AppleScript を使う

このガイドには、AppleScript を使用し KaleidaGraph Macintosh 版を自動化するために役立つ情報と例題が記載されています。Apple Event エンジンから KaleidaGraph をどのようにコントロールするかを実演する、基本的なものから複雑なものまで様々なサンプルスクリプトが取り上げられています。

始める前に、以降のセクションは AppleScript の使用方法について解説することを目的としたものでないことをご理解下さい。AppleScript に慣れていない場合は、AppleScript について充分に解説している参考書や Web サイトを読まれることをお勧めします。

KaleidaGraph は Apple Event により動作可能であり、KaleidaGraph 自体が Apple Event エンジンでないことを理解する必要があります。KaleidaGraph や他のアプリケーションをコントロールする唯一の方法は、AppleScript のような Apple Event エンジンを用いることです。

1.1 AppleScript の基本

次の用語は、このガイドの例題すべてにわたり使用されます。

- **整数** - ゼロを含む、正の整数（0、1、2、3 ...）
- **文字列** - 引用符 （"）で囲まれた 255 までの文字列
- **ブール** - yes/no/true/false
- **空白類文字** - スペース /tab

また、次の AppleScript コマンドも例題で使用されています。これらのコマンドに詳しくない場合は、このセクションを続ける前に理解されることをお勧めします。

activate	choose file	delay	if	list folder	result
& （アンパサンド）	contains	display dialog	info for	open document file	set
as	copy	get	keystroke	repeat	tell

コメント は、2 つのうちどちらかの方法でスクリプトに追加できます。最初の方法は、コメントの前に 2 つのハイフン （--）を置くことです。これは一行コメントに役立ちます。

2 つ目の方法は、コメントの最初と最後に “(*” と “*)” の文字を置きます。これは複数行のコメントに役立ちます。

AppleScript エディタで文字列に **引用符** （"）を使用したいときは、それぞれの引用符の前にバックスラッシュ （\）を置かなければなりません。この理由は、引用符がテキスト文字列の最初と最後を示すためです。たとえば、テキスト文字列 "file=\"filename\"" には文字列の引用符の前にバックスラッシュが含まれています（日本語入力では円記号）。

1.2 KaleidaGraph の用語説明

1.2.1 概要

AppleScript には、アプリケーションがスクリプトで使用するためにサポートしている固有のイベントを表示する方法があります。これは、AppleScript エディタで **ファイル** > **用語説明を開く** を選び、目的のアプリケーションを選択することで表示できます。アプリケーションがサポートしている全てのイベントがダイアログに表示されます。特定のイベントを選択し、これらに関する一般的な情報を得ることができます。これは Finder と System Events (スクリプト機能を拡張するためのコマンドが提供される) の用語説明を開く際にも役立ちます。

KaleidaGraph によってサポートされるイベントは 2 つのカテゴリーに分類されます。最初のカテゴリーは、イベントに必須のセット（Required Suite）で、どのアプリケーションでもサポートしている一般的なイベントです。Run、Open、Print、Quit イベントは、すべて Required Suite の一部とみなされます。

2 つめのカテゴリーに分類されるイベントは KaleidaGraph Events です。これらは、KaleidaGraph アプリケーションのみに関連するイベントです。これらのイベントの例として、AppendColumn、RunTextScript、Rebuild が含まれます。

以降のセクションのイベントは、アルファベット順でリストされています。これは、用語説明でリストされている順番と異なることに注意してください。

この例題で使用する構文は、次のとおりです。

イベント - デフォルト設定を含む、イベントの記述

イベント　パラメータ

[適用されるオプションパラメータ]

[適用されるイベント実行結果]

イベントの使用例

注意：　オプションパラメータは、必須パラメータと区別するために括弧で括られます。スクリプトによって返される結果も、イベントやそのパラメータから分離するために括弧で括られます。

1.2.2 イベントのリスト

AppendColumn - このイベントは最前面のデータウィンドウにあるデータに１つ以上のデータ列を追加します。データは、タブ区切りの文字列としてストアする必要があります。文字列の開始行には、任意にタイトルを含むことができます。デフォルトでは、開始行にデータが含まれているとみなされます。

AppendColumn 文字列

[Titles ブール]

```
set item1 to "1.00	1.40	6.20
1.50	4.50	5.55"

set item2 to "Time	Test #1	Test #2
1.00	1.40	6.20
1.50	4.50	5.55"

tell application "KaleidaGraph"
	GetData "-1"
	set string1 to result
	AppendColumn string1
	SelectWindow "Data 2"
	AppendColumn item1
	SelectWindow position 3
	AppendColumn item2 with Titles
end tell
```

AppendRow - このイベントは最前面のデータウィンドウにあるデータの下に１つ以上のデータ行を追加します。データは、タブ区切りの文字列としてストアする必要があります。文字列の開始行には、任意にタイトルを含むことができます。デフォルトでは、開始行にデータが含まれているとみなされます。

AppendRow 文字列

[Titles ブール]

```
set item1 to "1.00	1.40	6.20
1.50	4.50	5.55"

set item2 to "Time	Test #1	Test #2
1.00	1.40	6.20
1.50	4.50	5.55"

tell application "KaleidaGraph"
	GetData "-1"
	set string1 to result
	AppendRow string1
	SelectWindow "Data 2"
	AppendRow item1
	SelectWindow position 3
	AppendRow item2 with Titles
end tell
```

