

CONTENTS

Learning Aids – データセットの処理	2
イントロダクション.....	2
シナリオ.....	3
準備.....	4
Run を手動で読み込み.....	6
手動で別の Run を読み込み.....	8
手動でグラフを作成.....	9
プロシージャを使ってデータを読み込み.....	11
プロシージャを使って大量のデータを読み込み.....	13
プロシージャを使ってグラフを作成.....	14
さらに次のステップへ進みたい方は.....	16

Learning Aids – データセットの処理

Learning Aids には使い方を学ぶための Experiment が含まれています。

メニュー Help → Show Igor Pro Folder を選択して、Learning Aids フォルダ、さらに Tutorials フォルダを開きます。

本ドキュメントで使う Experiment は「Data Folder Tutorial.pxp」です。

この Experiment は、データフォルダを使って、データセットを処理する方法を示しています。これは次のような質問に対する回答として書かれたものです。

「私は、時間関数として同じ信号名（ウェーブ）を持つ多くのデータセットを読み込む必要があります。これらのファイルを開き、他のファイルの同じ信号とともにプロットして比較するプログラムが必要です。プロットには同じ信号の複数のインスタンスが表示されるため、セット名によってプロット上で信号を識別する必要があります。ただ、各セット内では信号は同じ名前を持っています。」

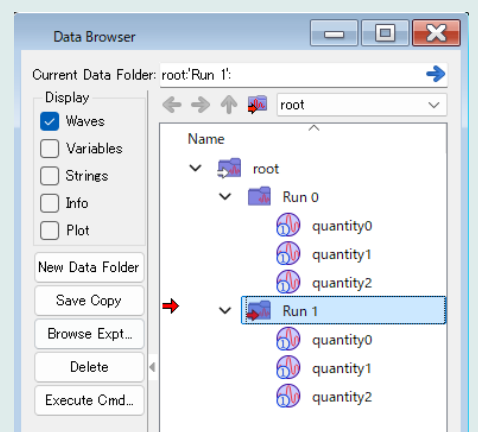
イントロダクション

Igor のデータフォルダは、ファイルシステムフォルダのようなものですが、ファイルシステムフォルダがディスク上のファイルを格納するのに対して、Igor のデータフォルダはデータ（ウェーブ、数値変数、文字列変数）をメモリに格納します。

ここでは、Igor データフォルダを「データフォルダ」と呼び、ファイルシステムフォルダを単に「フォルダ」と呼ぶことにします。

Igor の Data Browser を使って、データフォルダを表示、操作することができます。

次の図は、Run 0 と Run 1 と名付けられた2つのデータセットを示しています。各セットは、それぞれ独自のデータフォルダに保存され、quantity0、quantity1、quantity2 と名付けられた3つの測定量で構成されています。



このチュートリアルでは、このようなデータセットをメモリにロードする方法と、異なるセットから対応するデータを比較する方法を説明します。

例えば、Run 0 から quantity0、Run 1 から quantity0 のグラフを作る方法を説明します。

ここでは、手動でプロセスを実行する方法を説明し、その後、Igor プロシージャを使ってプロセスを自動化する方法を示します。

Igor プロシージャは理解しやすいように、できるだけシンプルになるようにしました。

これらは Igor プログラマーがさらにカスタマイズできる出発点を提供します。

シナリオ

1 時間に 1 回、5 時間継続する実験を実行していると仮定します。
各時間帯のデータは「Run」（ラン）と呼ばれるセットです。
各 Run で 3 つの測定値が生成され、それらは 1 つのファイルに保存されます。
したがって、ディスクに保存された 1 日分のデータは次のようになります。



	quantity0	quantity1	quantity2
0	1	2	3
1	2	3	4
2	3	4	5
3	4	5	6
4	5	6	7
5	6	7	8
6	7	8	9
7	8	9	10
8	9	10	11
9	10	11	

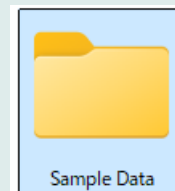
5 つのデータセット（Run 0 から Run 4）を Igor に読み込み、5 つすべての Run から quantity0 を示すグラフを作成します。

