Using Origin from LabVIEW

Table of Contents

1 LabVIEW VI

The Origin installation provides a collection of custom building-block Virtual Instruments (VIs) that enable the user to create their own VIs to communicate with Origin. These custom VIs can be used for operations such as opening and closing communication with Origin, exchanging data back and forth between Origin and LabVIEW, and sending commands to Origin.

In order to access these Origin sub-VIs in the LabVIEW Functions Palette, select **Tools: Copy Origin Sub-VI to LabVIEW vi.lib\addons\Origin...** to automatically copy these files and optionally delete any older versions of these files.

Alternatively or for Origin 8.0 and earlier, you may copy the LabVIEW Libraries from *<Origin Program Folder>\Samples\COM Server and Client\LabVIEW\Palette* to the **vi.lib\addons\Origin** subfolder under the LabVIEW installation folder. Specifically, please copy the following files:

- **OriginApp_LV7.llb** - Basic VIs that handles the Origin OPJ files, worksheet and columns
- **OriginAppClassics_LV7.llb** - Older VIs exsited before Origin 8 (deprecated)
- **OriginWave_LV7.llb** - VIs that handles Origin matrix objects
- **OriginMatrix_LV7.llb** - VIs that handles LabVIEW Waveform data

Note:

- The custom VIs are designed to work in LabVIEW versions 7.0 or later. They do not work properly in LabVIEW version 5.0 or 6.1.
- Once sub-VI, OA_ConnectToOrigin, has been modified. If you have VIs that have made use of this sub-VI, you will need to replace it with the modified version.

Topics covered in this document:

2 Custom VIs for LabVIEW

This section provides details on each of the custom LabVIEW VIs provided with Origin.

Topics covered in this section:

2.1 OriginApp Sub-VIs

2.1.1 OriginApp Sub-VIs

These are the basic VIs that handles the Origin OPJ files and access to Origin worksheet and columns.

OriginApp Sub-VIs	Description
OA_AddOriginPath	Add various Origin path before your own file path, like UFF, or installed path.
OA_AddWorksheet	Add a new worksheet in an existed workbook using the specified template.
OA_Col-Get-dt	Get sampling interval and offset from a column.
OA_Col-GetData(Numeric)	Get a numeric array from a Column.
OA_Col-GetData(string)	Get text array from a Column.
OA_Col-SetData	Put a numeric array into a Column.
OA_Col-SetEvenSampling	Set sample rate of a Y column, so you don't need a separate data array of x axis.
OA_Col-Setting	Set Column data types and labels.
OA_ConnectToOrigin	Connect to Origin project and show the main window of Origin. Using ApplicationSI will not open new instance.
OA_CreateOGW	Create a worksheet for saving data to an Origin OGW file.

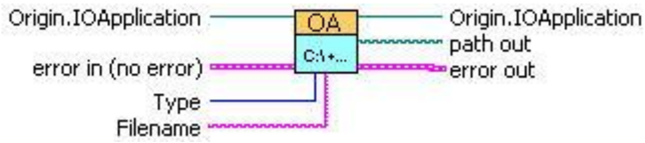
OA_Exit	Exit Origin.
OA_FindAddWorksheet	Find a Worksheet in the current project from a book sheet range string.
OA_FindWorkBook	Find a Workbook in the current project.
OA_FindWorksheet	Get a Worksheet object from a book sheet range string.
OA_GetColumn	From a Worksheet object to get a Column object by index or by name.
OA_IsVisible	Check if the Origin Application is shown or hidden.
OA_Load	Load an opj file or ogw file, or to start a new empty project if no file is specified.
OA_NewEmptyGraph	Create an empty graph window.
OA_NewWorksheet	Create a new worksheet in a new created workbook using the specified template.
OA_OpenNewOrigin	Start a new instance of Origin.
OA_PE_mkdir	Call the pe_mkdir X-Function to make a sub folder in the current folder in Origin.
OA_PlotWksCols	Plot a range of cells from a Worksheet to a GraphLayer.
OA_RunBatchProcess	Perform batch processing with opened analysis template.
OA_Save	Save an .opj file to disk.
OA_Wks-Get-2Cols	Bundle an x column and a y column of a worksheet to a cluster.
OA_Wks-GetCell(numeric)	Get numeric data from a worksheet cell.
OA_Wks-GetCell(String)	Get string data from a worksheet cell.
OA_Wks-GetRange	Get data from a worksheet.
OA_Wks-PutData	Send a 2D array to an Origin worksheet, starting from the c1 column.
OA_Wks-SetCell	Put data into a worksheet cell.

