



# IVI-COM 計測器ドライバ プログラミング・ガイド (IVI Config Utility 編)

Dec 2003 Revision 1.1

## 1- 概要

IVI 計測器ドライバの特徴の 1 つは、インターチェンジャビリティ機能です。インターチェンジャビリティ機能を利用すると、計測器を交換しても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させる事ができます。

インターチェンジャビリティ機能をアプリケーションから利用するには、交換する前後の両方の計測器の IVI-COM 計測器ドライバが提供されており、且つそれら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。また、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、特定の計測器ドライバのスペシフィック・インターフェースではなく、IVI 仕様で定義されたクラス・インターフェースを使用する必要があります。

インターチェンジャビリティ機能を利用する場合は、実際に使用する計測器(及びそのドライバ)の選択とその詳細設定を、アプリケーションの外部ストレージにおく必要があります。この外部ストレージを IVI コンフィグレーション・ストアと呼びます。IVI コンフィグレーション・ストアには、インストールされた計測器ドライバと仮想化された計測器の情報が保存されます。

ここでは、仮想インストルメントの設定を行うためのツールである **Kikusui IVI Config Utility** の使い方を中心に説明します。

### 1-1 仮想インストルメント

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の作成を行う前にやっておかなければならない事は、仮想インストルメントの作成です。インターチェンジャビリティ機能を実現するには、アプリケーション・コード内に特定の IVI-COM 計測器ドライバに依存した記述(例えば Kikusui4800 型で直接オブジェクトを生成したり、"GPIB0::3::INSTR"のような特定 VISA リソース名の記述など)をするべきではありません。これらの事柄をアプリケーション内に直接記述すると、インターチェンジャビリティを損ないます。

その代わりに、IVI-COM 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・ストアを置く事によってインターチェンジャビリティを実現します。アプリケーションは 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選択を間接的に行い、間接的にロードされた計測器ドライバを特定機種に依存しないクラス・インターフェースを通じてアクセスします。

IVI コンフィグレーション・ストアは通常、/Program Files/IVI/Data/IviConfigurationStore.XML ファイルで、IVI Configuration Server DLL を通じてアクセスされます。この DLL を利用するのは、主に IVI-COM 計測器ドライバと計測器ドライバのベンダーによって提供されるコンフィグレーション・ツールであって、アプリケーションからは通常は使いません。弊社の場合はコンフィグレーション・ツールとして、**Kikusui IVI Config Utility** を提供しています。これを利用する事で、仮想インストルメントの設定を行うことができます。

## Notes:

仮想インストルメントは、ロジカル・ネームによって識別されます。

本ガイドブックでは、IVI-COM Kikusui4800 計測器ドライバ(KIKUSUI PIA4800 シリーズ DC 電源コントローラ)を使用する例を示します。他機種用の IVI-COM 計測器ドライバでも、ほぼ同様の手順で使用できます。

## 2- Kikusui IVI Config Utility

### 2-1 ユーティリティの起動

**Start ボタン→Programs→IVI→Kikusui IVI Config Utility** メニューから、**Kikusui IVI Config Utility** を起動する事ができます。起動直後の画面は下の図のようになっています。画面左側はツリー・ビューになっていて、2つの階層が表示されています。上側は **Software Modules** で、この階層には現在インストールされている IVI-COM 計測器ドライバが全て表示されます。弊社製以外の、例えば Agilent Technologies 製や Tektronix 製の IVI-COM 計測器ドライバも表示されます。下側は **Logical Names** で、ここには利用可能な全ての仮想インストルメントが表示されます。下図の例では、**Kikusui4800** というロジカル・ネームで識別される仮想インストルメントが設定されています。