準備

これから、上で説明したサンプルデータファイルを作成します。
まず、サンプルファイルを保存するフォルダーを作ります。

Data Folder Tutorial の Experiment が開いているところから始めます。

1. デスクトップなど任意の場所に Sample Data という名前のフォルダーを作ります。



2. Igor に戻ります。

Macro メニューから **Create Sample Data Files** を選択します。

これにより、サンプルファイル用のフォルダーを選択するよう促すダイアログが表示されます。

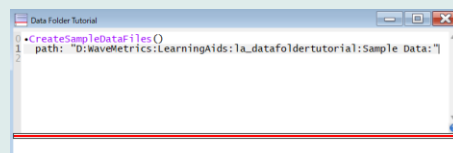
Browse ボタンをクリックするとサブダイアログが表示されます。

Sample Data フォルダーがハイライトされるまで移動し、**OK** ボタンをクリックします。

これにより、メインダイアログに戻ります。

Macros	Windows	Notebook	Misc
Create Sample Data Files			Ctrl+0
Load Data Set From File			Ctrl+1
Load Data Sets From Folder			Ctrl+2
Plot Data From Selected Sets			Ctrl+3

3. CreateSampleDataFiles プロシージャは、Sample Data フォルダーに5つのファイルを作成します。



4. Sample Data フォルダーを見てください。

「Run 0.dat」から「Run 4.dat」という名前の5つのファイルが表示されているはずです。

名前
Run 0.dat
Run 1.dat
Run 2.dat
Run 3.dat
Run 4.dat

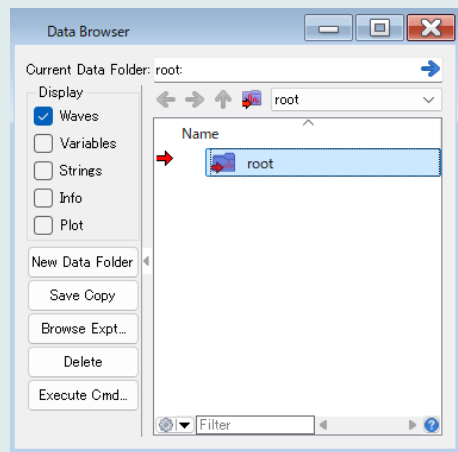
5. Igor に戻り、Data メニューから Data Browser を選択して、Data Browser を開きます。

root という名前のデータフォルダーが 1 つ表示されるはずです。

root データフォルダー以外のものが見える場合は、以前の実験で使ったものが残っている状態です。

その場合、root データフォルダーをクリックして、左の Delete ボタンをクリックして残っているデータを削除します。

本当にデータを削除してよいかを聞いてきますので、削除してください。

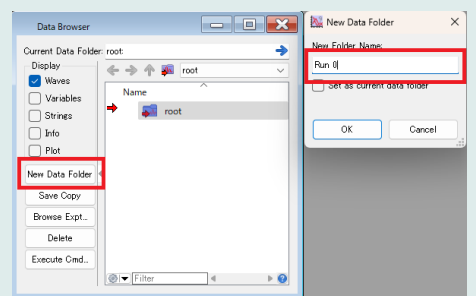


Run を手動で読み込み

これから、Run の1つを Igor のデータフォルダーに読み込みます。

1. Data Browser で、New Data Folder ボタンをクリックします。

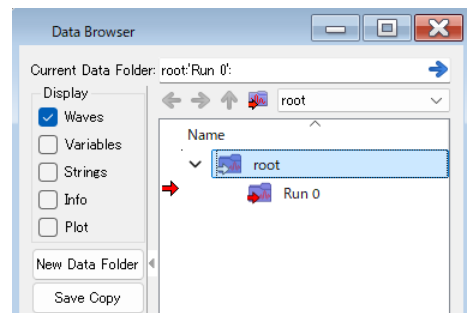
表示されるダイアログに「Run 0」と入力し、OK ボタンをクリックします。



2. Data Browser で、root データフォルダーの隣にある赤い矢印を「Run 0」データフォルダーまで下にドラッグします。

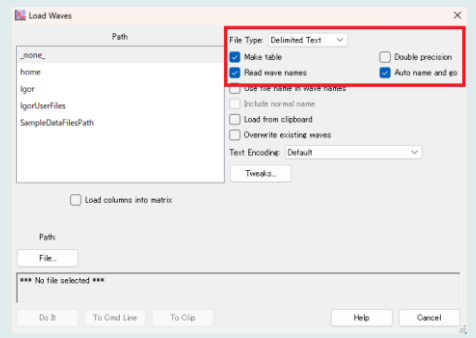
これで「Run 0」データフォルダーが「Current data folder」になりました。