2.1.2 OA_AddOriginPath

2.1.2.1.1 Description

Add various Origin path before your own file path, like UFF, or installed path.

2.1.2.1.2 Connector Pane



2.1.2.1.3 Controls and Indicators

	Origin.IOApplication is a reference to Origin.ApplicationSI.

I32	**Type** is the path type before the **FileName**. It must equal one of the following values:
	`APPPATH_GROUP = 3  //The path of the Group folder`
	`APPPATH_PROGRAM = 4 //The path of the Origin executable`
	`APPPATH_USER = 2 //The path of the User Files Folder`
	The default **type** is **APPPATH_USER**.

abc	**FileName** is a string of your filename or file path
	Error In contains error information that occur before this VI or function runs.
	Origin.IOApplication is a reference to Origin.ApplicationSI.
	Path Out is a full path of the file.
	Error Out contains error information.

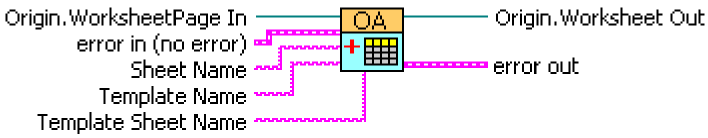
Note: This VI is available since Origin 8 SR2.

2.1.3 OA_AddWorksheet

2.1.3.1.1 Description

Add a new worksheet in an existed workbook using the specified template.

2.1.3.1.2 Connector Pane



2.1.3.1.3 Controls and Indicators

	Origin.WorksheetPage In is a reference to Origin.WorksheetPage.
	Error In contains error information that occur before this VI or function runs.
abc	**Sheet Name** is the name of the new sheet to be added.
abc	**Template Name** is the name of the template to create the new sheet from.
abc	**Template Sheet Name** is the name of the sheet in the template to create the new sheet from.
	Origin.WorkSheet output Worksheet refnum.
	Error out contains error information.

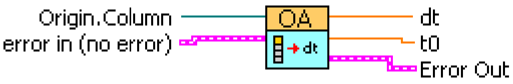
Note: This VI is available since Origin 8 SR5.

2.1.4 OA_Col-Get-dt

2.1.4.1.1 Description

Get sampling interval and offset from a column.

2.1.4.1.2 Connector Pane



2.1.4.1.3 Controls and Indicators

	Origin.Column is a reference to Origin.Column.
	error in(no error) contains error information that occur before this VI or function runs.
DBL	**dt** is the sampling interval of the column, or X increment between successive points.
DBL	**t0** is the offset in X.
	Error Out contains error information.

Note: This VI is available since Origin 8 SR4.

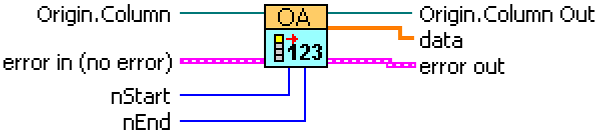
2.1.5 OA_Col-GetData(Numeric)

2.1.5.1.1 Description

Get a numeric array from a Column.