Close - このイベントは 一つのウィンドウを閉じます。ウィンドウは、タイトルバーの名前や位置（前面から背面方向で最前面ウィンドウはポジション 0 です）による指定がない限り、デフォルトで最前面のウィンドウが閉じます。デフォルトの動作では、何か変更があったときに保存を尋ねますが、保存オプションを使用してその内容を保存せずにウィンドウを閉じたり、自動的に変更を保存することもできます。

Close 文字列

[position 整数]

[saving yes/no/ask]

```
tell application "KaleidaGraph"
	Close "Data 1"
	Close position 1
	Close without saving
	Close position 3 with saving
end tell
```

CloseAllWindows - ウィンドウの内容を保存せずに、表示されているすべてのウィンドウを閉じます。

CloseAllWindows

```
tell application "KaleidaGraph"
	CloseAllWindows
end tell
```

CloseFrontWindow - ウィンドウの内容を保存せずに、最前面のウィンドウを閉じます。

CloseFrontWindow

```
tell application "KaleidaGraph"
	CloseFrontWindow
end tell
```

ExportPlot - このイベントは最前面のプロットをファイルまたはクリップボードに書き出します。(KaleidaGraph が最前面のアプリケーションであることが条件）デフォルトでは、ファイルは現在のディレクトリにストアされます。SetRefDirectory コマンドを使用するか、ファイル名にパスを含めることで異なる場所を指定できます。ファイルの属性は文字列の一部としてセットできます。これは、ヘルプファイルおよび KaleidaGraph マニュアルの 12.2.2 節で解説される数式スクリプトコマンド #PICT/OPT の下に記載されています。

ExportPlot 文字列

[Name エイリアス]

[Clipboard ブール]

[Result: 画像]

```
set item2 to "
type = tiff
dpi 600
cmyk yes
file = \"Sample TIFF.tif\""

tell application "KaleidaGraph"
	activate
```

```
        ExportPlot item2
        ExportPlot with Clipboard
end tell
```

GetData - このイベントは最前面のデータウィンドウで選択を行い、タブ区切りの文字列としてデータをストアします。選択範囲は空白類文字で区切られた数字を含む文字列で、順序は （開始行、終了行、開始列、終了列）です。例外として、開始行に -1 を入力した場合は、ウィンドウ全体が選択されます。0 で始まる行と列のポジションのアドレス選択は、データウィンドウ（ポジション 0, 0）の左上隅からカウントされます。

GetData 文字列

[Result: 文字列]

```
tell application "KaleidaGraph"
        GetData "-1"
        GetData "0 100 1 3"
end tell
```

GetWindowName - このイベントは 1 つ以上のウィンドウの名前を返します。ウィンドウポジション（前面から背面方向で、最前面のウィンドウがポジション 0 です）が与えられた場合、ウィンドウの名前は文字列として返されます。それ以外は、すべてのウィンドウの名前がレコードに返されます。

GetWindowName

[position 整数]

[Result: 文字列またはレコード]

```
tell application "KaleidaGraph"
        GetWindowName
        set record1 to result
        GetWindowName position 1
        set string1 to result
end tell
```

LoadPlotScript - このイベントは指定したプロットスクリプトファイルを読み込みます。デフォルトでは、コマンドは KaleidaGraph プリファレンスフォルダ内にあるスクリプトを探します。他のフォルダから開くには、Open コマンドを使用してください。

LoadPlotScript 文字列

```
set item1 to ":Scripts:Test script"

tell application "KaleidaGraph"
        LoadPlotScript "Scripts:Sample script"
        RunPlotScript
        LoadPlotScript item1
        RunPlotScript
end tell
```

Open - このイベントは指定ファイルを開きます。これは KaleidaGraph が認識できるすべてのファイルタイプを開くために使用できます。

Open alias 文字列

```
set item1 to "Macintosh HD:Applications:KaleidaGraph 4.5:Examples:Plots:Axial Strain.QPC"

tell application "KaleidaGraph"
    Open alias "Macintosh HD:Applications:KaleidaGraph 4.5:Examples:Data:Example.QDA"
    Open alias item1
    copy {"Macintosh HD:Applications:KaleidaGraph 4.5:Examples:Data:Housing Starts.QDA",
      "Macintosh HD:Applications:KaleidaGraph 4.5:Examples:Data:Pressure Data.QDA"} to
      List1
    repeat with i in List1
        Open alias i
    end repeat
end tell
```

OpenDatafile - このイベントはバイナリとテキストデータファイルを開くのに使用します。他のファイルタイプ（プロット、スタイル、マクロ）では、Open イベントを使用します。データファイルの属性は文字列の一部としてセットされ、KaleidaGraph マニュアルの 12.2.2 節で解説される数式スクリプトコマンド #DATAFILE の下に記載されています。

OpenDatafile 文字列

[Name エイリアス]

