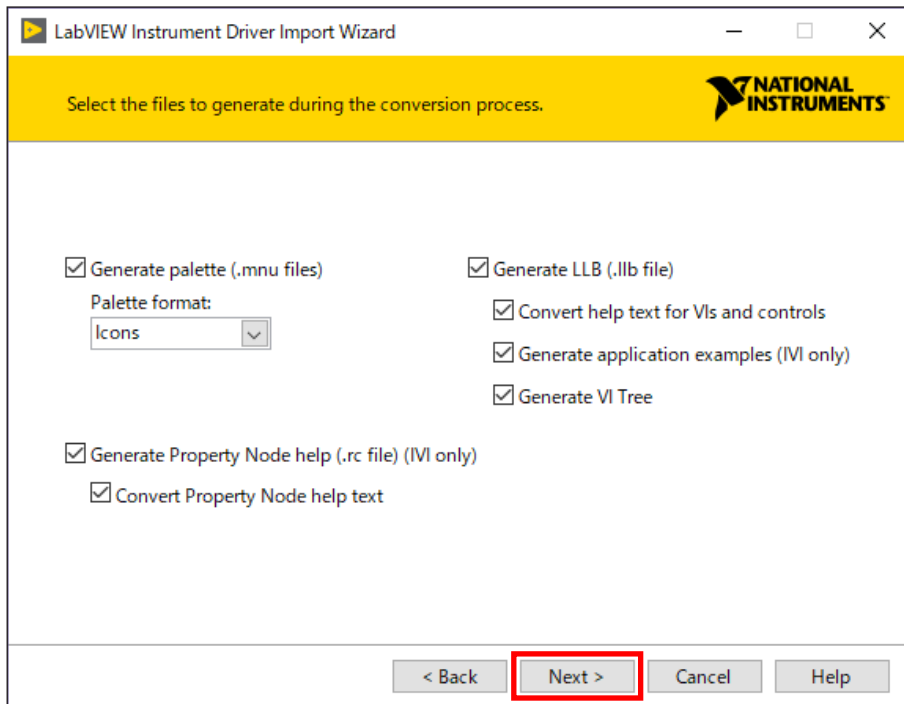
kipwr.fp は以下に格納されています。

C:/Program Files/IVI Foundation/IVI/Drivers/kipwr

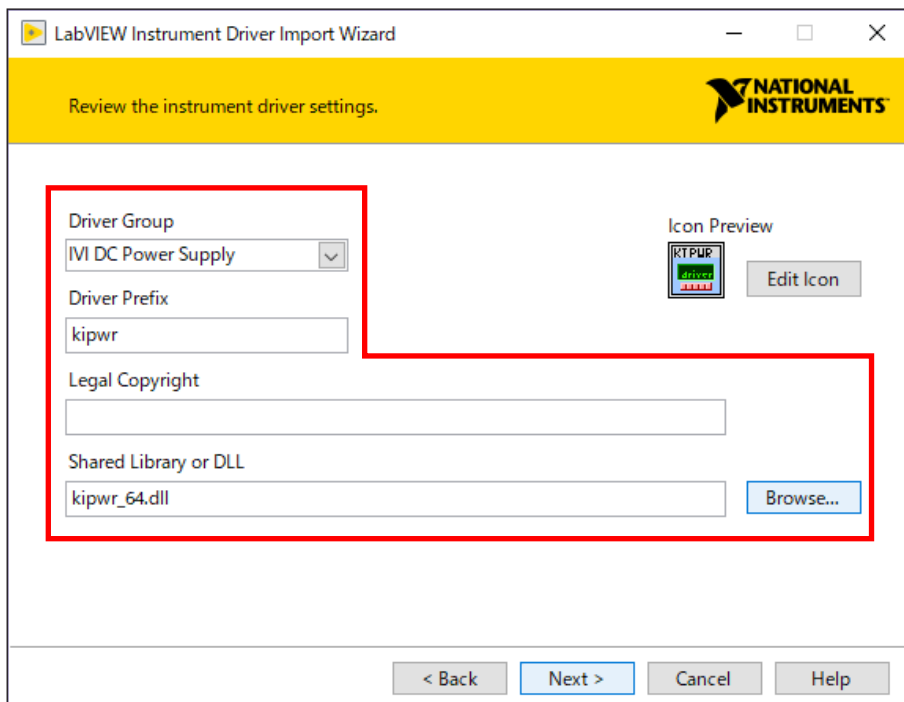
.h ファイル (C 言語用ヘッダ・ファイル) および .sub ファイル (属性情報ファイル) は自動的に指定されます。



- 5 生成ファイルの種別が全項目選択されていることを確認し、[Next >] をクリックします。



- 6 それぞれの項目を設定します。



設定項目	内容
[Driver Group]	<p>IVI 計測器ドライバが所属する計測器クラスを選択します。この例では、IviDCPwr クラス (IVI 直流電源) を指定します。</p> <p>インポートする IVI 計測器ドライバがどのクラスにも属さない場合は、[IVI Generic] を選択してください。</p>
[Driver Prefix]	<p>手順 4 で選択した .fp ファイルであることを確認します。</p>
[Legal Copyright]	<p>計測器ドライバに含める著作権表示を設定します。この例では、空欄に変更します。</p>
[Shared Library or DLL]	<p>ドライバの DLL ファイルを指定します。この例では、以下のディレクトリに格納されている [kipwr.dll] を指定します。</p> <p>C:/ProgramFiles/IVI Foundation/IVI/BIN</p> <p>デフォルトではワイルドカードを使用したファイル名が入力されていますが、正しい DLL ファイルを指定してください。</p>

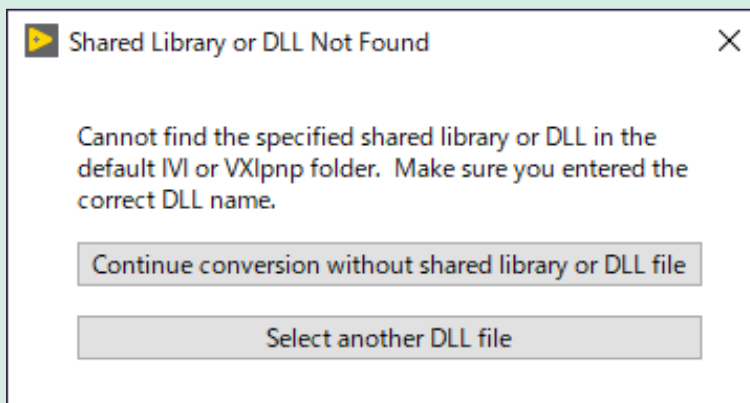
7 [Next >] をクリックします。

関数のツリー画面が表示されます。

Memo

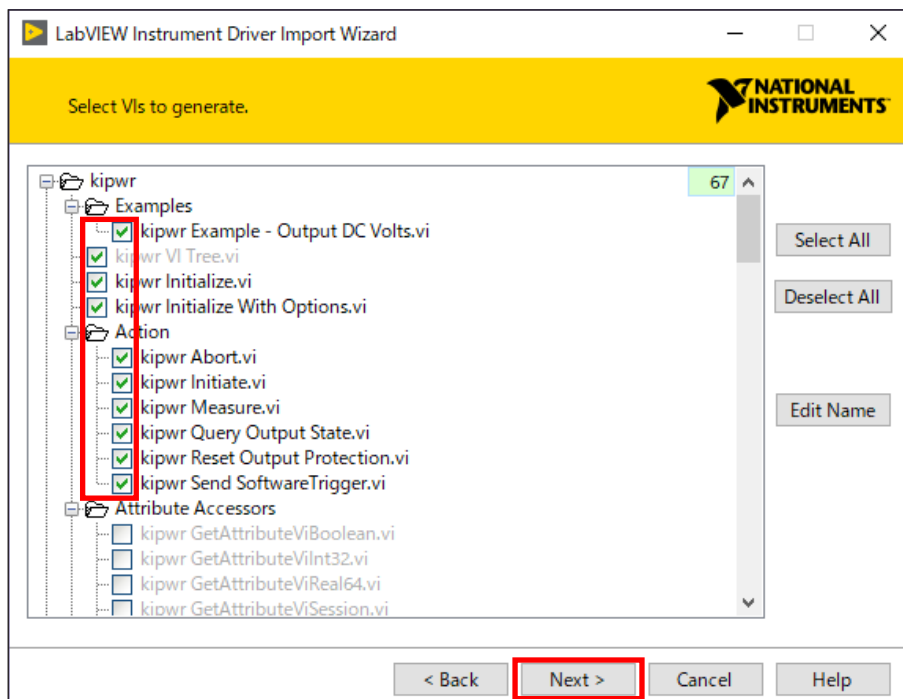
• [Shared Library or DLL] の指定が不適切な場合は、以下のダイアログが表示されます。