Minimum Version Required: Origin 8.5 SR0

2.1.5.1.2 Connector Pane



2.1.5.1.3 Controls and Indicators

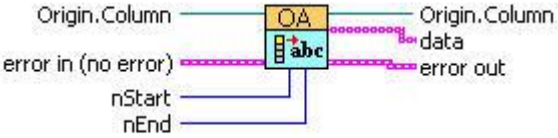
	Origin.Column is a reference to Origin.Column.
I32	**nStart** is the starting index to set array, default is 0.
I32	**nEnd** is the ending index to set array, default is -1.
	error in(no error) contains error information that occur before this VI or function runs.
	Origin.Column is a reference to Origin.Column.
DBL	**Data** is a 1D array of numeric data
	Error Out contains error information.

2.1.6 OA_Col-GetData(string)

2.1.6.1.1 Description

Get text array from a Column.

2.1.6.1.2 Connector Pane



2.1.6.1.3 Controls and Indicators

	Origin.Column is a reference to Origin.Column.
I32	**nStart** is the starting index to set array, default is 0.
I32	**nEnd** is the ending index to set array, default is -1.
	Error In contains error information that occur before this VI or function runs.
	Origin.Column is a reference to Origin.Column.
abc	**Data** is a 1D array of string.
	Error Out contains error information.

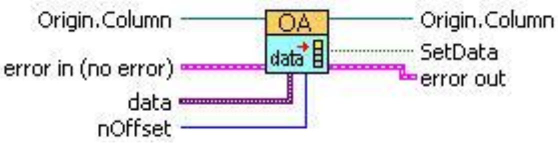
Note: This VI is available since Origin 8 SR2.

2.1.7 OA_Col-SetData

2.1.7.1.1 Description

Put a numeric array into a Column.

2.1.7.1.2 Connector Pane



2.1.7.1.3 Controls and Indicators

	Origin.Column is a reference to Origin.Column.
	Data is a variant, accepts 1D array of data (numeric or string).
I32	**nOffset** is the starting index to set array, default is 0, and can be -1 to append data to the end of the column
	error in(no error) contains error information that occur before this VI or function runs.
	Origin.Column is a reference to Origin.Column.
TF	**SetData** is **TRUE** if the data was set successful or **FALSE** to indicate that the data was set unsuccessful.
	Error Out contains error information.

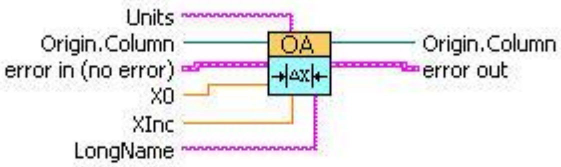
Note: This VI is available since Origin 8 SR2.

2.1.8 OA_Col-SetEvenSampling

2.1.8.1.1 Description

Set sample rate of a Y column, so you don't need a separate data array of x axis.

2.1.8.1.2 Connector Pane



2.1.8.1.3 Controls and Indicators

	Origin.Column is a reference to Origin.Column.

DBL	**X0** is the start point of the x axis.
DBL	**XInc** is the Sampling Interval, or X increment between successive points.
abc	**LongName** is a string of LongName set to the x axis.
abc	**Units** is a string of Unit set to the x axis.
	Error In contains error information that occur before this VI or function runs.
	Origin.Column.Out is a reference to Origin.Column.
	Error Out contains error information.

Note: This VI is available since Origin 8 SR2.

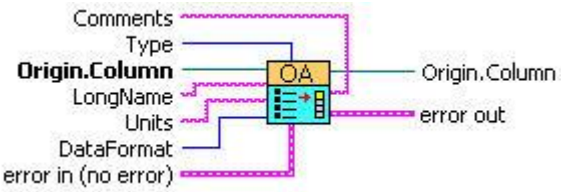
2.1.9 OA_Col-Setting

2.1.9.1.1 Description

Set Column data types and labels.

Minimum Version Required: Origin 8.5 SR0

2.1.9.1.2 Connector Pane



2.1.9.1.3 Controls and Indicators

	Origin.Column is a reference to Origin.Column.