[Result: 文字列]

```
set item1 to "Macintosh HD:Applications:KaleidaGraph 4.5:Examples:Data:"

set item2 to "file = \"Macintosh HD:Applications:KaleidaGraph 4.5:Examples:Data:Text
  Example.TXT\"
delimiter = tab
read_titles = yes"

tell application "KaleidaGraph"
    SetRefDirectory item1
    OpenDatafile "file = \"Sunspot Data.QDA\""
    OpenDatafile item2
end tell
```

PlotPrint - このイベントは 一つ以上のプロットをプリントします。デフォルトで、最前面のプロットはプロットウィンドウから直接プリントされます。PageLayout オプションを使用して、レイアウトウィンドウから複数のプロットをプリントすることができます。

PlotPrint

[PageLayout ブール]

```
tell application "KaleidaGraph"
    activate
    PlotPrint
    SetSelection "10 20 0 3"
    Rebuild
    PlotPrint with PageLayout
end tell
```

Print - このイベントは指定ファイルをプリントします。これは、任意のデータファイル（バイナリやテキスト）または、プロットファイルをプリントできます。

Print alias 文字列

```
set item1 to "Macintosh HD:Applications:KaleidaGraph 4.5:Examples:Plots:Axial Strain.QPC"

tell application "KaleidaGraph"
	activate
	Print alias "Macintosh HD:Applications:KaleidaGraph 4.5:Examples:Data:Example.QDA"
	Print alias item1
	copy {"Macintosh HD:Applications:KaleidaGraph 4.5:Examples:Data:Housing Starts.QDA",
	   "Macintosh HD:Applications:KaleidaGraph 4.5:Examples:Data:Pressure Data.QDA"} to
	   List1
	repeat with i in List1
		Print alias i
	end repeat
end tell
```

Quit - このイベントは KaleidaGraph アプリケーションを終了します。デフォルトの動作は、何か変更があったときに保存を尋ねますが、オプション指定によって変更を自動的に保存したり、保存せずに終了することができます。

Quit yes/no/ask

```
tell application "KaleidaGraph"
	activate
	Quit no
end tell
```

Ready - このイベントは KaleidaGraph が起動しているかどうかチェックをします。

Ready

[Result: ブール]

```
set item1 to "Macintosh HD:Applications:KaleidaGraph 4.5:Examples:Data:"

tell application "KaleidaGraph"
	repeat while (Ready) is false
	end repeat
	OpenDatafile "file = \"Sunspot Data.QDA\""
end tell
```

Rebuild - このイベントはデータウィンドウの現選択内容に基づきプロットを再描画します。

Rebuild

```
tell application "KaleidaGraph"
	activate
	SetSelection "100 200 0 4"
	Rebuild
end tell
```

Run - このイベントは KaleidaGraph アプリケーションを起動させます。アプリケーションが起動していない場合は、tell コマンドがプログラムを起動するため、このイベントは技術的に必要ありません。

Run

```
tell application "KaleidaGraph"
	Run
	activate
end tell
```

RunPlotScript - このイベントは現在のプロットスクリプトを実行します。

RunPlotScript

```
tell application "KaleidaGraph"
	LoadPlotScript "Scripts:Sample script"
	RunPlotScript
end tell
```

RunTextScript - このイベントは文字列としてストアされているテキストスクリプトを実行します。このイベントはヘルプファイルまたは、KaleidaGraph マニュアルの 12.2.2 節に記載されている数式スクリプトコマンドのいずれかを実行することができます。もし #FORMULA コマンドを使用している場合、このコマンドに #FORMULA と #END 文を含める必要はありません。これは、RunTextScript イベントのデフォルト動作です。

RunTextScript 文字列

[Result: 画像]

```
set item1 to "
c4=mean([0:0, 1:3]);
c5=stderr([0:0, 1:3]);"

set test2 to "
#SCRIPT
plot_type=scatter
begin_group
x 0
y 4
end_group
#END"

tell application "KaleidaGraph"
	Open alias "Macintosh HD:Applications:KaleidaGraph 4.5:Examples:Data:Example.QDA"
	RunTextScript item1
	RunTextScript item2
end tell
```

SelectWindow - このイベントは名前（タイトルバー）または位置（前面から背面方向で、最前面ウィンドウがポジション 0）によって指定したウィンドウを選択します。

SelectWindow 文字列

[position 整数]

```
set item1 to "
c4=mean([0:0, 1:3]);
c5=stderr([0:0, 1:3]);"

set item2 to "Data 3"

tell application "KaleidaGraph"
	SelectWindow position 1
	RunTextScript item1
	SelectWindow "Data 1"
	RunTextScript item1
	SelectWindow item2
	RunTextScript item1
end tell
```

SendData - このイベントはデータを KaleidaGraph に送り、新規データウィンドウに表示します。データは、タブ区切りで文字列としてストアする必要があります。デフォルトでは、開始行にタイトルが含まれていると見なされます。

SendData 文字列

[Titles ブール]

```
set item1 to "1		1.1		1.5
2		2.2		2.5
3		3.3		3.5"

tell application "KaleidaGraph"
	if Ready is true
		SendData item1 without titles
	end if
end tell
```

SetRefDirectory - このイベントは参照ディレクトリを設定します。文字列は、目的のディレクトリの完全もしくは部分修飾パスです。最後の名前がファイルでない場合は、コロン （:） で終わる必要があります。ディレクトリを 1 つ繰り上げるには、パスは 2 つのコロン （::） で表します。デフォルトのディレクトリをアプリケーションのディレクトリに設定するには、パスをブランク （""） のままにします。