つまり、Igor が新しいウェーブを作るときには、明示的に指示しない限り、「Run 0」データフォルダー内のウェーブを検索することになります。



3. Ctrl+L を押して、Load Waves ダイアログを表示します。

File Type のポップアップメニューが Delimited Text に設定され、Make Table、Read Wave Names、Auto name and go のチェックボックスがチェックされ、その他のチェックボックスがすべてオフになっていることを確認します。



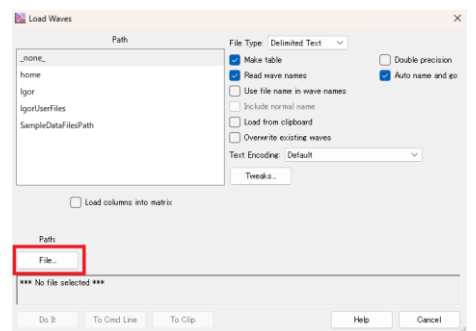
4. File ボタンをクリックします。

Open File ダイアログが表示されます。

プルダウンメニューから All Files を選択する必要がある場合があります。

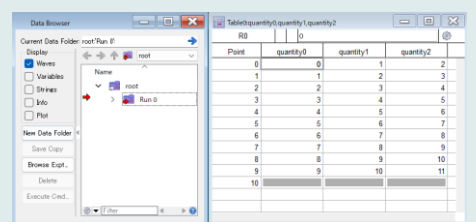
Sample Data フォルダーにある Run 0 ファイルを見つけ、「開く」ボタンをクリックします。

これで Load Waves ダイアログに戻ります。



5. Do It ボタンをクリックします。

これにより、Run 0 ファイル内のデータが Run 0 データフォルダーに読み込まれ、テーブル形式で表示されます。

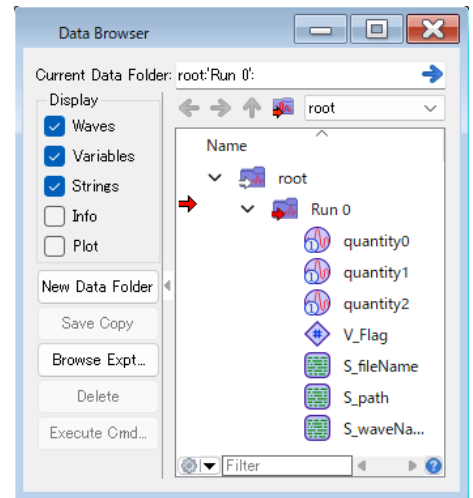


6. Data Browser ウィンドウをクリックします。

「>」記号をクリックして、Run 0 データフォルダーを開きます。

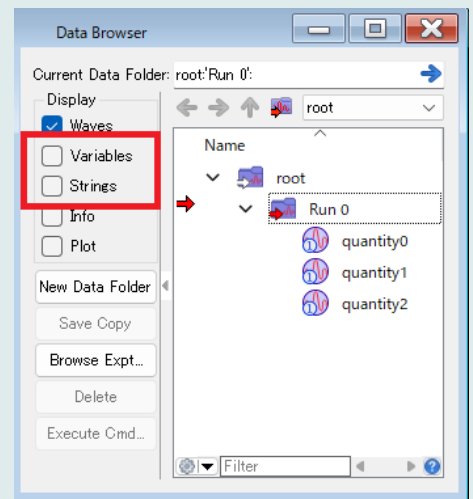
これで、Run 0 データフォルダーの内容が表示されます。
このフォルダーには、quantity0、quantity1、quantity2 の3つのウェーブが含まれているはずです。

Data Browser では、ウェーブがアイコンで表示され、テーブルではウェーブが数値の列として表示されます。



7. Data Browser には、先ほど実行した LoadWave コマンドで作成された数値、文字列変数のアイコンも表示されているかもしれませんがありません。

今回の操作では、これらのアイコンは邪魔なので、Data Browser ウィンドウの Variables と Strings のチェックボックスを外して、表示されないようにしておきます。