I32	**Type** is the designation of the column. It must be one of the following values:
	COLTYPE_NO_CHANGE = -1
	COLTYPE_Y = 0
	COLTYPE_NONE = 1
	COLTYPE_ERROR = 2
	COLTYPE_X = 3
	COLTYPE_LABEL = 4
	COLTYPE_Z = 5

	COLTYPE_X_ERROR = 6
	COLTYPE_GROUP = 7
	COLTYPE_SUBJECT = 8
	The default value is **COLTYPE_NO_CHANGE**.

I32	**DataFormat** is the data type of the column. It must be one of the following values:
	DF_NO_CHANGE =-1
	DF_DOUBLE =0
	DF_TEXT =1
	DF_TIME = 2
	DF_DATE = 3
	DF_TEXT_NUMERIC = 9
	DF_FLOAT = 32
	DF_SHORT = 33
	DF_LONG =34
	DF_CHAR =35
	DF_BYTE = 38
	DF_USHORT = 39
	DF_ULONG =40
	DF_COMPLEX = 41
	The default value is **DF_NO_CHANGE**.

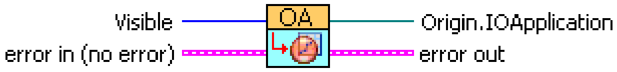
abc	**LongName** is a string of LongName set to the Column.
abc	**Units** is a string of Unit set to the Column.
abc	**Comments** is a string of comment set to the Column.
	Error In contains error information that occur before this VI or function runs.
	Origin.Column is a reference to Origin.Column.
	Error Out contains error information.

2.1.10 OA_ConnectToOrigin

2.1.10.1.1 Description

Connect to Origin project and show the main window of Origin. Using ApplicationSI will not open new instance.

2.1.10.1.2 Connector Pane



2.1.10.1.3 Controls and Indicators

I32	**Visible** is a property which allows you to change the visible state of the Origin application window. The different states include hidden, shown, minimized, maximized, and bring to front. It must be one of the following value:
	`MAINWND_HIDE = 0`
	`MAINWND_SHOW = 1`
	`MAINWND_SHOWMINIMIZED = 2`
	`MAINWND_SHOWMAXIMIZED =3`
	`MAINWND_SHOW_BRING_TO_FRONT =100`
	The default value is **MAINWND_SHOW**.

	error in(no error) contains error information that occur before this VI or function runs.
	Origin.IOApplication is a reference to Origin.ApplicationSI.
	Error Out contains error information.

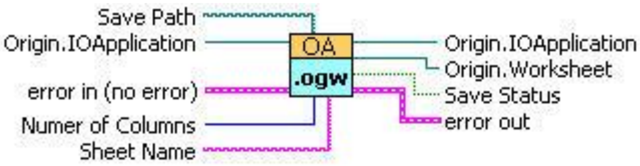
Note: This VI is available since Origin 8 SR2, but the connectors were modified in SR4, so you will have to replace it from the SR4 palette.

2.1.11 OA_CreateOGW

2.1.11.1.1 Description

Create a worksheet for saving data to an Origin OGW file.

2.1.11.1.2 Connector Pane



2.1.11.1.3 Controls and Indicators

	Origin.IOApplication
	Path accepts a full path of the ogw file that you are saving.
I32	**Number of Columns** controls the number of columns in the worksheet while creating the new ogw file.
abc	**Sheet Name** specify the name of the sheet of the new ogw file.
	Error In contains error information that occur before this VI or function runs.
	Origin.IOApplication.Out is a reference to Origin.ApplicationSI.
	Origin.Worksheet is a reference to Origin.Worksheet.
TF	**Save Status** is **TRUE**, file is saved successfully, otherwise, saved failed.
	Error Out contains error information.

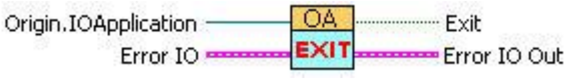
Note: This VI is available since Origin 8 SR2.

2.1.12 OA_Exit

2.1.12.1.1 Description

Exit Origin.

2.1.12.1.2 Connector Pane



2.1.12.1.3 Controls and Indicators

	Origin.IOApplication is a reference to Origin.ApplicationSI.
	Error In contains error information that occur before this VI or function runs.
TF	**Exit** is TRUE, exit Origin successfully, otherwise, failed.
	Error Out contains error information.