SetRefDirectory 文字列

```
set item1 to "Macintosh HD:Applications:KaleidaGraph 4.5:Examples:Data:"

tell application "KaleidaGraph"
	SetRefDirectory item1
	OpenDatafile "file = \"Housing Starts.QDA\""
end tell
```

SetSelection - このイベントは最前面のデータウィンドウで選択を行います。選択範囲は、空白類文字で区切られる数字を含む文字列で、その順番は [開始行、終了行、開始列、終了列] です。例外として、開始行に -1 を入力した場合は、ウィンドウ全体が選択されます。0 で始まる行と列のポジションのアドレス選択は、データウィンドウ（ポジション 0, 0）の左上隅からカウントされます。

SetSelection 文字列

```
tell application "KaleidaGraph"
	GetData "0 15 0 1"
	set string1 to result
	SelectWindow position 1
	SetSelection "0 15 1 2"
	AppendColumn string1
end tell
```

1.3 AppleScript の例題

このセクションのすべての例題は、同じ基本形式に従います。概要はスクリプトが何を行うのかを説明します。また、例題のスクリプトは、AppleScript エディタに直接入力したときに表示されるのと同じように記載しています。以下のスクリプトは、スクリプトの各ステップを記述した説明です。

注意： 以下の例題では Run イベントを使用しないで下さい。これは、KaleidaGraph がまだ起動していない場合、tell コマンドで起動するためです。

1.3.1 基本例題

基本例題 1

この例題のスクリプトは次の機能を実行します。

- KaleidaGraph を最前面にもってきます。
- 最前面のデータウィンドウにデータ選択を行い、文字列としてこれをストアします。
- 前から 2 番目のウィンドウを選択します。
- 最前面のデータウィンドウに文字列を追加します。

スクリプトは次のとおり、動作についての説明が続きます。

```
tell application "KaleidaGraph"
	activate
	GetData "-1"
	set string1 to result
	SelectWindow position 1
	AppendColumn string1
end tell
```

GetData コマンドは、最前面のデータウィンドウのセルすべてを選択し、タブ区切りデータのテキスト文字列としてストアします。このテキスト文字列は、**set** コマンドを使用して string1 と呼ばれる変数にストアされます。**SelectWindow** コマンドは、前から 2 番目のウィンドウをアクティブにします（position 1）。**AppendColumn** コマンドは、最前面のデータウィンドウに string1 からデータを追加します。

基本例題 2

この例題のスクリプトは次の機能を実行します。

- データファイルを開く。
- データセットで計算を実行します。
- 変更内容を保存した後、データウィンドウを閉じます。

スクリプトは次のとおり、動作についての説明が続きます。

```
tell application "KaleidaGraph"
      OpenDatafile "file = \"Macintosh HD:Applications:KaleidaGraph 4.5:Examples:Data:Housing
         Starts.QDA\""
      RunTextScript "
c4=mean([0:0, 1:3]);
c5=stderr([0:0, 1:3]);"
      Close with saving
end tell
```

OpenDatafile コマンドは、指定したデータファイルを開きます。**RunTextScript** コマンドは、データファイルの一部に平均と標準誤差を計算する 2 つの数式入力コマンドを実行します。**Close** コマンドは、データファイルを保存しデータウィンドウを閉じます。

基本例題 3

この例題のスクリプトは次の機能を実行します。

- KaleidaGraph を最前面にもってきます。
- 最前面のデータウィンドウで選択を行います。
- 2 つの数式入力コマンドを実行します。
- 行データの散布図を作成します。

スクリプトは次のとおり、動作についての説明が続きます。

```
set test1 to "
c0 = index()+1;
c1 = log(c0);"

set test2 to "
#SCRIPT
x axis limits maximum 80
x axis limits minimum 0
plot_type=scatter
begin_group
x 0
y 1
end_group
#END"

tell application "KaleidaGraph"
      activate
      SetSelection "0 77 0 0"
      RunTextScript test1
      RunTextScript test2
end tell
```

スクリプト開始時の 2 つの **set** コマンドは、test1 と test2 変数にテキスト文字列をストアします。文字列 test1 には、2 つの数式入力コマンドが含まれています。文字列 test2 には、プロットスクリプトを実行するための情報が含まれています。

activate コマンドは、KaleidaGraph をアクティブなアプリケーションにします。**SetSelection** コマンドは、最前面のデータウィンドウで列 0 の行 0 から行 77 までを選択します。最初の **RunTextScript** コマンドは、数式入力から 2 つのコマンドを実行することによって、データウィンドウで 2 つのデータ列を作成します。2 番目の **RunTextScript** コマンドは、このデータの散布図を生成します。

1.3.2 中級例題

中級例題 1

この例題のスクリプトは次の機能を実行します。

- 2 つの異なるウィンドウを選択し、これらを前面に持ってきます。
- それぞれのデータウィンドウにデータを追加します。

スクリプトは次のとおり、動作についての説明が続きます。

```
set item1 to "1.00        1.40        6.20
1.50        4.50        5.55
2.00        4.30        5.00
2.50        5.35        4.50
3.00        4.75        4.10"

set item2 to "Time        Test #1     Test #2
1.00        1.40        6.20
1.50        4.50        5.55
2.00        4.30        5.00
2.50        5.35        4.50
3.00        4.75        4.10"

tell application "KaleidaGraph"
      activate
      SelectWindow "Data 1"
      AppendColumn item1
      SelectWindow position 1
      AppendColumn item2 with Titles
end tell
```