[x]をクリックしてダイアログを閉じ、DLL ファイルを指定し直すか、IVI 計測器ドライバのインストール状態を再確認してください。



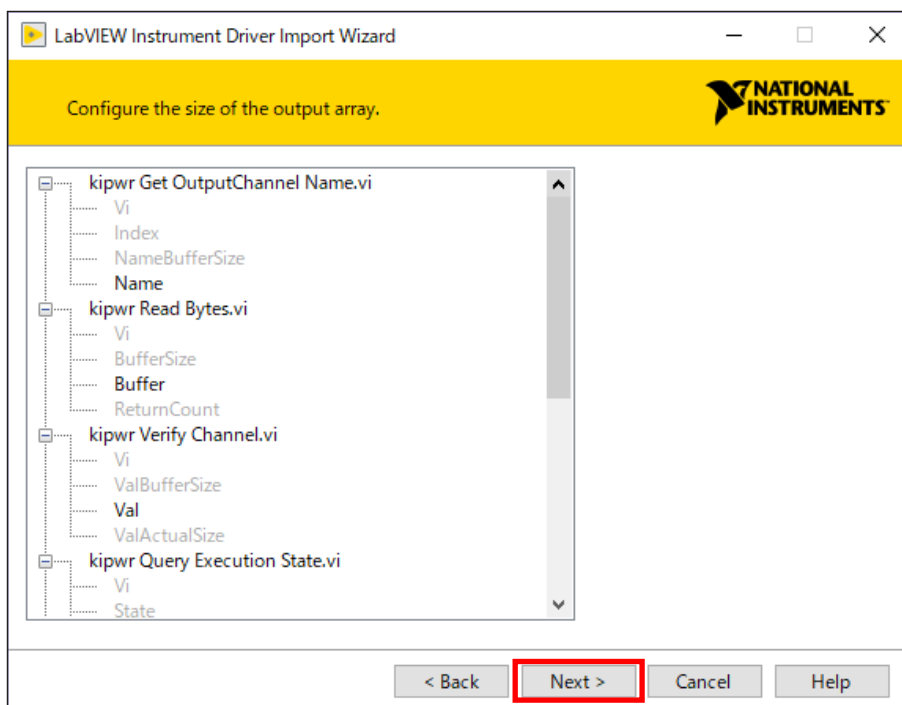
8 生成する VI を選択し、[Next >] をクリックします。

通常はデフォルトのまま変更する必要はありません。



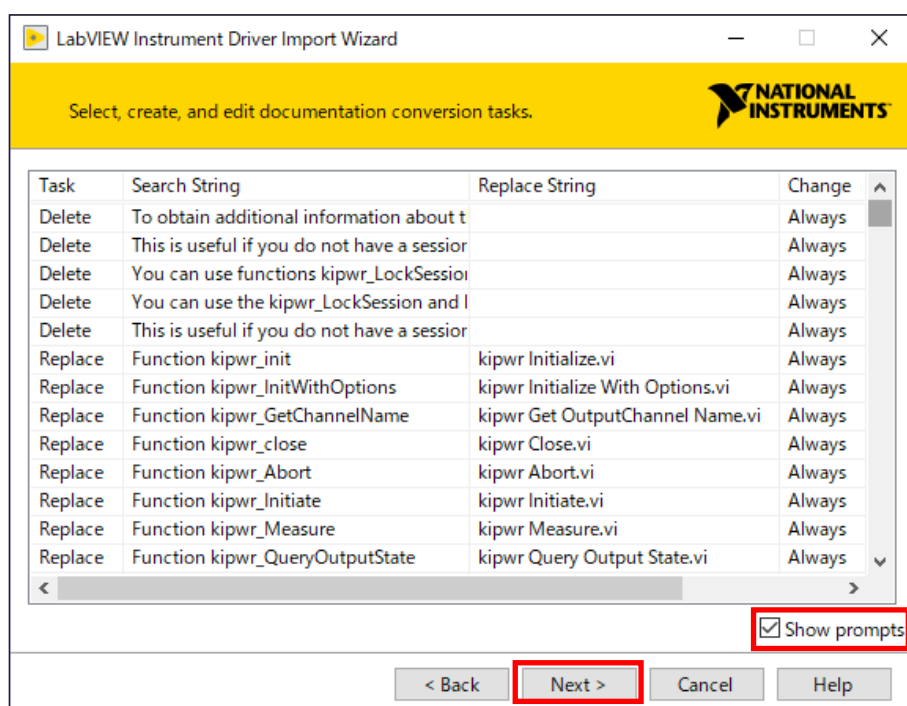
9 [Next >] をクリックします。

VI で使用するメモリサイズの変更画面が表示されますが、通常は変更する必要はありません。



10 ドキュメント（主にコンテキスト・ヘルプ）に関する変換条件を指定し、[Next >] をクリックします。

ドキュメントに関する変換のほとんどは、「Function」を「VI」に置き換えるものです。重要性は低く、また変換の際に確認プロンプトが都度表示されることを避けるため、本ガイドでは [Show prompts] のチェックを外すことを推奨します。



変換条件の概要が表示されます。

11 [Next >] をクリックします。

変換が開始されます。変換が完了すると、LabVIEW のデフォルト計測器ドライバ・ディレクトリに IVI 計測器ドライバのサブディレクトリ（この例では kipwr サブディレクトリ）が作成され、LabVIEW から直接利用可能な IVI-C ラッパーが生成されます。

kipwr IVI-C ラッパーは、LabVIEW のブロックダイアグラム上で Instrument I/O ファンクション・パレットから参照できます。

Memo

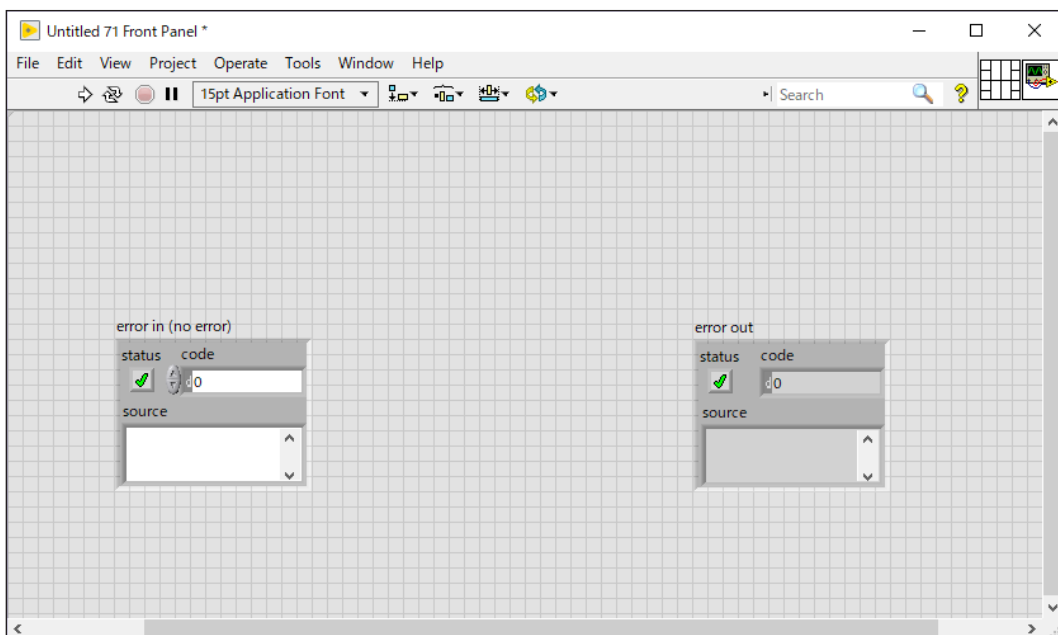
- 通常、LabVIEW のデフォルト計測器ドライバ・ディレクトリは以下となります。
C:/Program Files/National Instruments/LabVIEW 2018/instr.lib
- IVI-C ラッパーは、VI ライブラリ・ファイル (.lib)、複数のパレット・メニューファイル (.mnu) などが含まれたファイルセットとして生成されます。
- 生成された .lib ファイルは計測器ドライバの実体ではなく、IVI-C ドライバへのラッパー・モジュールです。
そのため、完成したアプリケーションの実行時、ターゲット・マシンに IVI 計測器ドライバがインストールされている必要があります。
- ターゲット・マシンへ IVI 計測器ドライバをインストールするには、開発マシンへのインストールと同様に、ドライバのインストーラを使用してください。
DLL などの各データをコピーしただけでは正常に動作しません。
- ターゲット・マシンには、LabVIEW Runtime Engine に加えて IVI Compliance Package をインストールする必要があります。