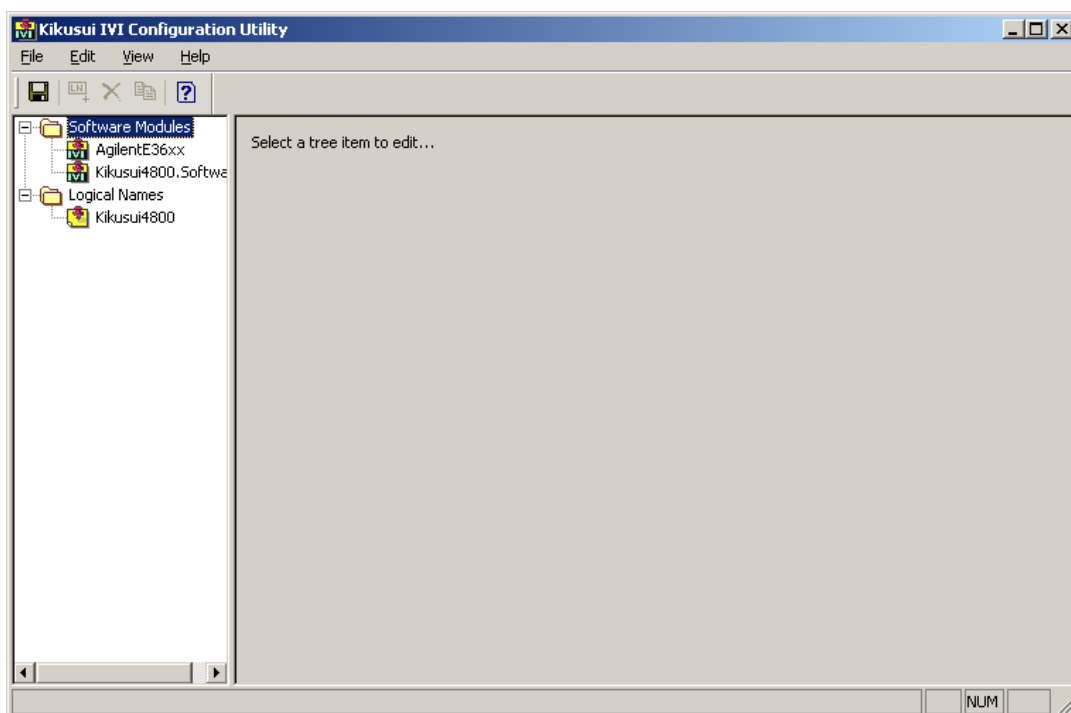


Figure 2-1 Kikusui IVI Config Utility メイン・スクリーン

ロジカル・ネーム Kikusui4800 は、Kikusui4800 IVI-COM ドライバをインストールした時に作成されるデフォルトの仮想インストルメントです。これをカスタマイズしても良いのですが、ここでは新規に作成する例を示します。

## Notes:

IVI-COM 計測器ドライバをセットアップすると、デフォルトの仮想インストルメント(ロジカル・ネーム)が作成される場合があります。これは仮想インストルメントの作成をする際のサンプルとして用意されています。ロジカル・ネームを変更して利用しても良いでしょう。

デフォルトの仮想インストルメントは、IVI-COM ドライバのベンダーやバージョンによっては提供されない場合があります。

## 2-2 ロジカル・ネームの追加

ツリー表示の上で **Logical Names** と書かれた部分を右クリックすると、コンテキスト・メニューが表示されるので、**Add Logical Name...**を選択します。すると **Add Logical Name** ダイアログが表示されます。