スクリプト開始時の 2 つの **set** コマンドは、item1 と item2 変数にテキスト文字列をストアします。文字列 item1 には、タイトルのない 5 行のデータが含まれます。文字列 item2 には、6 行のデータが含まれます。

activate コマンドは、KaleidaGraph をアクティブなアプリケーションにします。**SelectWindow** コマンドは、**" データ 1"** のタイトルウィンドウを前面にもってきます。**AppendColumn** コマンドは、このデータウィンドウに文字列 item1 のデータを追加します。2 番目の **SelectWindow** コマンドは、前から 2 つめのウィンドウをアクティブにします。2 番目の **AppendColumn** コマンドは、最初の行にタイトルが含まれるのを除き、1 番目と同様に動作します。activate コマンドは、KaleidaGraph をアクティブなアプリケーションにします。

中級例題 2

この例題のスクリプトは次の機能を実行します。

- 保存されたデータファイルを開きます。
- 折れ線グラフを作成します。
- TIFF ファイルとしてプロットを保存します。

スクリプトは次のとおり、動作についての説明が続きます。

```
set path1 to "Macintosh HD:Applications:KaleidaGraph 4.5:Examples:Data:"

set item1 to "
#SCRIPT
y axis title \"Range\"
plot_type = Line
begin_group
x 0
y 1 to 3
end_group
#END"

set item2 to "
type = tiff
dpi 600
cmyk yes
file = \"Sample TIFF.tif\""

tell application "KaleidaGraph"
	SetRefDirectory path1
	OpenDatafile "file = \"Sunspot Data.QDA\""
	RunTextScript item1
	ExportPlot item2
end tell
```

スクリプト開始時の 3 つの **set** コマンドは、path1、item1、item2 変数にテキスト文字列をストアします。文字列 path1 には、Data フォルダのパスが含まれます。文字列 item1 には、プロットスクリプトを実行するための情報が含まれます。文字列 item2 には、TIFF ファイルで書き出すためのパラメータが含まれます。

SetRefDirectory コマンドは、path1 に含まれる文字列にカレントディレクトリを設定します。**Sunspot Data** ファイルは、**OpenDatafile** コマンドを使用して開きます。**RunTextScript** コマンドは、文字列 item1 で定義されたプロットスクリプトを実行します。**ExportPlot** コマンドは、PICT ファイルとして新しく作成されたプロットを保存します。このファイルは、元のデータファイルと同じディレクトリに保存されます。

中級例題 3

この例題のスクリプトは次の機能を実行します。

- 数式入力から 2 つのコマンドを使用してデータを生成します。
- 折れ線グラフを作成して、データに多項式カーブフィットを適用します。
- データウィンドウでカーブフィット値をストアし、残差を計算します。
- 得られたプロットをプリントします。

スクリプトは次のとおり、動作についての説明が続きます。

```
set data1 to "
macro(\"Pi Series\");
macro(\"sinc(5x)\");"

set data2 to "
name(\"Fit Values\", c3);
c3=poly(c0,c1);
name(\"Residuals\", c2);
c2=c3-c1;"

set plot1 to "#SCRIPT
plot_type line
add_fit 1 polynomial 4
begin_group
x 0
y 1
end_group
#END"

tell application "KaleidaGraph"
	RunTextScript data1
	RunTextScript plot1
	RunTextScript data2
	PlotPrint
end tell
```

スクリプト開始時の 3 つの **set** コマンドは、data1、data2 および plot1 変数にテキスト文字列をストアします。文字列 data1 には、**マクロ**メニューから 2 つのマクロを実行する 式が 2 つ含まれています。文字列 data2 には、多項式カーブフィットの残差と式を計算する式が含まれます。文字列 plot1 は、データの折れ線グラフを作成して 4 次多項式カーブフィットをあてはめます。

最初の **RunTextScript** コマンドは、2 つのデータ列を作成します。2 つめの **RunTextScript** コマンドは、プロットを作成し、カーブフィットをあてはめます。最後の **RunTextScript** コマンドは多項式カーブフィットの値と残差を計算します。得られたプロットは **PlotPrint** コマンドを使用してプリントされます。

1.3.3 上級例題

上級例題 1

この例題のスクリプトは次の機能を実行します。

- ユーザーにデータファイルを開くように促します。
- KaleidaGraph を最前面に移動します。
- データの散布図を作成して線形カーブフィットをあてはめ、プロットをプリントします。
- データウィンドウで選択を行います。
- プロットを更新し、プリントします。

スクリプトは次のとおり、動作についての説明が続きます。

```
set test1 to "
#FORMULA
c4=mean([0:0, 1:3]);
name(\"Average\", c4);
#END

#SCRIPT
y axis title \"Range\"
plot_type scatter
add_fit 3 linear
begin_group
x 0
y 1 to 2
y 4
end_group
#END"

tell application "Finder"
	activate
	open document file (choose file)
end tell

tell application "KaleidaGraph"
	activate
	RunTextScript test1
	PlotPrint
	SelectWindow position 1
	SetSelection "0 8 0 4"
	Rebuild
	PlotPrint
end tell
```