プログラムを構成する

関数を追加する

スペシフィック・インターフェースを使用して関数を追加する方法を説明します。ここでは、電圧、電流、出力の関数を追加する場合を例に説明します。

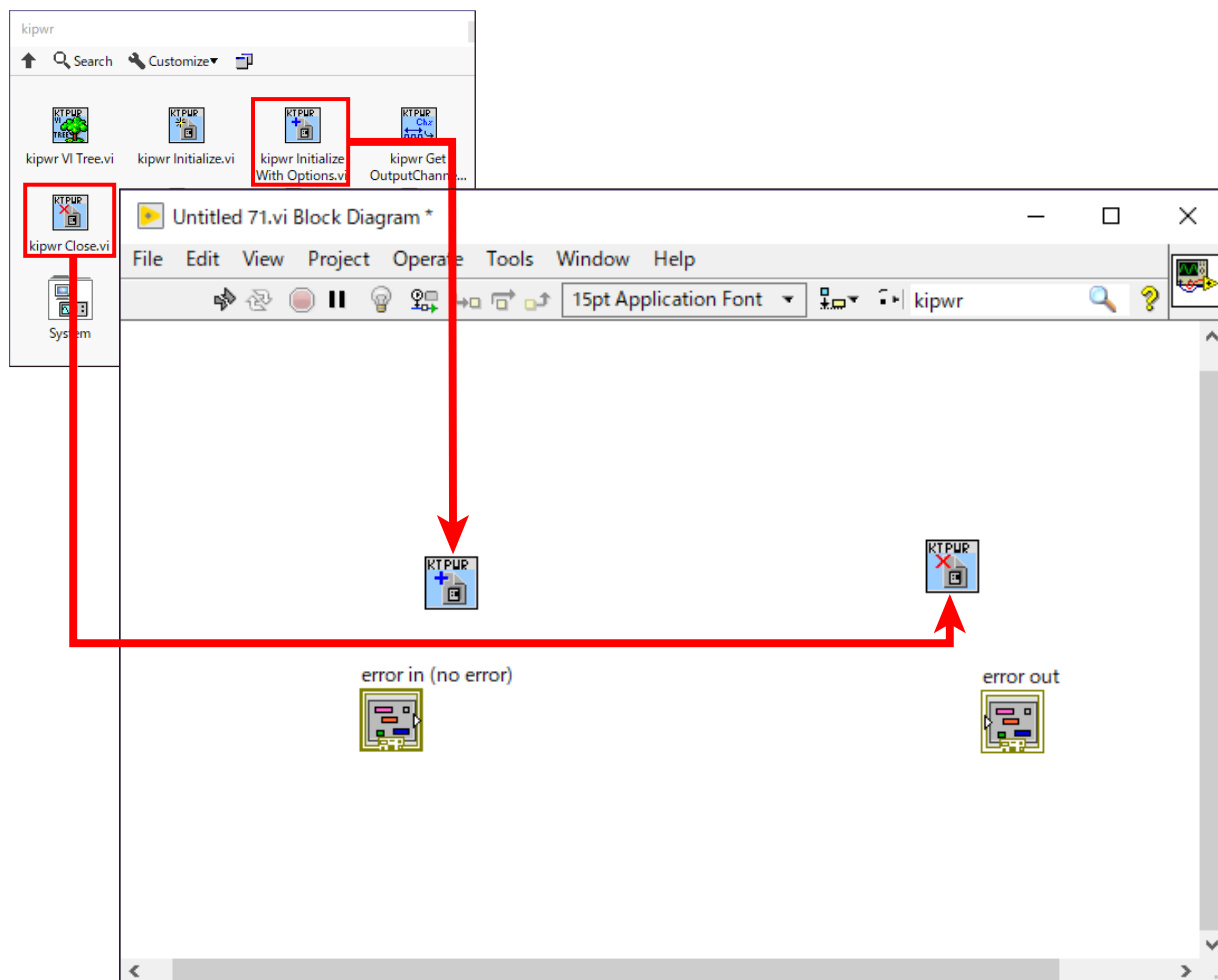
- 1 **Front Panel** 画面を表示して、**[error in]** クラスタと **[error out]** クラスタを置きます。



- 2 **Block Diagram** 画面を表示して右クリックし、**[Instrument I/O]** > **[Instr Drivers]** > **[kipwr]** から **kipwr** 関数パレットを開きます。

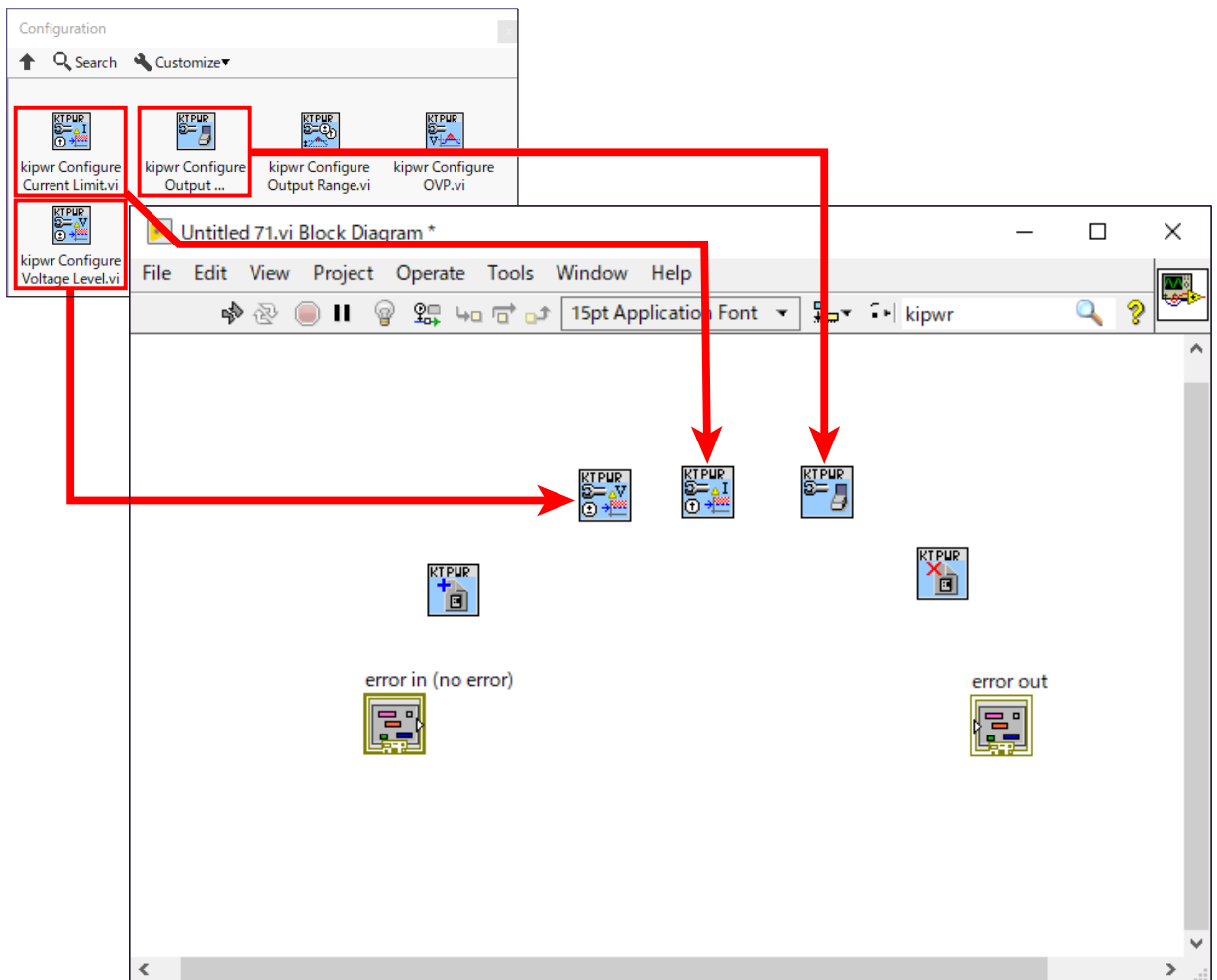
3 Block Diagram 画面に以下を追加します。

- Kipwr Initialize With Options.vi
- Kipwr Close.vi



4 [Configuration] パレットを開き、以下の VI を追加します。

パラメータ	VI
電圧を設定するパラメータ	Configure Voltage Level.vi
電流を設定するパラメータ	Configure Current Limit.vi
アウトプットを設定するパラメータ	Configure Output Enabled.vi



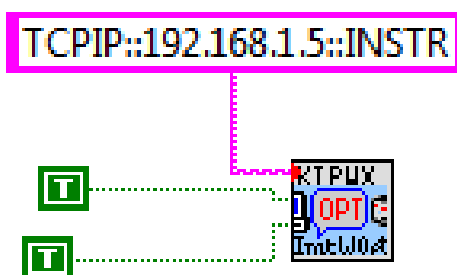
続いて、追加した関数にパラメータを設定します。

パラメータを設定する

追加した関数にパラメータを設定し、プログラムを構成する方法を説明します。ここでは、電圧 20 V、電流 2 A を設定して、出力をオンにする場合を例に説明します。

以下の説明では、ネットワーク上の Kikusui PWR-01 シリーズ直流電源が IP アドレス 192.168.1.5 に設定されているという前提で、パラメータを設定する例を説明します。

- 1 [resource name]、[ID Query]、[Reset Device] のパラメータを、[Initialize With Options.vi] に渡します。



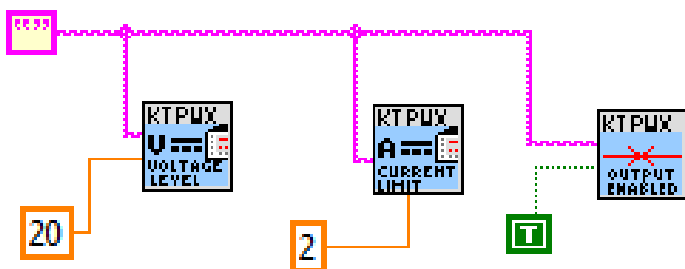
- 2 電圧、電流、アウトプットを設定するパラメータを、それぞれの VI に追加します。