8. テーブルをクリックし、右上のウィンドウを閉じる「X」をクリックします。

表を保存するかどうか聞いてきますので、「No Save」ボタンをクリックします。

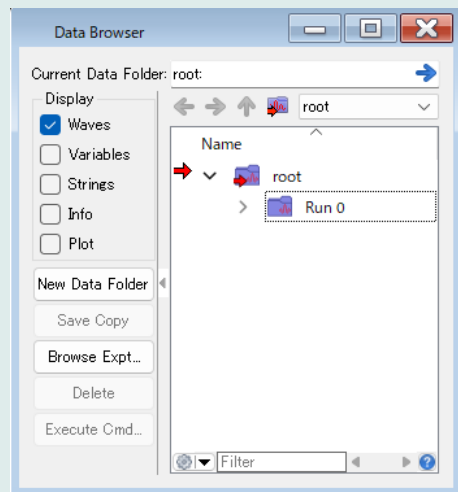
Data Browser ウィンドウに3つのウェーブがまだ表示されていることに注目してください。

テーブルを閉じてもデータが残っていることを示しています。

手動で別の Run を読み込み

Run 0 というデータセットを1つ読み込みました。
それでは、別のデータセットを読み込む方法を説明します。

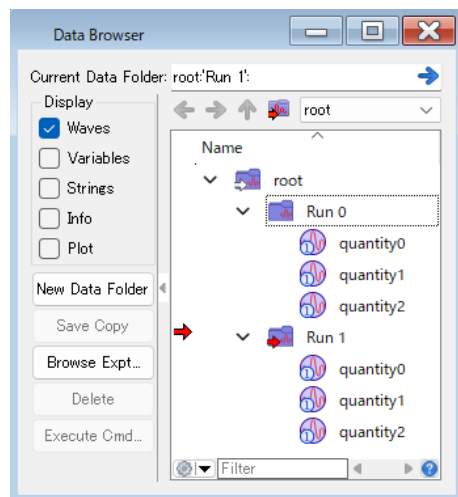
1. Data Browser で、赤い矢印を root データフォルダーまでドラッグして、Current data folder にします。



2. 前の「Run を手動で読み込み」の手順を繰り返します。ただ、「Run 0」と書かれた部分を「Run 1」と読み替えます。

完了したら、ここに戻ります。

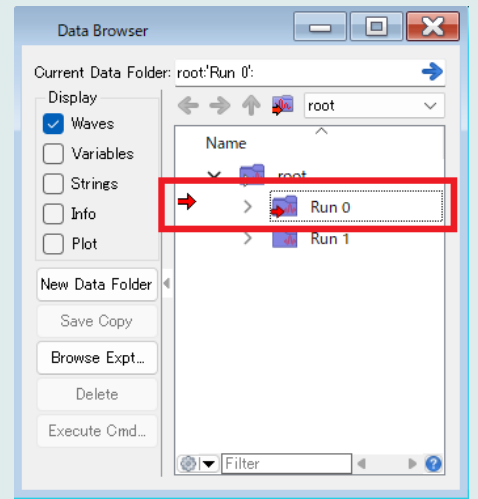
この時点で、Data Browser ウィンドウは次ようになっているはずです。



手動でグラフを作成

2つのデータセットが揃ったので、両方のセットから quantity0 を示すグラフを作成します。

1. Data Browser で、赤い矢印を「Run 0」データフォルダーまでドラッグします。

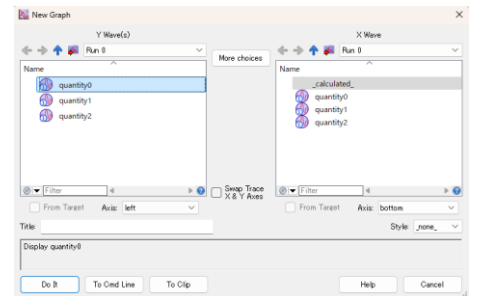


2. Windows メニューから New Graph を選択します。

表示されるダイアログには、現在のデータフォルダーからのウェーブのみが表示されます。

Y Waves リストで quantity0 をクリックし、Do It をクリックします。

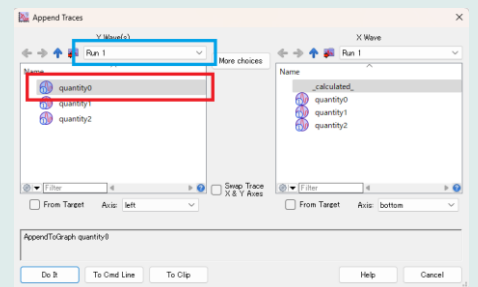
これにより、フルパスが root:'Run 0':quantity0 であるウェーブのグラフが作成されます。