スクリプト開始時の **set** コマンドは、2 つの式とプロットスクリプトを実行するための情報を test1 変数にストアします。

このスクリプトの Finder 部分は、ユーザーがデータファイルを選択して開くことができるように Open ダイアログを表示します。**RunTextScript** コマンドは、平均を計算する文字列 test1 でコマンドを実行し、プロットの生成および線形回帰をあてはめます。得られたプロットは **PlotPrint** コマンドを使用してプリントされます。

SelectWindow コマンドは、データウィンドウを前面に持ってきます。**SetSelection** コマンドは、列 0 ～列 4、行 0 ～行 8 までをハイライト表示します。**Rebuild** コマンドは、データウィンドウで現在選択されているデータを使用して、プロットを更新してカーブフィットします。2 つめの **PlotPrint** コマンドは、更新されたプロットをプリントします。

上級例題 2

この例題のスクリプトは次の機能を実行します。

- 指定フォルダ内のすべてのファイルのファイルタイプを確定します。
- 検索した KaleidaGraph のバイナリデータファイルを開きます。
- 指定したファイル形式を使用して、すべてのテキストファイルを開きます。
- 開いたファイルの合計をダイアログで表示します。

スクリプトは次のとおり、動作についての説明が続きます。

```
set path1 to "Macintosh HD:Applications:KaleidaGraph 4.5:Examples:Data:"

set item1 to "
delimiter = tab
read_titles = yes
del_number = 1"

set x to 0

tell application "KaleidaGraph"
	list folder path1
	set List1 to result
	repeat with i in List1
		set item2 to path1 & i
		info for file item2
		set rec1 to result
		if rec1 contains {file type:"QDAT"} or rec1 contains {file type:"TEXT"} then
			if rec1 contains {file type:"QDAT"} then
				Open alias item2
				copy x + 1 to x
			else
				set item3 to (("file =" & "\"" & item2) & "\"") & item1
				OpenDatafile item3
				copy x + 1 to x
			end if
		end if
	end repeat
	display dialog (result as string) & " data windows were opened by the script."
end tell
```

スクリプト開始時の 3 つの **set** コマンドは、path1 と item1 変数に文字列をストアし、さらに x 変数に 0 の値をストアします。文字列 path1 には、**Data** フォルダのパスが含まれます。文字列 item1 には、KaleidaGraph にテキストファイルを読み込むためのパラメータが含まれます。

list folder コマンドは、path1 で指定したフォルダ内のすべての項目のリストをコンパイルします。
set コマンドは、List1 変数でこのリストをストアします。repeat コマンドは、フォルダ内の各ファイルでループ操作の命令ができるように、ループの設定を行います。i 変数には、現在評価しているファイルの名前がストアされます。

ループ内の最初の **set** コマンドは、path1 と i 文字列を単一文字列に連結します。この文字列は、item2 変数にストアされます。**info for** コマンドは、特定のファイルに関する情報のリストを取得します。ループ内の 2 つめの set コマンドを使用して、この情報は rec1 変数のレコードとしてストアされます。

if コマンドは、各ファイルのファイルタイプが QDAT または TEXT かどうかをテストします（ QDAT は KaleidaGraph のバイナリデータファイル、TEXT は一般的なテキストファイルのファイルタイプです)。ファイルタイプがこれらのいずれにもマッチしない場合は、ファイルはスキップされ、スクリプトはフォルダ内の次のファイルで作動します。ファイルタイプが QDAT の場合は、ファイルは **Open** コマンドを使用して開きます。ファイルタイプが TEXT の場合は、**set** コマンドはテキスト文字列を連結し、item3 変数の結果の文字列にストアします。このファイルは、**OpenDatafile** コマンドを使用して開かれます。

copy コマンドは、データファイルが開かれるたびに x 変数の値を増やします。すべてのファイルが処理された後、**display dialog** コマンドはスクリプトによって開かれたファイル数を示します。

上級例題 3

この例題のスクリプトは次の機能を実行します。

- 特定のディレクトリにある任意の KaleidaGraph のデータファイルを検索して変更日でソートします。
- 最新のデータファイルを開きます。
- 各データセットから折れ線グラフを作成します。
- データウィンドウと同じ名前でプロットを保存して TIFF 画像で書き出します。

スクリプトは次のとおり、動作についての説明が続きます。

```
set path1 to "Macintosh HD:Applications:KaleidaGraph 4.5:Examples:Data:"

set script1 to "
#SCRIPT
plot_type line
begin_group
x 0
y 1
end_group
#END"

tell application "Finder"
	set List1 to every document file of folder path1 whose name extension is in {"qda"}
	set List1 to (sort List1 by modification date)
	set theNames to {}
	repeat with oneFile in List1
		set end of theNames to name of oneFile
	end repeat
	repeat with i from 1 to 3
		set item1 to path1 & item i of theNames

		tell application "KaleidaGraph"
			activate
			Open alias item1
			GetWindowName position 0
			set name1 to result
			RunTextScript script1

			tell application "System Events"
				keystroke "s" using command down
				keystroke name1 & (ASCII character of 13)
				delay 1
			end tell

			set item2 to "
			type = tiff
			dpi = 600
			min_whitespace = yes
			file = \""
```

```
                ExportPlot item2 & path1 & name1 & ".tif\""
                Close
                Close
            end tell
        end repeat
    end tell
```