- 3 制御対象の直流電源のチャンネル名の文字列を、手順 2 と同じ 3 つの VI に共通で渡します。

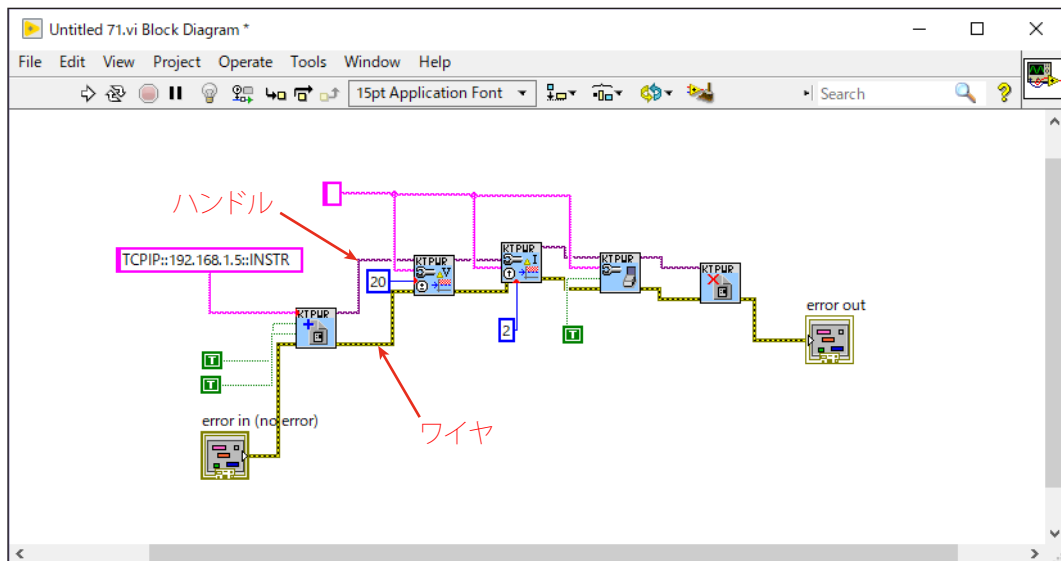
この例では、空白文字列をチャンネル名としています。

空白文字列を使用できるのはチャンネルが一つの場合です。

実際に使用できるチャンネル名の詳細は、ドライバのオンライン・ヘルプなどを参照してください。



4 [error in] クラスタから [error out] クラスタをワイヤで接続し、計測器セッション (ハンドル) を接続します。



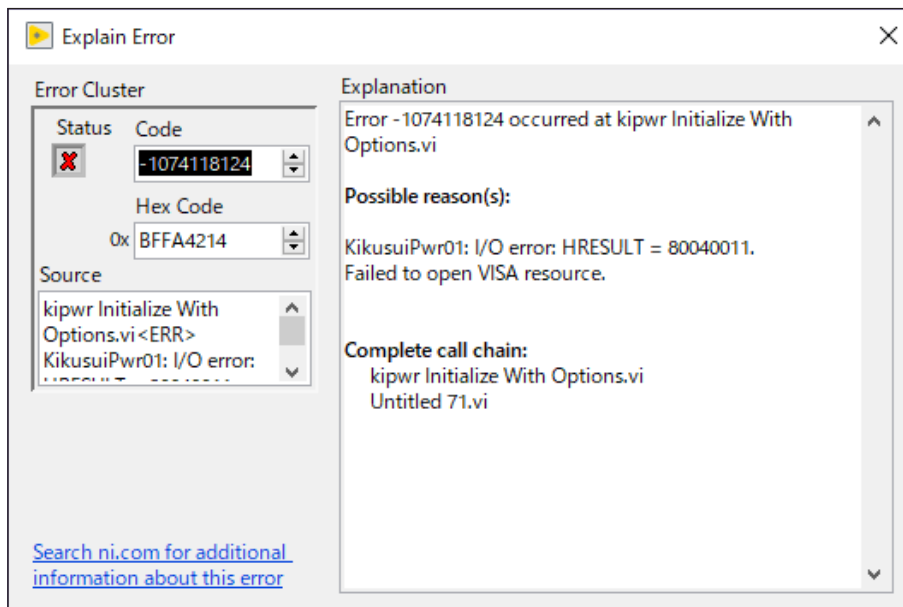
プログラムを実行する

ここまでに設定した構成内容でも、プログラムを実行することが可能です。

Initialize With Options.vi の [Reset Device] パラメータには、デフォルトで「TRUE」が指定されています。そのため、プログラムを実行すると計測器がリセットされたのち、通信が開始されます。

関数やパラメータの内容および設定値については、「関数の解説」(⇒ 19 ページ) を参照してください。

- 実際に計測器が接続されている場合、[error out] クラスタのエラー・コードに「0」が表示されれば、[Initialize With Options] および [Close] は成功しています。
- 通信に失敗した場合、VISA ライブラリの設定が正しく行われていない場合などは例外が発生し、[error out] クラスタに表示されます。
[error out] クラスタを右クリックし、[Explain Error] を選択すると、詳細を確認できます。



関数の解説

ここでは、プログラムを構成する関数（VI）について、その内容や設定値などを説明します。

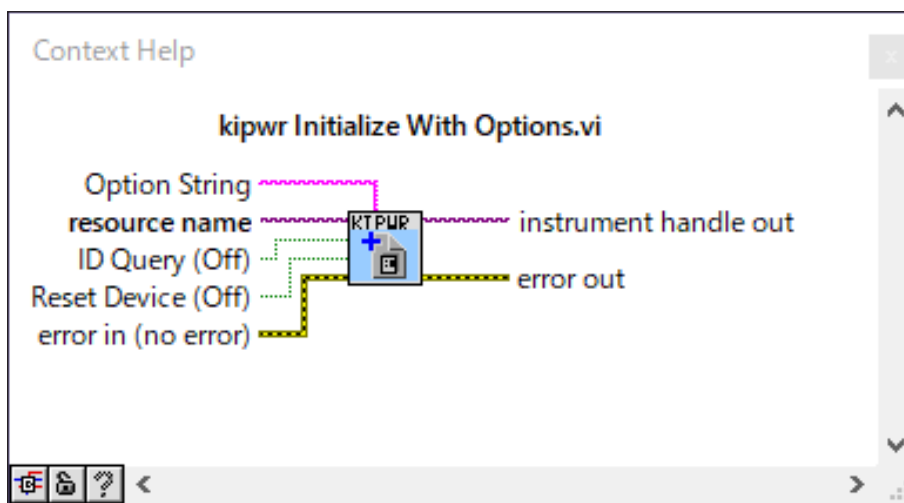
Memo

IVI-C 計測器ドライバおよび VXI Plug&Play 計測器ドライバには、関数名のルールとして、<prefix> という表記があります。この表記は、各計測器ドライバの識別名を表します。例えば「<prefix>Initialize.vi」という表記は、kipwr 計測器ドライバでは「kipwr Initialize.vi」となります。

セッションの開始

セッションの開始には、Initialize With Options.vi を使用します。

Initialize With Options.vi は IVI 仕様書で定義された VI で、すべての IVI 計測器ドライバに含まれています。



kipwr Initialize With Options.vi で設定できるパラメータは以下の通りです。

パラメータ	タイプ	説明
resource name	String	計測器が接続されている I/O インターフェース、アドレスなどによって決定される VISA リソース名の文字列です。 例えば、IP アドレス 192.168.1.5 の計測器を VXI-11 で接続している場合は、「TCPIP::192.168.1.5::INSTR」となります。
ID Query	Boolean	VI_TRUE を指定した場合、計測器に対して "*IDN?" などの ID クエリを発行して機種情報を問い合わせます。
Reset Device	Boolean	VI_TRUE を指定した場合、"*RST" コマンドなどを発行して計測器の設定をリセットします。
Option String	String	RangeCheck Cache Simulate QueryInstrStatus RecordCoercions InterchangeCheck 上記の IVI 定義の設定を変更できます。 また、計測器ドライバごとに独自に定義される [DriverSetup] パラメータがサポートされている場合は、その設定を行うことができます。

Option String を設定するには

Option String は以下のように設定してください。

- Option String は文字列パラメータです。複数の項目を設定する場合には、カンマで区切ります。下記のサンプル文字列と同じ書式で設定してください。

QueryInstrStatus=TRUE, Cache=TRUE, DriverSetup=12345

- 機能名称および設定値はケース・インセンシティブです。大文字と小文字は区別されません。
- 設定値は ViBoolean 型です。「VI_TRUE」、「VI_FALSE」、「1」、「0」のいずれかが設定できます。
- 特に設定値を指定しない場合には、下記の IVI 仕様書で定義されたデフォルト値が適用されます。
 - [RangeCheck]、[Cache] : VI_TRUE
 - [RangeCheck]、[Cache] 以外 : VI_FALSE
- 計測器ドライバによっては、[DriverSetup] パラメータがサポートされています。
[DriverSetup] パラメータは、IVI 仕様書では定義されない項目を Initialize With Options.vi の呼び出し時に指定するパラメータです。利用目的や書式はドライバに依存します。
そのため、[DriverSetup] を設定する場合は [Option String] の最後の項目として指定します。
[DriverSetup] の指定内容については、ドライバの Readme またはオンライン・ヘルプなどを参照してください。

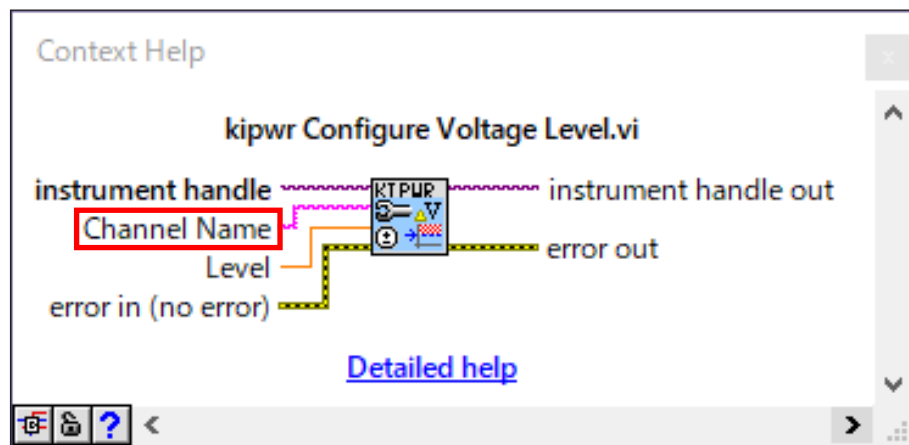
Memo

- <prefix> Initialize.vi および <prefix> Initialize With Options.vi を除くすべての VI (ドライバ関数) は、左上の入力パラメータが [instrument handle (in)] になります。
- <prefix> Close.vi を除くすべての VI (ドライバ関数) は、右上の出力パラメータが [instrument handle out] になります。
この出力パラメータは、次に来る VI の [instrument handle (in)] に接続します。
- <prefix> Initialize.vi は VXI Plug&Play ドライバ仕様との互換性のために残されている VI です。[Option string] パラメータを指定できない点を除き、<prefix> Initialize With Options.vi と動作は同じです。