3. Data Browser で、赤い矢印を「Run 1」データフォルダーまでドラッグします。

グラフをクリックし、Graph メニューから Append Traces to Graph を選択します。

表示されるダイアログには、Current data folder からのウェーブのみが表示されます。

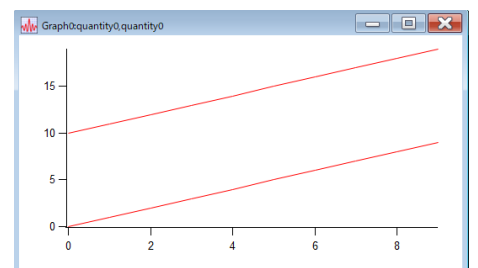


将来的には、コマンドラインから次のコマンドを実行して、このグラフを作ることを好むようになるかもしれません。

```
Display root:'Run 0':quantity0, root:'Run 1':quantity0
```

「Run 0」と「Run 1」の周りにシングルクォートが使われていることに注意してください。

これは、これらのデータフォルダー名にスペースが含まれているためです。



クオートを使わないと Igor は「Run 0」が1つの名前であることが理解できません。

4. グラフウィンドウを閉じます。

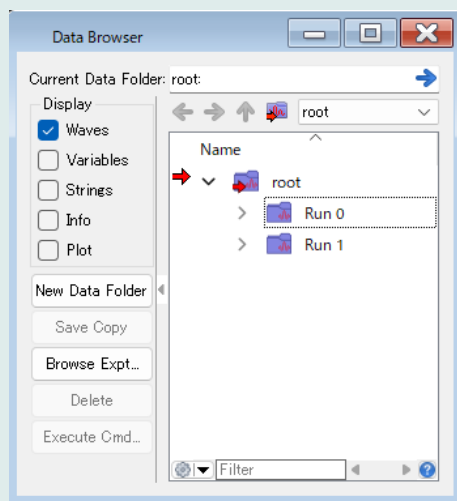
保存するかどうかを聞いてきたら、No Save をクリックします。

プロシージャを使ってデータを読み込み

実際には大量のデータを生成するので、データの読み込みとグラフの作成の処理を効率化できるのが望ましいです。Igor のプロシージャを書くことで、これを実現できます。実は、ここまでですで行ってきています。

1. Data Browser をクリックします。

赤い矢印を root データフォルダーにドラッグして Current data folder にします。



2. Macros メニューから、Load Data Set From File を選択します。

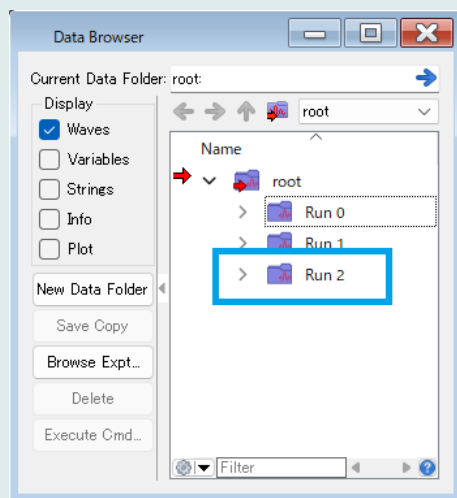
これにより、ファイルを選択するダイアログが開きます。プルダウンメニューから All Files を選択する必要がある場合があります。

Macros	Windows	Notebook	Misc
Create Sample Data Files			Ctrl+0
Load Data Set From File			Ctrl+1
Load Data Sets From Folder			Ctrl+2
Plot Data From Selected Sets			Ctrl+3