最初の **set** コマンドはフォルダへのパスを定義します。2 つめの **set** コマンドは、プロット作成するために使用するスクリプトをストアします。

スクリプトは特定のフォルダ内の KaleidaGraph データファイルの場所を検索することから開始します。これらのファイルは変更日でソートされます。ソートされた順で変数にストアされます。

repeat コマンドは 3 つの最新のデータファイル上で動作するループを設定します。**activate** コマンドによって KaleidaGraph は前面に移動し、データファイルを開きます。後で使用するためにデータウィンドウの名前がストアされ、**RunTextScript** コマンドはプロットを作成します。

keystroke コマンドは **Save Graph** コマンドの選択ファイル名の置き換えプロットの保存を引き起こします。 それから、**ExportPlot** コマンドによって TIFF 画像でプロットが書き出されます。ファイルは元のデータファイルと同じディレクトリに保存されます。

次に **Close** コマンドはプロットとデータウィンドウを閉じます。このスクリプトは、残りの 2 つのデータファイルでも繰り返されます。

1.4 Tips

delay コマンドは、続行する前にバックグランドで何らかの競合で待機が必要になった場合、スクリプトに一時停止を挿入するために使用できます。**delay 3** は、3 秒の一時停止を引き起こします。

次のようなルーチンを使用して KaleidaGraph でメニューアイテムを選択できます。

```
tell application "KaleidaGraph" to activate
tell application "System Events"
    tell application process "KaleidaGraph"
        click menu item "開く..." of menu "ファイル" of ¬
            menu bar item "ファイル" of menu bar 1
    end tell
end tell
```

keystroke コマンドを使用してキーボードかキーを発行できます。このコマンドの構文も、**Shift** および **Command** のような修飾キーを使用しています。

1.5 トラブルシューティング

スクリプトを試行しているときに問題が発生したら、AppleScript エディタでイベントログを使用して、イベント、返された値、結果を表示して問題解決に役立ててください。

次のいずれかが起こるとエラーが生じます。

- ウィンドウが存在せずに位置番号を指定する。
- 誤った名前でウィンドウを指定する。
- データウィンドウを開かずに **GetData** や **SetSelection** イベントを使おうとしている。
- プロットウィンドウを開かずに **ExportPlot**、**PlotPrint**、**Rebuild** イベントを使おうとしている。

RunTextScript コマンド内で数式スクリプトコマンドを使用する際、# 文がすべての大文字に入力されていなかったり、数式スクリプトコマンド内のいくつかの行がインデントされているとエラーが出ます。数式スクリプトコマンドはスクリプトの **tell** コマンドで並べる必要があります。

RunTextScript イベントの一部として #COPY コマンドを首尾よく使用するには、クリップボードにデータをコピーする前にアクティブなアプリケーションにしておく必要があります。最も簡単な方法は RunTextScript イベントの前に **activate** コマンドを置くことです。そうしないと、クリップボードの情報は、#COPY が実行されたときに変更されません。これは、最前面のアプリケーションからのみクリップボードの内容を置き換えることが可能だからです。

1.6 Apple Event

KaleidaGraph は、以下の Apple Event をサポートしています。各イベントの説明は表の後にあります。

イベント （クラス： QKPT）				
ID	ダイレクト	オプション	リターン	意味
call	なし	なし	なし	保存せずに全てのウィンドウを閉じます。
cfns	なし	なし	なし	保存せずに最前面のウィンドウを閉じます。
clos	テキスト	なし	なし	オプションでその内容を保存し、指定したウィンドウを閉じます。
epic	テキスト	ファイル	PICT （opt）	最前面のプロットを PICT のファイルまたはクリップボードに書き出します。
gsel	テキスト	なし	テキスト	最前面のデータウィンドウから指定したデータ選択の値を返します。
kdoc	テキスト	ファイル	なし	指定したデータファイルを読み込みます。
kgqt	なし	なし	なし	KaleidaGraph をできるだけ早く終了します。
ldac	テキスト	なし	なし	最前面のデータウィンドウに、一つ以上のデータ列を追加します。
ldap	テキスト	なし	なし	最前面のデータウィンドウに、一つ以上のデータ行を追加します。
ldat	テキスト	なし	なし	開始行にタイトルを持つ、タブ区切りデータを読み込みます。
ldnt	テキスト	なし	なし	開始行にデータを持つ、タブ区切りデータを読み込みます。
lscp	テキスト	ファイル	なし	指定したプロットスクリプトを読み込みます。
oapp	なし	なし	なし	アプリケーションを実行します。
odoc	テキスト	ファイル	なし	指定した書類を開きます。
pdoc	テキスト	ファイル	なし	指定した書類をプリントします。
prpl	テキスト	なし	なし	一つ以上のプロットをプリントします。
quit	テキスト	なし	なし	アプリケーションを終了します。
redy	なし	なし	なし	KaleidaGraph が動作しているかどうかを確認します。
refd	テキスト または typeFSS	なし	なし	参照ディレクトリを設定します。
rplt	なし	なし	なし	データウィンドウで現在の選択を使用し、アクティブなプロットを再プロットします。
rscp	なし	なし	なし	現在のプロットスクリプトを実行します。
selw	テキスト	なし	なし	最前面のデータウィンドウで選択を設定します。
ssel	テキスト	なし	なし	最前面のデータウィンドウで選択を設定します。
tscp	テキスト	なし	PICT （opt）	テキストスクリプトを実行します。
wlis	テキスト	なし	テキスト	リストの 一つ以上のウィンドウの名前を取得します。位置が指定されないときは、全てのウィンドウリストが返されます。