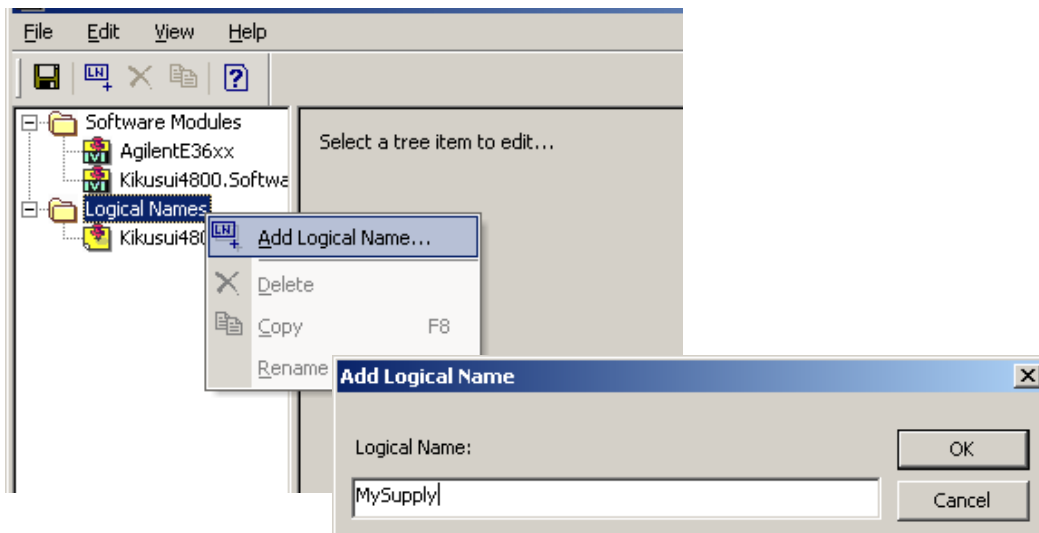


Figure 2-2 Logical Name の追加

ここでは新規作成したい仮想インストルメントの名前を指定します。ロジカル・ネームに使用できる文字は、英数字(A..Z, a..z, 0..9)、アンダースコア(\_)、及び感嘆符(!)のみです。また、既に存在するロジカル・ネームは使用できません。ロジカル・ネームはケース・センシティブ(大文字・小文字の区別あり)です。

ここでは、例として MySupply という名前を指定します。すると、この名前が **Logical Names** に追加されて表示されます。ロジカル・ネームを与えて仮想インストルメントを作成したら、**Logical Name**、**Driver Session**、**Hardware Asset**、**Virtual Names**、のタブ・ページで具体的な設定を行います。