チャンネル名の設定

電源装置やオシロスコープなどの場合、IVI 計測器ドライバは複数のチャンネルが装備されていることを前提に設計されています。

そのため、計測器のパネル設定に関する操作を行うドライバ関数の多くは、[Channel Name] パラメータにチャンネルを指定する必要があります。



例えば本ガイドの例の KikusuiPwr (kipwr) ドライバは、直流電源装置のドライバです。チャンネル名には制御対象のチャンネル名を指定してください。

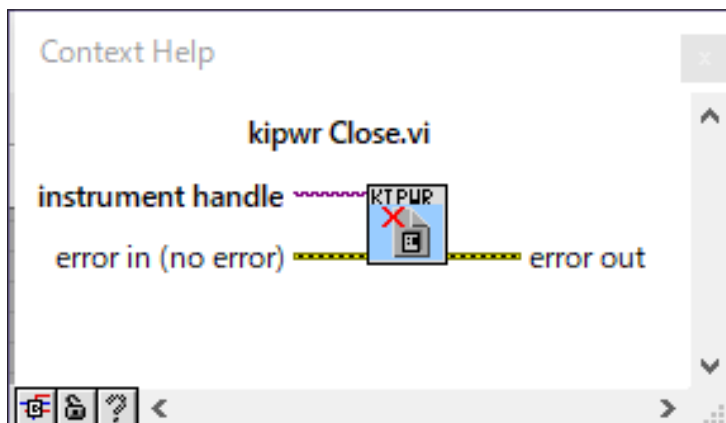
実際に使用できるチャンネル名は、ドライバによって異なります。設定の際は、ドライバの Readme またはオンライン・ヘルプなどを参照してください。

Memo

- チャンネルが1つのみの場合は、チャンネル名に空白文字列（「」）を指定しても動作させることができます。
- チャンネルが複数ある場合は、使用するチャンネル名を指定する必要があります。通常、電源装置では「Output1」のように指定します。
- PWR-01 シリーズをマルチドロップ拡張で運用する場合、チャンネル番号は0から順に割り当てられます。そのため、最初のチャンネル名は「Output0」となります。

セッションのクローズ

計測器ドライバのセッションをクローズするには、Close.vi を使用します。



クラス・インターフェースを使用したプログラミング

IVI 仕様で定義された計測器クラスのインターフェースを使用してプログラミングすることにより、計測器クラス・ドライバを利用したインターチェンジャビリティを実現できます。

インターチェンジャビリティを利用すると、アプリケーションを再度コンパイル・リンクすることなく計測器を交換できます。

Memo

- インターチェンジャビリティを利用するには、交換前後の両機種に対して IVI-C 計測器ドライバが提供されており、かつそれらのドライバが同じ計測器クラスに属している必要があります。異なる計測器クラス間でのインターチェンジャビリティは実現できません。
- クラス・インターフェースを使用したプログラミングでは、使用できる機種固有の機能が制限されます。機種固有の機能を最大限に使用するには、スペシフィック・インターフェースを使用してプログラミングしてください。(⇒ 5 ページ)

プログラムを構成する

仮想インストルメントを作成する

インターチェンジャビリティ機能を利用するアプリケーションを作成するには、あらかじめ仮想インストルメントを作成する必要があります。

Memo

インターチェンジャビリティ機能を損なうため、アプリケーション・コード内に特定の IVI-C 計測器ドライバに依存した記述（例：kipwr_init 関数の直接コール）や、特定 VISA アドレス（リソース名）の記述（例："TCPIP::192.168.1.5::INSTR"）などを行わないでください。

IVI 仕様では、計測器ドライバとアプリケーションの外部に IVI コンフィグレーション・ストアを置くことでインターチェンジャビリティ機能を実現します。

アプリケーションは機種固有の計測器ドライバを直接使うのではなく、計測器クラス・ドライバと呼ばれる特別な計測器ドライバを通じて制御します。

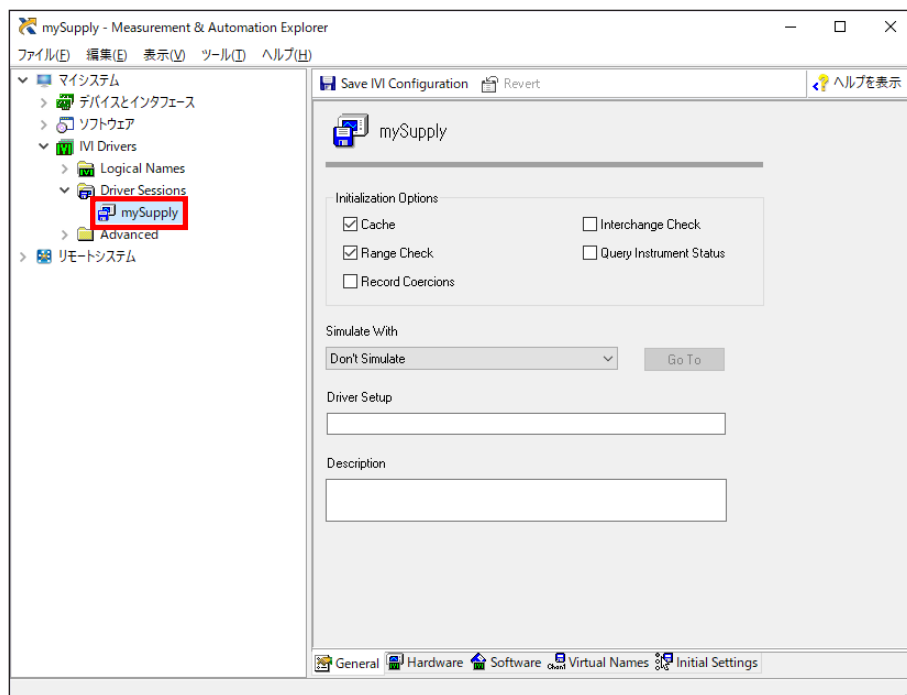
制御の際は、IVI コンフィグレーション・ストアの内容に従って計測器ドライバの DLL を選択し、間接的にロードされた計測器ドライバに、機種に依存しないクラス・ドライバの関数を通じてアクセスします。

IVI コンフィグレーション・ストアには、XML ファイル (C:/ProgramData/IVI Foundation/IVI/IviConfigurationStore.xml) が使用されます。主に IVI 計測器ドライバや一部の VISA/IVI コンフィグレーション・ツールが、IVI Configuration Server DLL を通じてアクセスします。アプリケーションが利用することは通常はありません。

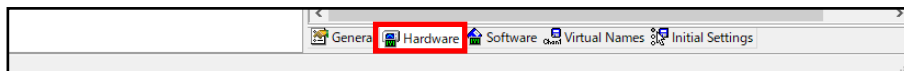
LabWindows/CVI を使用する場合は、National Instruments 社製のソフトウェア NI-MAX (NI Measurement and Automation Explorer) を使用して IVI ドライバのコンフィグレーションを行います。

以降、NI-MAX を使用して仮想インストルメントを作成する手順について説明します。

- 1 はじめに、**Driver Session** を作成します。NI-MAX を起動し、**[IVI Drivers]** ノードの階層を参照します。
- 2 **[Driver Session]** を右クリックし、**[Create New (case-sensitive)]** を選択します。
- 3 **Driver Session** の名前を設定します。
ここでは **[mySupply]** とします。



4 次に、Hardware Asset を作成します。[Hardware] タブを選択します。



Memo

Hardware Asset（ハードウェア・アセット）では、使用する計測器がどのような経路に接続されているかが示されます。

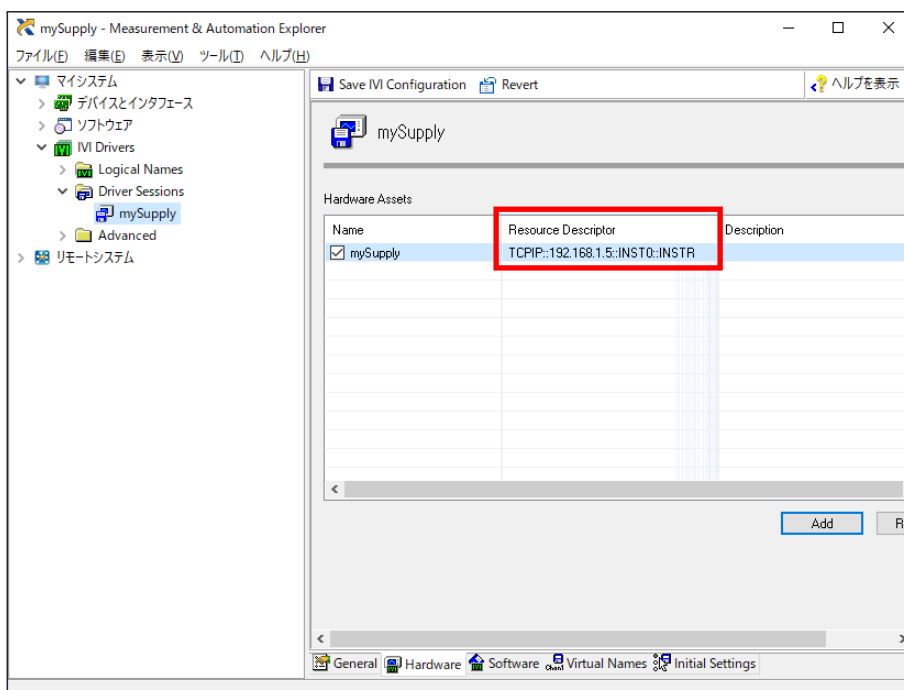
5 [Add] をクリックして Hardware Asset を新規作成します。