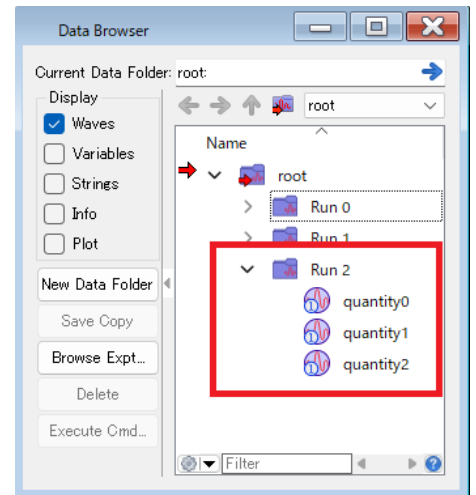
Sample Data フォルダーまで移動して、「Run 2.dat」をクリックして、「開く」ボタンをクリックします。

3. Data Browser で Load Data Set From File プロシージャが「Run 2」という名前の新しいデータフォルダーを作ったことに注目してください。

このプロシージャでは、新しいデータフォルダーの名前として、選択したファイル名が使われます。



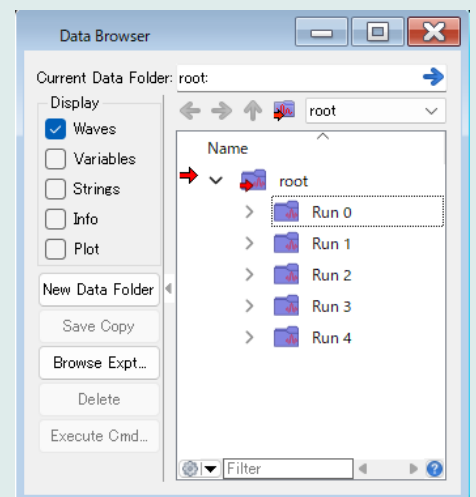
4. Data Browser で「Run 2」アイコンを開き、新しいデータフォルダーに quantity0、quantity1、quantity2 の3つのウェーブが含まれていることを確認します。



5. 「Run 3.dat」ファイルを読み込むために、Load Data Set From File プロシージャをもう一度使います。

「Run 4.dat」ファイルを読み込むために、Load Data Set From File プロシージャをもう一度使います。

この時点で Data Browser ウィンドウは次のようになっているはずです。



プロシージャを使って大量のデータを読み込み

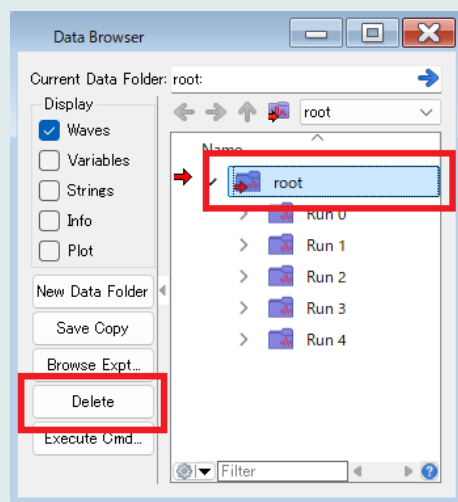
次のプロシージャ Load Data Sets From Folder は、特定のフォルダー内のすべてのデータファイルを一度に読み込みます。

Igor にある既存のデータフォルダーをすべて削除して、このプロシージャを実行して、すべてのデータをもう一度読み込んでみます。

1. Data Browser で root アイコンをクリックします。

左の Delete ボタンをクリックします。

Igor が本当にデータを削除してよいか聞いてきますので、そのまま削除してください。

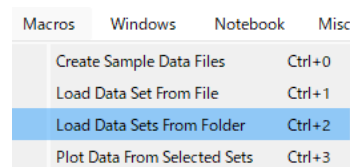


2. Macros メニューから、Load Data Set From Folder を選択します。

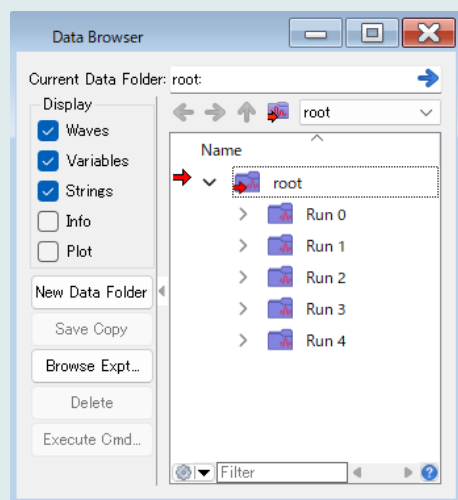
これにより、Choose Folder Containing Data Files ダイアログが開きます。