## 2-3 タブ・ページでの設定

### Logical Name タブ

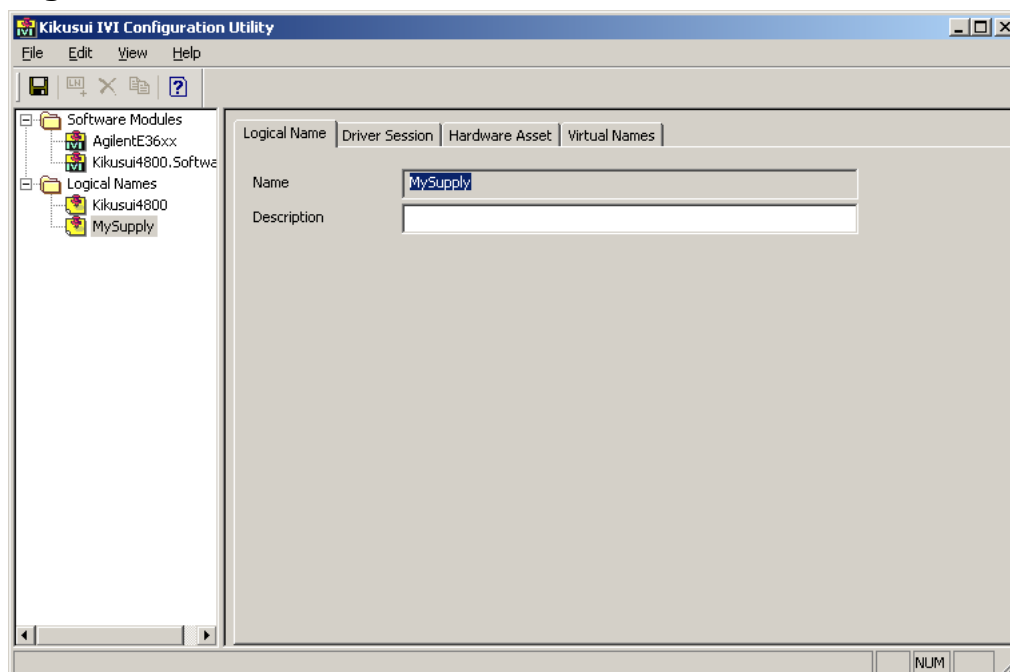


Figure 2-3 Logical Name タブ

このページで設定できるのは **Description** だけです。機能上の特別な意味は無いのでここでは省略します。空白のまま放置して構いません。

### Driver Session タブ

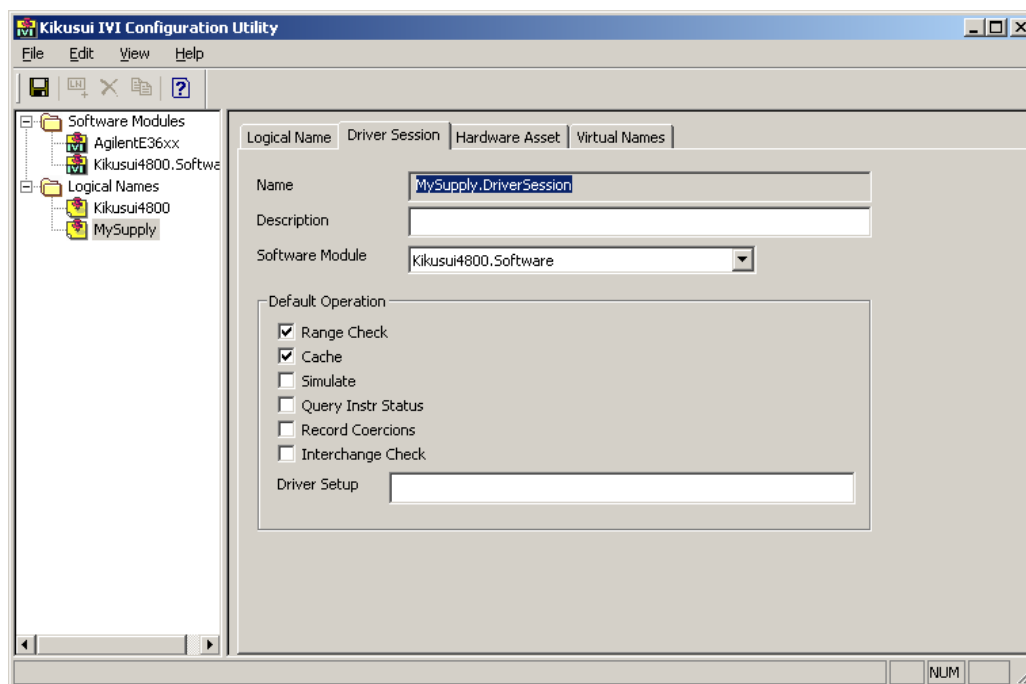


Figure 2-4 Driver Session タブ

**Description** は機能上の特別な意味は無いので空白のまま放置して構いません。**Software Module** では、コンボボックスで提示されるソフトウェア・モジュールを 1 つ選択します。これは、この

仮想インストルメントをどの計測器ドライバでホストするかを決定する重要な設定です。ここでは Kikusui4800.Software でホストしてもらうので、それを選択してください。選択肢として提示されたものの以外は指定できません。

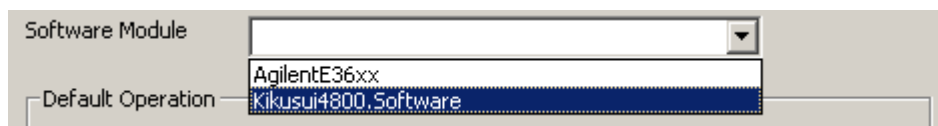


Figure 2-5 Software Module の選択

**Default Operation** では、**Range Check**、**Cache**、**Simulate**、**Query Instr Status**、**Record Coercions**、**Interchange Check**、の ON/OFF 設定、及び **Driver Setting** 文字列を設定します。ここで設定した内容は、アプリケーションが Initialize メソッドを呼び出したときのデフォルト設定となります。

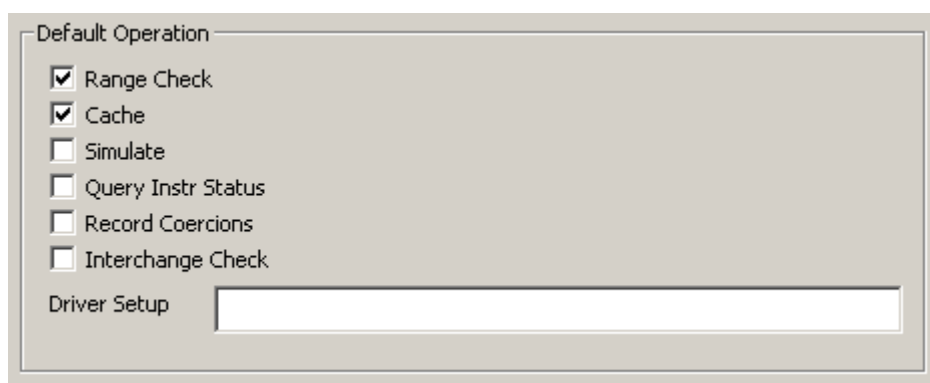


Figure 2-6 Default Operation の設定

**Range Check** が有効になっている場合、ドライバはプロパティの put\_アクセスにおける入力値とメソッド呼び出しのパラメータ値をチェックします。この機能は、計測器に対して範囲外の値を送る前の事前チェックとして行われます。アプリケーションのデバッグ時に役に立ちます。