6 Hardware Asset の名前を設定します。

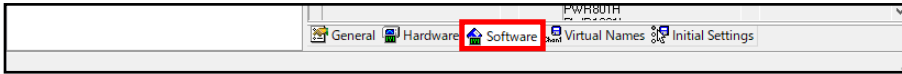
ここでは [mySupply] とします。

7 [Resource Descriptor] に、使用する計測器が接続されている VISA アドレスを指定します。

この例では、「TCPIP::192.168.1.5::inst0::INSTR」を指定しています。



8 次に、**Software Module** リンクを設定します。**[Software]** タブを選択します。

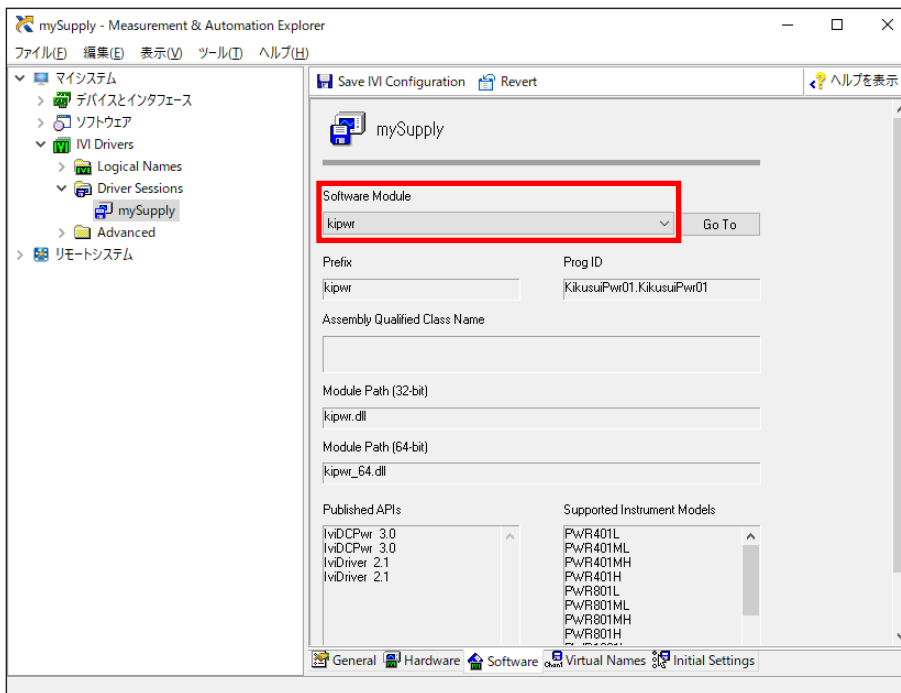


Memo

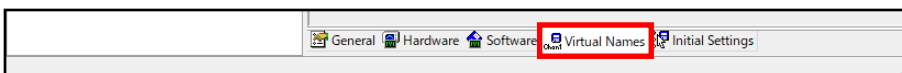
Software Module (ソフトウェア・モジュール) は、計測器ドライバモジュール (DLL モジュール) を示します。

9 **[Software Module]** のリストから、使用する計測器ドライバモジュールを選択します。

この例では、**[kipwr]** を選択しています。



10 次に、**Virtual Name** を作成します。**[Virtual Names]** タブを選択します。



Memo

チャンネルの指定が必要な計測器ドライバの場合、有効なチャンネル名は計測器ドライバによって異なります。
そのため、Virtual Name タブでチャンネル名を仮想化する設定をします。

- 13 最後に、**Logical Name** を作成し、リンクを設定します。画面左のツリー表示で [IVI Drivers] 以下の階層を確認します。

Memo

Logical Name (ロジカル・ネーム) とは、NI-MAX で設定される仮想計測器の名前に相当します。

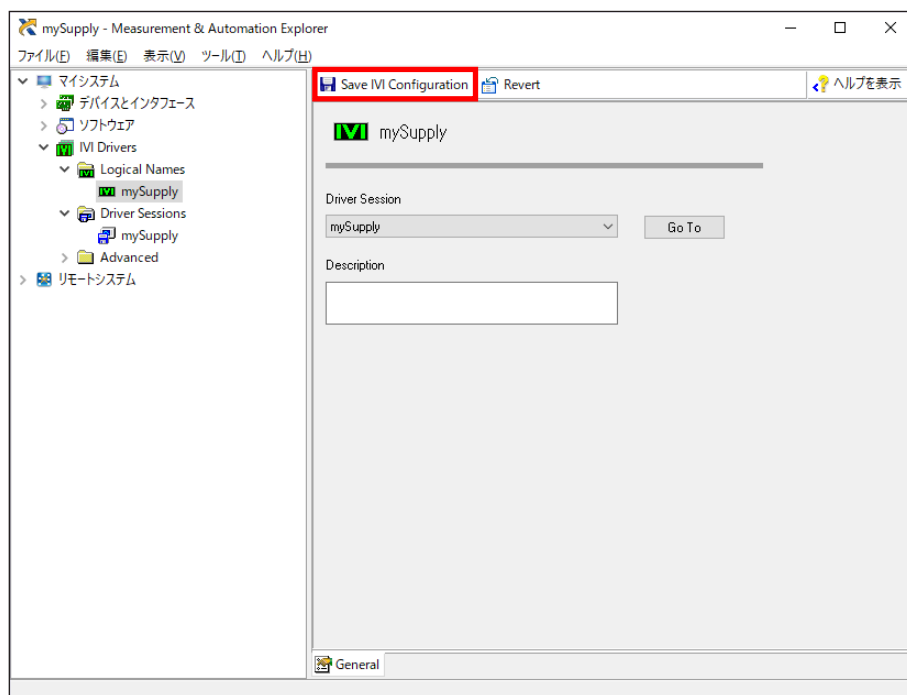
- 14 [Logical Name] を右クリックして [Create New (case-sensitive)] を選択します。