注意： アプリケーション署名は **QKPT** です。

call

このイベントは内容を保存せずに、全てのウィンドウを閉じます。

cfns

このイベントは内容を保存せずに、最前面のウィンドウを閉じます。

clos

このイベントはオプションで内容を保存し、指定したウィンドウを閉じます。

epic

このイベントはいくつかの形式があり、両方のパラメータはオプションです。このイベントの構文は、#PICT/OPT 数式スクリプトコマンドと同じです。このコマンドは、ユーザーズガイドの 12.2.2 節で解説しています。

パラメータが指定されない場合は､最前面のプロットは PICT で返されます｡ダイレクトオブジェクトはテキストで､倍率、PostScript PICT、ファイルパラメータを設定します。オプションの FILE パラメータは typeFSS や typeAlias で、PICT ファイルに保存することができます。

注意： ファイルが指定されない場合のみ、PICT が返されます。

gsel

最前面のデータウィンドウに選択を設定し、それに含まれるデータを返します。選択範囲は、空白で区切られた数字としてダイレクトオブジェクトに含まれます。このイベントの構文は、#COPY 数式スクリプトコマンドと同じです。このコマンドは、ユーザーズガイド 12.2.2 節で解説しています。

kdoc

ダイレクトオブジェクトかファイルオブジェクトで記述されたデータファイルをロードします。データファイルがテキストの場合は､ファイルを読み込むためにダイレクトオブジェクトの情報（あるいは現在のテキストファイルの定義）を使用します。オプションのファイルオブジェクトは、typeFSS あるいは typeAlias です。このイベントの構文は、#DATAFILE 数式スクリプトコマンドと同じです。このコマンドは、ユーザーズガイド 12.2.2 節 で解説しています。

kgqt

このイベントはできるだけ早く KaleidaGraph を終了させます。

ldac

最前面のデータウィンドウに、ダイレクトオブジェクトに含まれる 一つ以上のデータ列を追加します。データは、先頭行にデータを伴うタブ区切りとみなされます。

ldap

最前面のデータウィンドウに、ダイレクトオブジェクトに含まれる一つ以上のデータ行を追加します。データは、先頭行にデータを伴うタブ区切りとみなされます。

ldat

ダイレクトオブジェクトに含まれるデータを新しいデータウィンドウにロードします。データは、先頭行にタイトルを伴うタブ区切りとみなされます。

ldnt

ダイレクトオブジェクトに含まれるデータを新しいデータウィンドウにロードします。データは、先頭行にデータを伴うタブ区切りとみなされます。

lscp

ダイレクトオブジェクトかファイルオブジェクトで指定したプロットスクリプトをロードします。このファイルは、開いた最後のプロットスクリプトと同じフォルダに置くか、フルパスで指定しなければなりません。オプションのファイルオブジェクトは、typeFSS あるいは typeAlias です。

redy

KaleidaGraph が実行しているかどうか確認します。

refd

参照ディレクトリをダイレクトオブジェクトで指定したディレクトリに設定します。ダイレクトオブジェクトはテキスト、typeFSS あるいは typeAlias のタイプでもよく。タイプがテキストの場合、オブジェクトに新しいディレクトリのフルまたは部分パスのいずれかを記述するシンプルなテキスト文字列が含まれます。

参照ディレクトリは、スクリプトファイルを記述するものを除き、シンプルなファイル名や部分パスのベースディレクトリとして使用されます。スクリプトファイルは独自にベースディレクトリを持ちます。このイベントにオプションのパラメータはありません。

rplt

データウィンドウの現在の選択を使用し、アクティブなプロットを再プロットします。

rscp

現在のプロットスクリプトを実行します。

selw

名前あるいは位置によってウィンドウを選択します。

ssel

最前面のデータウィンドウに選択を設定します。選択範囲は、空白で区切られた数としてダイレクトオブジェクトに含まれます。このイベントの構文は、#SELECTION 数式スクリプトコマンドと同じです。このコマンドは、ユーザーズガイド 12.2.2 節で解説しています。

tscp

ダイレクトオブジェクトに含まれるテキストスクリプトを実行します。コマンドは、ユーザーズガイド 12.2.2 節に記載されているイベントによってサポートされます。全てのコマンドは、数式入力ウィンドウの添付ノートに数式スクリプトを書くために使用することができます。

wlis

リストにある 一つ以上のウィンドウ名を取得します。位置が指定されない場合は、全てのウィンドウリストが返されます。