**Cache** が有効になっている場合、ドライバは不必要な I/O を排除します。たとえば、もし或る値が計測器に既に設定されている場合、計測器に同じ設定値を再度送ることは冗長です。無駄な I/O を排除することでアプリケーションのパフォーマンスが向上します。

**Simulate** が有効になっている場合、ドライバは計測器への I/O を行わず、出力パラメータにシミュレートされた値を返します。この機能はアプリケーション開発で実際の計測器が準備できていない場合に役に立ちます。

**Query Instr Status** が有効になっている場合、ドライバは計測器への I/O を行う各メソッド呼び出し或いはプロパティへのアクセスの後に計測器への問い合わせを行います。計測器がエラーを報告した場合、ErrorQuery メソッドを使用して計測器からエラーメッセージを取得することができます。アプリケーションのデバッグ時に役に立ちますが、開発が完了したら無効に設定すればよいでしょう。

**Record Coercions** が有効になっている場合、ドライバは LONG と DOUBLE 値に関してコアーション(値の強制)記録のリストを作成します。ドライバがコアーション記録をサポートしない場合は、この設定は無視されます。

**Interchange Check** が有効になっている場合、ドライバは インターチェンジャビリティ警告の記録を作成します。ドライバがインターチェンジャビリティのチェックをサポートしない場合は、この設定は無視されます。

## Hardware Asset タブ

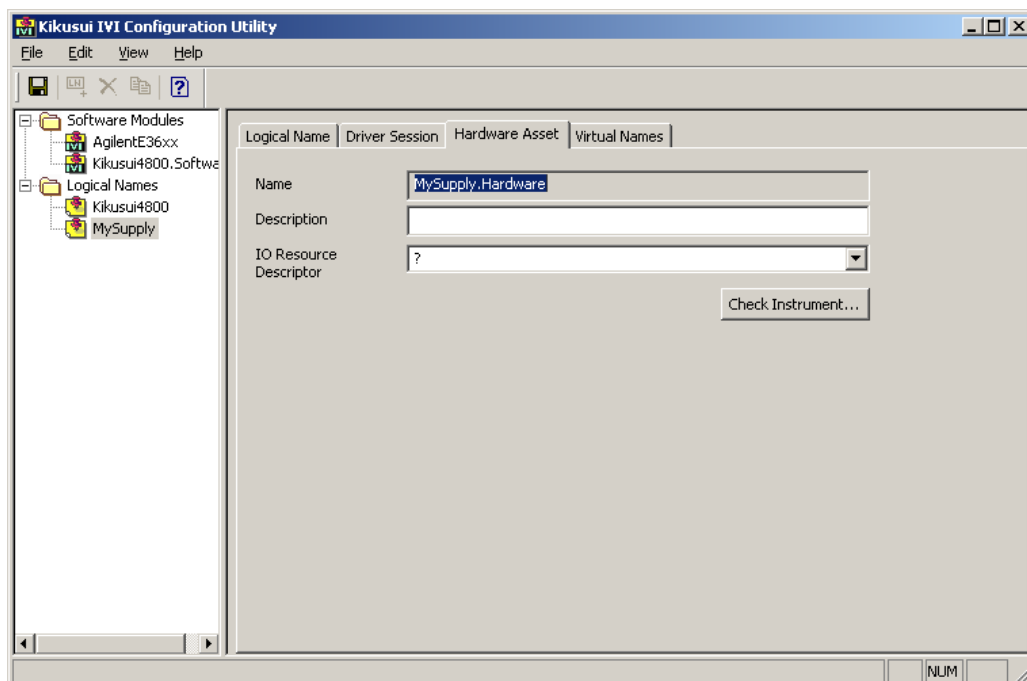


Figure 2-7 Hardware Asset タブ

**Description** は機能上の特別な意味は無いので空白のまま放置して構いません。**IO Resource Descriptor** のコンボボックスには現在使用可能な VISA I/O リソースが提示されます。選択肢の中から適切なものを選んで下さい。接続されていない計測器や TCP/IP 上の計測器などは、リストに表示されません。この場合には直接 VISA リソースを入力してください。アプリケーションがロジカル・ネームをパラメータにして Initialize メソッドを呼び出したとき、ここで設定された VISA リソースを使用して計測器との I/O を行います。