- 15 Logical Name の名前を設定します。

ここでは [mySupply] とします。

- 16 [Driver Session] で [mySupply] を選択します。

- 17 ツールバー上の [Save IVI Configuration] をクリックし、設定を保存します。

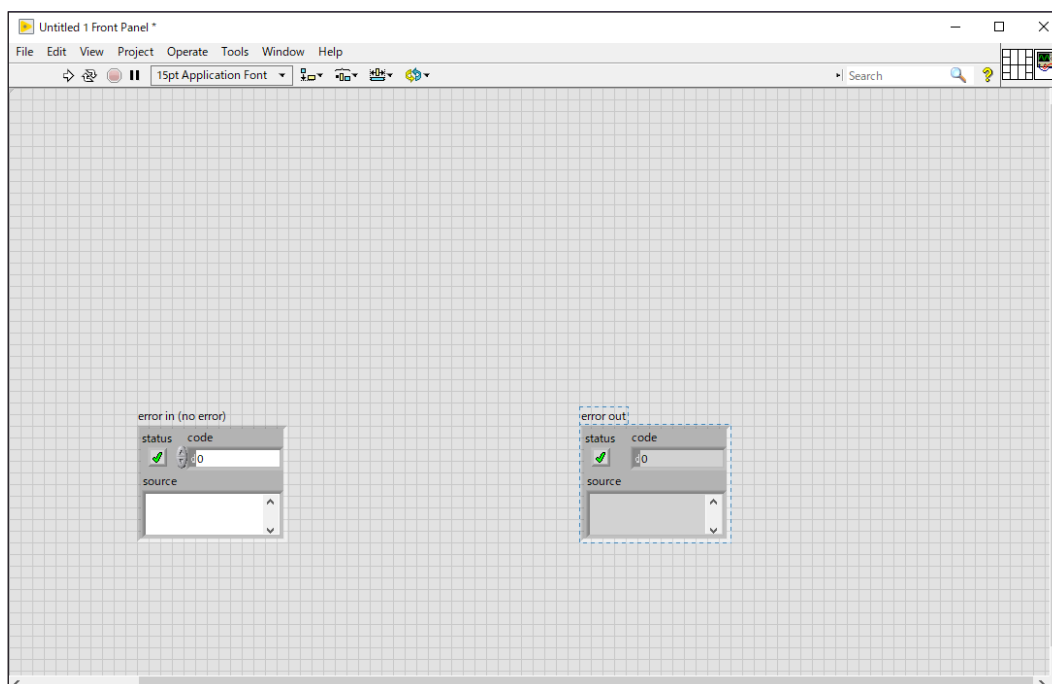


以上で、仮想インストルメントの作成は完了です。

関数を追加する

クラス・インターフェースを使用して関数を追加する方法を説明します。ここでは、電圧、電流、出力の関数を追加する場合を例に説明します。

- 1 **Front Panel** 画面を表示して、**[error in]** クラスタと **[error out]** クラスタを置きます。



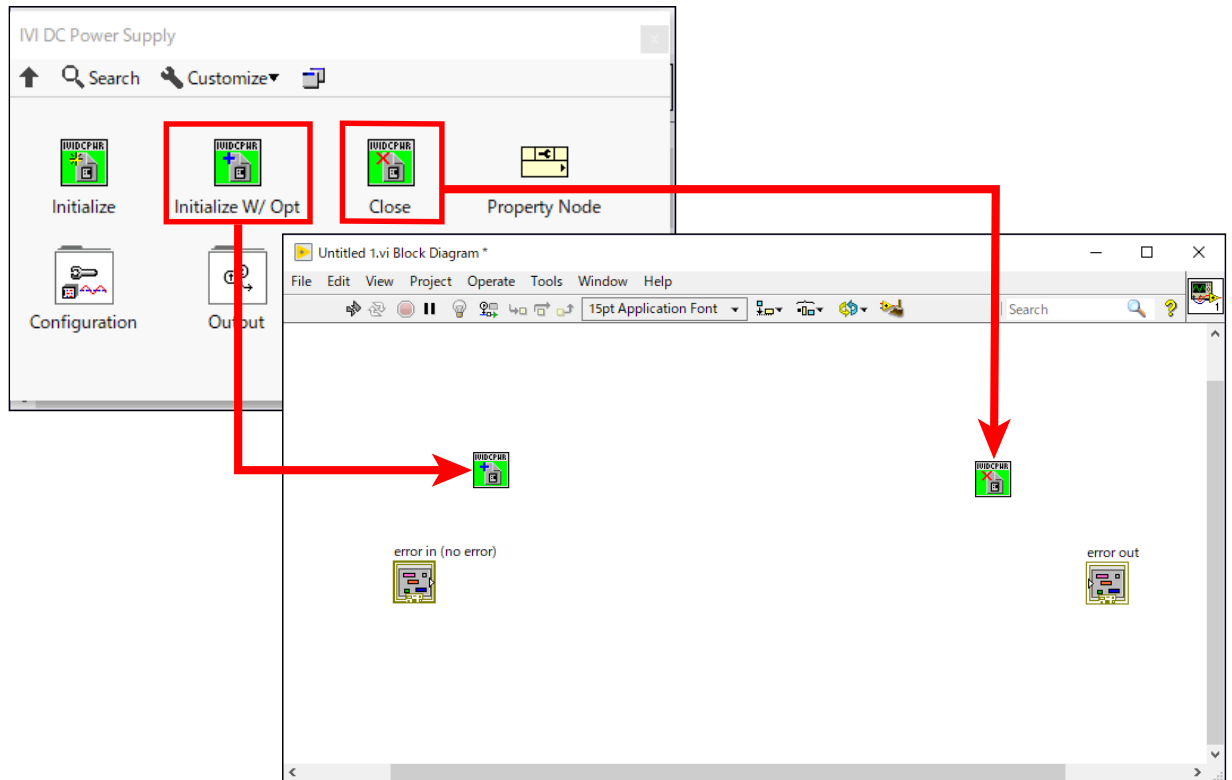
- 2 **Block Diagram** 画面を表示して右クリックし、**[Instrument I/O]** > **[IVI Class Drivers]** > **[DC Power Supply]** から **IviDCPwr** 関数パレットを開きます。

Memo

IVI Class Drivers のパレットが見つからない場合、適切なバージョンの IVI Compliance Package がインストールされていない可能性があります。

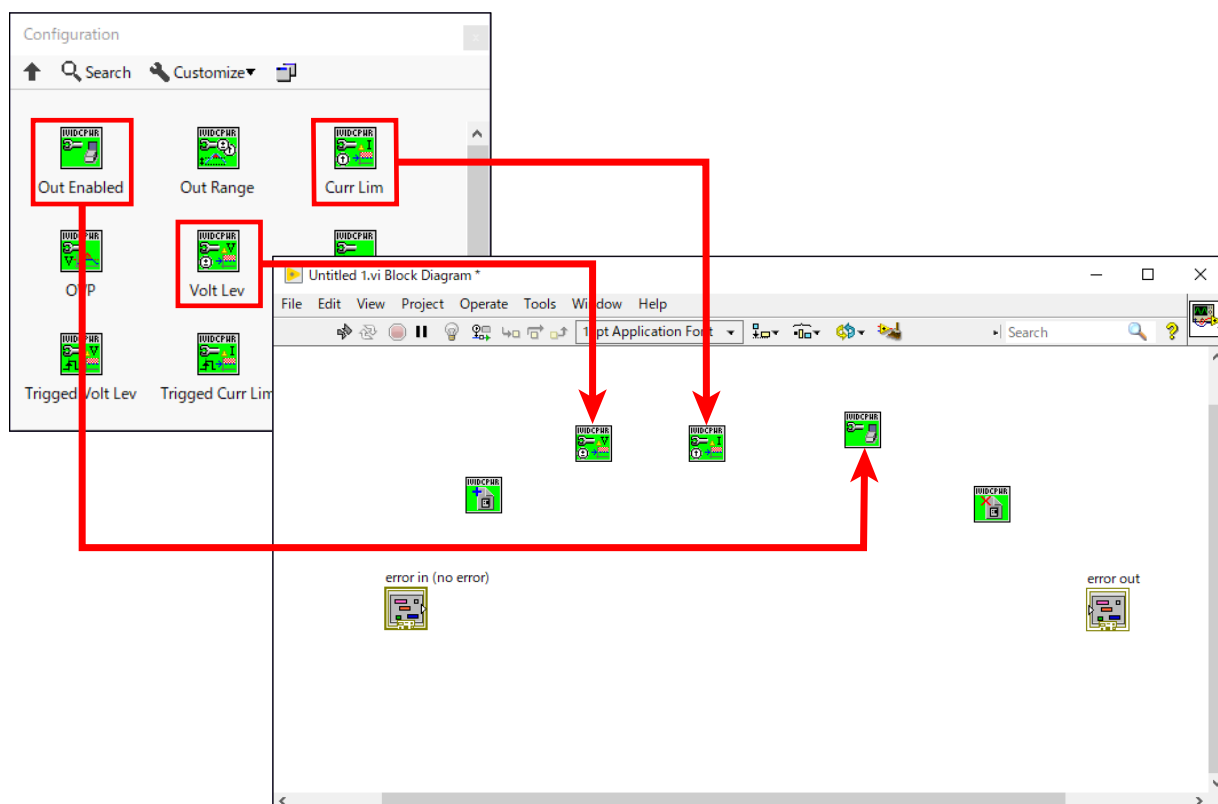
3 Block Diagram 画面に以下を追加します。

- Initialize W/Opt (Initialize With Options.vi)
- Close (Close.vi)



4 [Configuration] パレットを開き、以下を追加します。

パラメータ	VI
電圧を設定するパラメータ	Configure Voltage Level.vi
電流を設定するパラメータ	Configure Current Limit.vi
アウトプットを設定するパラメータ	Configure Output Enabled.vi



続いて、追加した関数にパラメータを設定します。

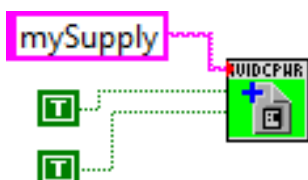
パラメータを設定する

追加した関数にパラメータを設定し、プログラムを構成する方法を説明します。ここでは、電圧 20 V、電流 2 A を設定して、出力をオンにする場合を例に説明します。

- 1 **Initialize With Options.vi** に、**[id query]**、**[reset device]** のパラメータを渡し、**[logical name]** に NI-MAX で設定した仮想インストルメントのロジカル・ネームを指定します。

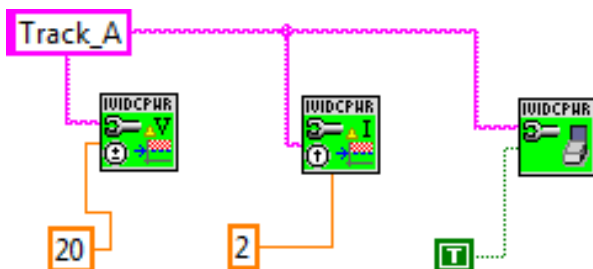
クラス・ドライバの Initialize With Options.vi では、VISA リソース名（VISA アドレス）ではなく、仮想インストルメント（IVI ロジカル・ネーム）を指定します。

この例では **[mySupply]** となります。

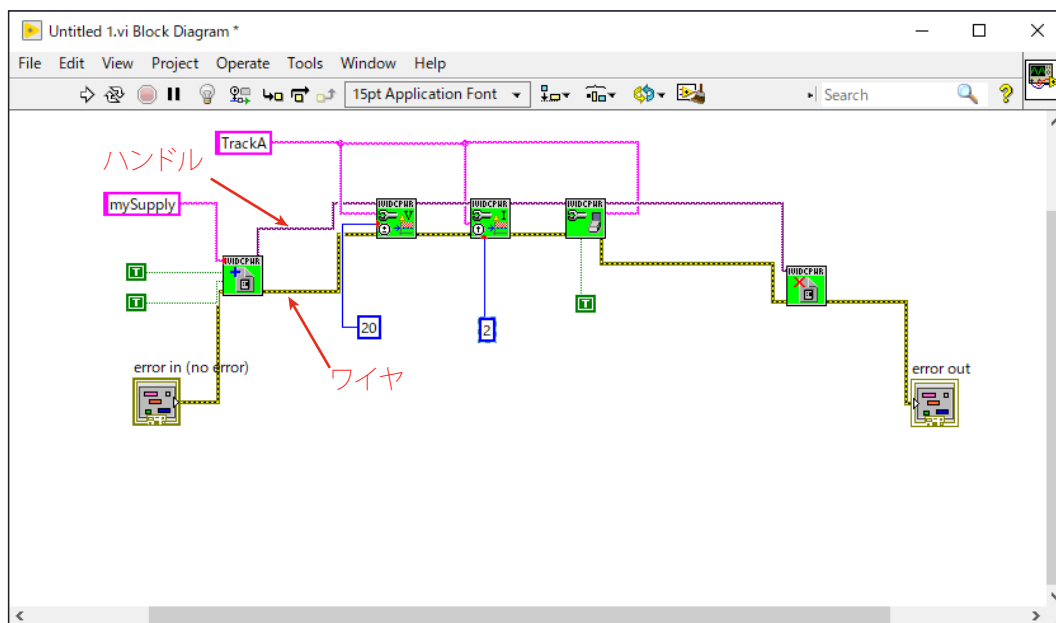


- 2 電圧、電流、アウトプットを設定するパラメータを、それぞれの VI に追加します。
- 3 制御対象の直流電源のチャンネル名の文字列を、手順 2 と同じ 3 つの VI に共通で渡します。