Choose Folder Containing Data Files ダイアログには、すでに Sample Data フォルダへのパスが表示されているはずです。
(表示されていない場合は、Browse ボタンを押して選択します)

OK ボタンをクリックします。



3. この時点で、Data Browser には再び5つのアイコン、Run 0、Run 1、Run 2、Run 3、Run 4 が表示されるはずです。



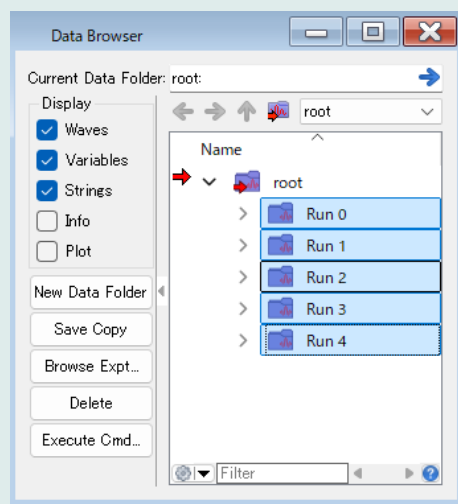
プロシージャを使ってグラフを作成

プロシージャ Plot Data From Selected Sets は、プロシージャが選択したデータセットに対して、事前に定義されたグラフを作成します。

1. Data Browser で Run 0 アイコンをクリックします。

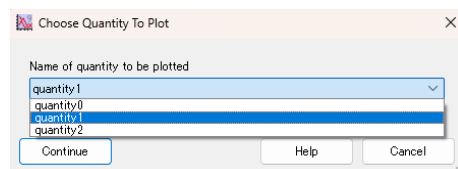
Ctrl-クリックで、Run 1、Run 2、Run 3、Run 4 を選択します。

5つすべてのデータフォルダーがハイライトされているはずですが。



2. Macro メニューから Plot Data From Selected Sets を選択します。

これにより、プロットする quantity の名前を入力するダイアログが表示されます。



実行してみると、グラフも表示されず、エラーメッセージも出ません。

3. 実はデフォルトの状態 (バージョン 9 での) では、コードが古い MacOS 対応の部分が残っていて、プロットが表示されません。

そのため、3か所修正する必要があります。

メニュー Windows → Procedure Windows → Procedure Window を選択します。

下記のそれぞれの場所を探して、追加、修正してください。

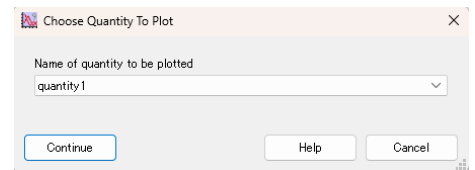
旧	新
<pre>leafName = "" pos = strlen(pathStr) - 1 do if (CmpStr(pathStr[pos], ":") == 0 leafName = pathStr[pos+1, strlen(pathStr)] break endif pos -= 1 while (pos > 0)</pre>	<pre>leafName = "" pos = strlen(pathStr) - 1 if (CmpStr(pathStr[pos], ":") == 0) pathStr=pathStr[0,pos-1] endif do if (CmpStr(pathStr[pos], ":") == 0) leafName = pathStr[pos+1, strlen(pathStr)] break endif pos -= 1 while (pos > 0)</pre>
<pre>// If either of the input parameters is "", put up a dialog to get file.</pre>	<pre>// If either of the input parameters is "", put up a dialog to get file.</pre>

<pre> if ((strlen(pathName)==0) % (strlen(pathName)==0) if (strlen(dataSetNameStr) == 0) Print "PlotDataFromSelectedSets bug" return -1 endif waveNameStr = dataFolderPathStr + ":" + quantityName if (i == 0) titleStr = quantityName Display/W=(5,42,595,366) \$waveNameStr as titleStr else </pre>	<pre> if ((strlen(pathName)==0) (strlen(pathName)==0)) if (strlen(dataSetNameStr) == 0) Print "PlotDataFromSelectedSets bug" return -1 endif waveNameStr = dataFolderPathStr + quantityName if (i == 0) titleStr = quantityName Display/W=(5,42,595,366) \$waveNameStr as titleStr else </pre>
--	---