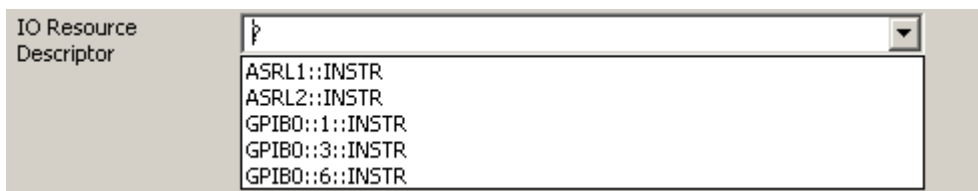


Figure 2-8 IO Resource Descriptor の選択

**Check Instrument** ボタンをクリックすると、VISA COM ライブラリを使用した簡単な I/O テストが行えます。

## Virtual Name タブ

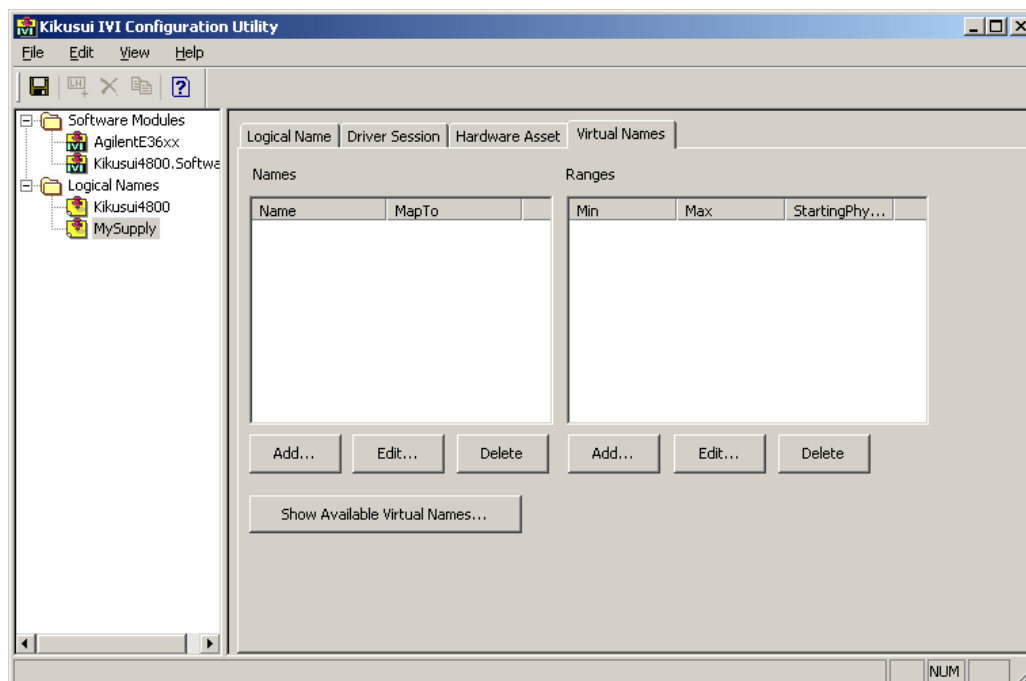


Figure 2-9 Virtual Names タブ

ここでは、バーチャル・ネームとフィジカル・ネームのマッピングを行います。これは、リピーテッド・キャパビリティを使用する場合の、オブジェクトの識別に使われます

リピーテッド・キャパビリティとは、全く同じ或いは類似する複数のオブジェクトを配列又はコンテナのように扱う概念です。例えば IviDCPwr クラスに属する計測器ドライバでは、出力チャンネルを複数装備したマルチ・トラック DC 電源装置を前提に設計されています。別の例では、IviScope クラスに属する計測器ドライバは、複数のトレース・チャンネルを装備するオシロスコープを前提に設計されています。このように、類似するオブジェクトを複数保持する場合にはリピーテッド・キャパビリティを仕様する事が IVI 仕様で推奨されています。

IviDCPwr クラスに属する Kikusui4800 IVI-COM ドライバでは、"N5!C1" のようなルールの名前で出力オブジェクトを識別しますが、このような名前は計測器ドライバ独自のものです。このようなドライバ独自の名前をフィジカル・ネームと言います。しかし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から特定ドライバに依存したフィジカル・ネームを使うわけにはいきません。そこでバーチャル・ネームを作成し、それがどのフィジカル・ネームにマップされるかを設定する必要があります。