クラス・ドライバを使用したプログラミングでは、仮想インストルメントで設定したチャンネルのバーチャル・ネームを設定します。この例では **[Track_A]** をチャンネル名としています。



4 [error in] クラスタから [error out] クラスタをワイヤで接続し、計測器セッション (ハンドル) を接続します。



関数の解説

ここでは、プログラムを構成する関数（VI）について、その内容や設定値などを説明します。

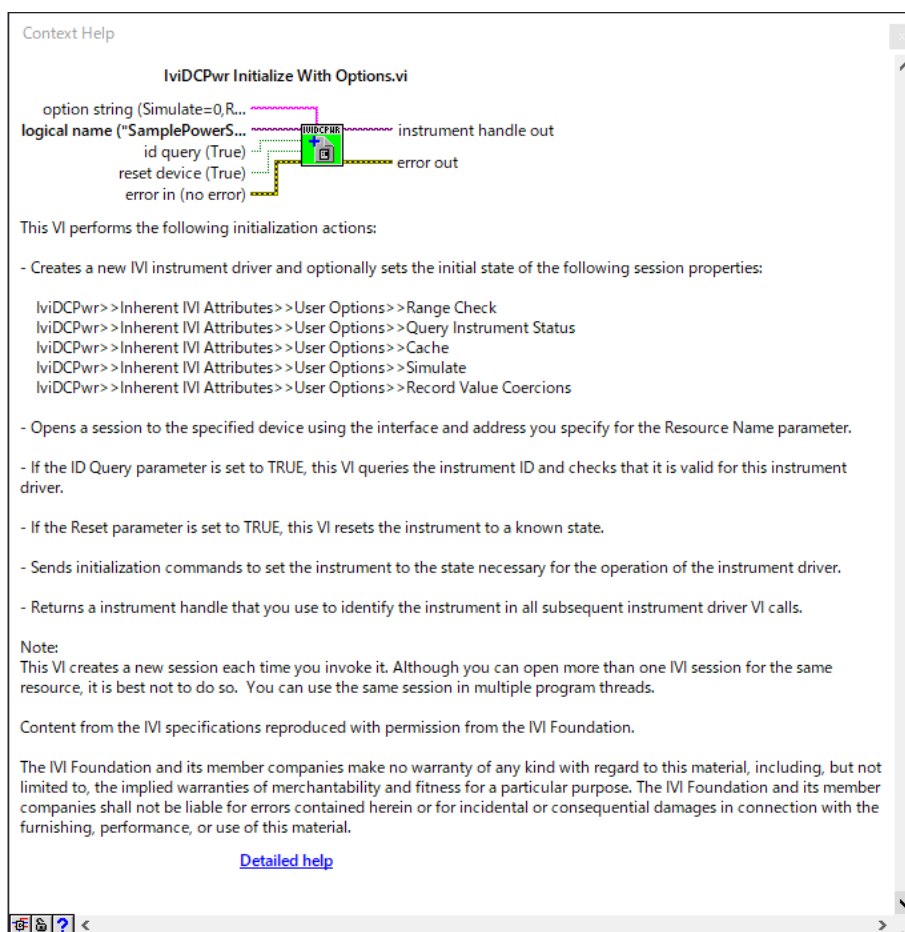
セッションの開始

クラス・ドライバでのセッションの開始には Initialize With Options.vi を使用します。

この例では、IviDCPwr クラス・ドライバを使用して説明します。VI（関数）に付くプレフィックスである「IviDCPwr」は、IviDCPwr クラス・ドライバ固有のものです。

Memo

クラス・ドライバを使用したプログラムには、kipwr（弊社製 PWR-01 シリーズ直流電源）や AgN57xx（Agilent 社製 N5700 シリーズ直流電源）といった特定機種 of 計測器ドライバへの依存がありません。



- クラス・ドライバは Initialize With Options.vi に直接 VISA アドレスを渡すことはできません。NI-MAX で作成した仮想インストルメントのロジカル・ネームを指定します。

Memo

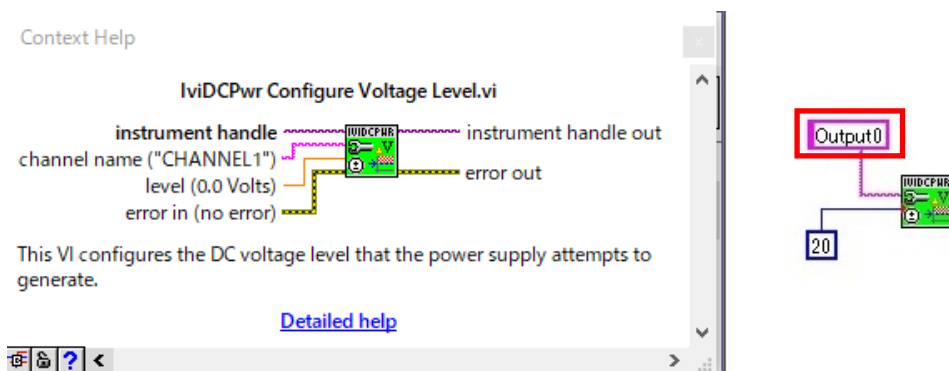
クラス・ドライバはロジカル・ネームから適切な計測器ドライバ DLL (Software Module) や VISA アドレス (Hardware Asset) を探し当て、最終的に kipwr Initialize With Options.vi 関数を間接的に呼び出します。

- Option String に渡す内容 (Cache、Range Check、Record Coercions、Interchange Check、Query Instrument Status、および Driver Setup 文字列) は、スペシフィック・ドライバを直接使用する場合と同じです。(⇒ 19 ページ)
- OptionString に渡す Cache、Range Check、Record Coercions、Interchange Check、Query Instrument Status、および Driver Setup 文字列が省略された場合のそれぞれのデフォルト値は、IVI コンフィグレーションの [Driver Session] > [General] ページで指定された値となります。

チャンネル名の設定

電源装置やオシロスコープなどの場合、IVI 計測器ドライバは複数のチャンネルが装備されていることを前提に設計されています。

そのため、計測器のパネル設定に関する操作を行うドライバ関数の多くは、[channel name] パラメータにチャンネルを指定する必要があります。



上記の図ではクラス・ドライバを使用し、チャンネル名には「Output0」という特定の計測器ドライバ（この場合は kipwr ドライバ）でのみ使用可能な名前を指定しています。

クラス・ドライバの関数に特定の測定器ドライバに依存したチャンネル名を指定しても計測器の制御を行うことはできますが、インターチェンジャビリティが損なわれます。

例えば、AgN57xx 計測器ドライバで有効なチャンネル名は「Output1」であるため、チャンネル名を「Output0」で指定している場合、プログラムを書き換えることなく計測器を AgN57xx に交換することはできません。

クラス・ドライバを使用したプログラミングでは、IVI コンフィグレーションでチャンネルのバーチャル・ネームを設定し、関数の [channel name] にバーチャル・ネームを指定することで、インターチェンジャビリティを実現できます。

例えば、「仮想インストルメントを作成する」(⇒ 24 ページ) では、バーチャル・ネームとして「Track_A」という名前を追加し、「Output0」というフィジカルネームに変換されるように設定しています。上記の Configure Voltage Level.vi の [channel name] に「Track_A」を指定することで、特定機種 of 計測器ドライバに依存しないプログラミングが行えます。

計測器ドライバを交換した場合は、IVI コンフィグレーションのいくつかの項目を変更するだけで、アプリケーション自体を変更することなく、動作を継続させることができます。(⇒ 39 ページ)

Memo

- IVI コンフィグレーションの設定情報が保存されている XML ファイル (IviConfigurationStore.xml) を手動で編集しないでください。
- IVI コンフィグレーションは、同一 PC 内のすべての 32bit/64bit 計測アプリ、およびすべてのログオンユーザで共有されます。

セッションのクローズ

計測器ドライバのセッションをクローズするには、Close.vi を使用します。

ClassPrefix Close

IVI Inherent Function

To form the *ClassPrefix* VI name, use the prefix of the class driver that you are working with. For example, if you are working with the IviDmm class driver, use the prefix IviDmm to create the correct VI name, IviDmm Close.

instrument handle

error in (no error)

error out

計測器を交換する

計測器を交換した場合は、仮想インストルメント（IVI コンフィグレーション）の Driver Session を変更するだけで、動作を継続させることができます。

アプリケーション自体を変更する必要はありません。

変更する Driver Session の設定は、以下の 3 つです。

項目	設定内容
[Hardware] タブ > [Hardware Assets] > [Resource Descriptor]	計測器の接続先 VISA アドレス
[Software] タブ > [Software Module]	使用する計測器ドライバ
[Virtual Names] タブ > [Physical Names]	仮想チャンネル名のマップ先物理名

交換した計測器に合わせて正しく設定すれば、アプリケーションを再度コンパイル・リンクすることなく動作します。

本ガイドの例では、計測器を Kikusui PWR-01 シリーズ直流電源（kipwr 計測器ドライバでホストされる計測器）から Agilent N5700 シリーズ直流電源（AgN57xx ドライバでホストされる計測器）に交換した場合、[mySupply] の設定を以下のように変更します。

項目	変更内容
[Hardware] タブ > [Hardware Assets] > [Resource Descriptor]	Kikusui PWR-01 シリーズ直流電源の接続先 VISA アドレス ⇒ Agilent N5700 シリーズ直流電源の接続先 VISA アドレス
[Software] タブ > [Software Module]	「kipwr」 ⇒ 「AgN57xx」
[Virtual Names] タブ > [Physical Names]	「Output0」 ⇒ 「Output1」

Memo

- IVI クラス・ドライバを利用したインターチェンジャビリティ機能は、計測器の交換前後での動作を保証するものではありません。交換後のシステムが正常に機能するかどうかを十分に検証してから運用してください。