保存しようとする、Experiment がパッキングされていて保存できないため、メニュー File → Save Experiment As を選択して、Experiment 全体を別の場所に保存してください。その後、次のステップに進みます。

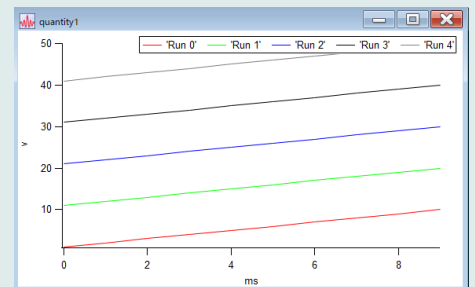
4. 再度、Macro メニューから Plot Data From Selected Sets を選択します。
ポップアップメニューから quantity1 を選択し、Continue ボタンをクリックします。

5つのデータセットのそれぞれからの quantity1 を示す新しいグラフが作成されます。



5. 右図のように、5つのデータセットのそれぞれから quantity1 を示す新しいグラフが作られているはずですが。

グラフの凡例は、quantity1 の各トレースがどのセットから来たのかを示しています。



さらに次のステップへ進みたい方は

ここまでの部分はなかなか良いスタートですが、このプロシージャでサポートされていないことをやりたくなることは間違いありません。

例えば、このプロシージャでは、拡張子が「.dat」で、ファイルの最初の行に数量名が記載されたタブ区切りテキストファイルがあり、数量を XY ペアではなく、X 単位を秒、データ単位をボルト（電圧）としてウェーブフォームで表示したいと想定しています。

Igor プログラミングに詳しい方のために、いくつかのヒントを紹介します。

Procedure Window を開いて、コードを読みます。

多くのコメントがあり、シンプルかつ論理的に保つよう書かれています。

おそらく、データ読み取り部分をより洗練させる必要があるでしょう。

データファイルのヘッダーデータをスキップする必要がある場合は、LoadWave/J ではなく LoadWave/G を使うか、LoadWave に読み込みを開始する正確な場所を指示する他のフラグを使うことができます。

データファイルに「.dat」拡張子がない場合は、プロシージャで「.dat」と書かれた部分を特定し、必要に応じて書き換えてください。

データファイルがテキストではない場合は、バイナリファイルを読み込む GBLoadWave などの別のファイルローダーを使う必要があります。

ファイルに意味のある列名がない場合は、LoadWave または他のファイルローダーにデフォルト名を読み込ませ、その後、名前変更コマンドを使って、意味のある名前を付けることができます。

または、LoadWave /B フラグを使って、各列に名前を指定することもできます。

SetWaveUnits 関数を、自分にとって意味のある単位がつくように変更したいとも思うはずですが。

データが XY データの場合は、いくつかの変更が必要になります。

例えば、PlotDataFromSelectedSets 関数内の Display および AppendToGraph コマンドを変更する必要があります。

LoadDataSetsFromFolder 関数を賢くして、フォルダー内にはあるが読み込みたいデータを含まないファイルをスキップできるようにする必要があるかもしれません。

その場合、ファイル名に StrSearch 関数を使って、データファイルであることを示すパターンが含まれているかどうかを確認できるかもしれません。

GetQuantityNameDialog マクロを、quantity0、quantity1、quantity2 ではなく、意味のある数量名をつくように変更したいかもしれません。

TraceColor 関数を呼び出す ModifyGraph コマンドの代わりに、スタイルマクロを作成したいかもしれません。適切なスタイルマクロを作成し、ModifyGraph を置き換えたら、TraceColor 関数を削除できます。

より深くネストされたデータフォルダー階層を使うことができます。

例えば、ある日のすべてのランを、取得日の名前のデータフォルダーに保存することができます。

あるいは、データを Experiment に恒久的に保存するのではなく、ロードし、処理した後、破棄するほうが良いという場合もあるかもしれません。

複数の Experiment で同じプロシージャを使いたい場合は、それらを User Procedures フォルダー内の別のプロシージャファイルに保存する必要があります。マニュアルの Procedure Windows の項を参照してください。