**Names** 側にある Add ボタンをクリックすると、Virtual Name ダイアログが表示されます。

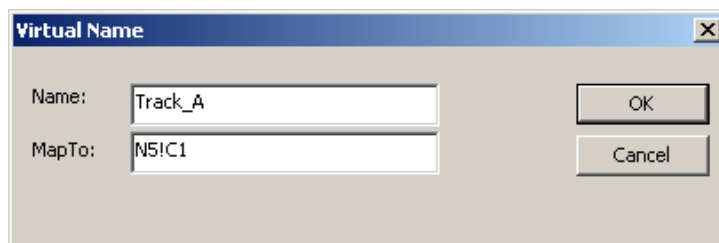


Figure 2-10 Virtual Name ダイアログ

ここでは例として、**Name** の項目に Track\_A、**MapTo** の項目に N5!C1 とタイプします。OK ボタンをクリックするとバーチャル・ネームが 1 つ出来上がりました。



この設定により、アプリケーションがリピーテッド・キャパビリティを通じて"Track\_A"のオブジェクトを参照しようとする、計測器ドライバはそのマップ先である"N5!C1"を指定されたと解釈します。これにより、計測器ドライバ内で"N5!C1"という名前が存在するオブジェクトを、"Track\_A"という名前で参照できるわけです。

同様の手順で必要な数だけマップを作成してください。同じ名前を Name に複数回指定する事はできません。全てのバーチャル・ネームは一意的な名前であればなりません。しかし、異なるバーチャル・ネームが同じフィジカル・ネームにマップする事は構いません。これは同じフィジカル・ネームに対して、複数のバーチャル・ネームでエイリアス(別名)を作った事になります。

また、全てのフィジカル・ネームに対して対応するバーチャル・ネームを作成する必要はありません。例えば、Kikusui4800 IVI-COMドライバでは"N30!C4"は有効なフィジカル・ネームですが、アプリケーションがそのオブジェクトを参照しないのであれば、それにマップされるバーチャル・ネームを作る必要はありません。

現在利用可能なバーチャル・ネームとそのマップ先フィジカル・ネームを全て表示するには、**Show Available Virtual Names** ボタンをクリックします。

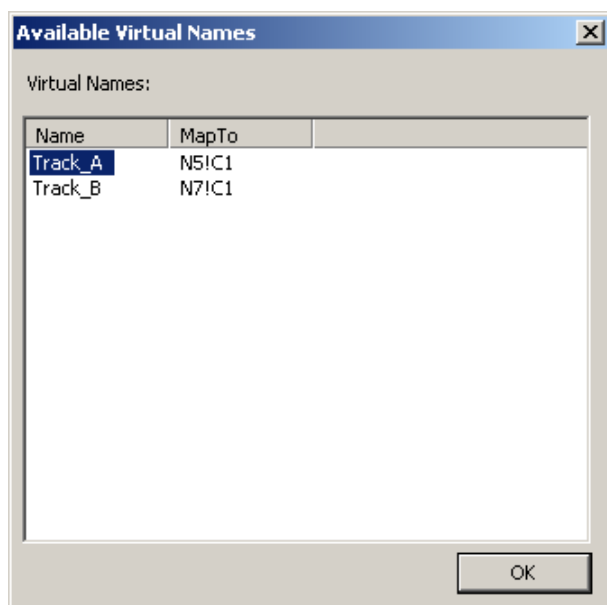


Figure 2-11 Show Available Virtual Names ダイアログ

マップ先として利用可能なフィジカル・ネームは、**Driver Session** タブの **Software Module** コンボボックスで選択されたソフトウェア・モジュール(計測器ドライバ・ソフトウェア)によって全く異なります。その情報を得るには、計測器ドライバのオンライン・ヘルプや Readme 文書を参照するか、または各言語別に用意されたプログラミング・ガイドブックに示されるサンプル・コードを使って取得してください。

## 2-4 設定の保存

仮想インストルメントの設定が完了したら、**File | Save** メニューで設定を保存してください。

### IVI-COM 計測器ドライバ・プログラミング・ガイド

本ガイドブックに登場する製品名・会社名等は各社の商標または登録商標です。

©2003 Kikusui Electronics Corp. All Rights Reserved.