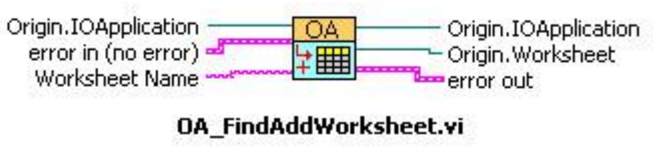
Note: This VI is available since Origin 8 SR2.

2.1.13 OA_FindAddWorksheet

2.1.13.1.1 Description

Find a Worksheet in the current project from a book sheet range string, like [book2]sheet2. If not find, add a new one. If the current project contain book2, then add a sheet named sheet2 in book2. If not, create a new worksheet in a new created workbook using the specified book name and sheet name.

2.1.13.1.2 Connector Pane



2.1.13.1.3 Controls and Indicators

	Origin.IOApplication
	Error In contains error information that occur before this VI or function runs.
abc	**Worksheet Name** is the name of the worksheet to be found or added. It is a range string eg. • **[Book1]Sheet2**: find or add Sheet2 in Book1 • **Book2** find the active sheet from Book2 or add Book2 • **Sheet2!** find Sheet2 from active book or add Sheet2 to active book You can use an empty string to get the active sheet from the active book, but if no active book or active book is not a workbook, then the output refnum for Origin.Worksheet will be invalid.
	Origin.IOApplication is a reference to Origin.ApplicationSI.
	Origin.Worksheet is a reference to Origin.Worksheet
	Error Out contains error information.

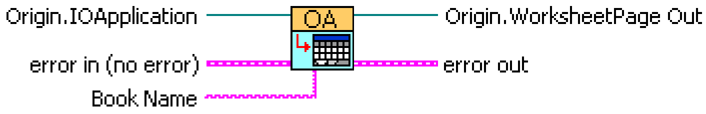
Note: This VI is available since Origin 9.

2.1.14 OA_FindWorkBook

2.1.14.1.1 Description

Find a Workbook in the current project.

2.1.14.1.2 Connector Pane



2.1.14.1.3 Controls and Indicators

	Origin.IOApplication
abc	**Book Name** is a name of a book that need to be found.
	Error In contains error information that occur before this VI or function runs.
	Origin.WorksheetPage Out is a reference to Origin.WorksheetPage.
	Error Out contains error information.

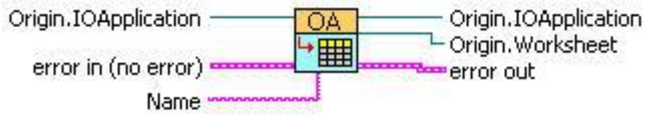
Note: This VI is available since Origin 8 SR5.

2.1.15 OA_FindWorksheet

2.1.15.1.1 Description

Get a Worksheet object from a book sheet range string, such as [Book1]Sheet2.

2.1.15.1.2 Connector Pane



2.1.15.1.3 Controls and Indicators

	Origin.IOApplication
abc	**Name** is a book sheet range string, such as "[Book1]Sheet1". You can use an empty string to get the active sheet from the active book, but if no active book or active book is not a workbook, then the output refnum for Origin.Worksheet will be invalid. The following variations are also supported: **Book1** find the active sheet from book1 **Sheet2!** find Sheet2 from the active book
	Error In contains error information that occur before this VI or function runs.
	Origin.IOApplication.Out is a reference to Origin.ApplicationSI.
	Origin.Worksheet is a reference to Origin.Worksheet
	Error Out contains error information.

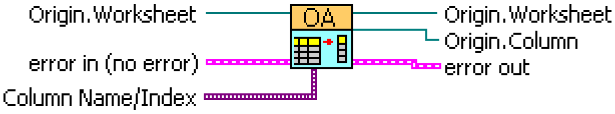
Note: This VI is available since Origin 8 SR2.

2.1.16 OA_GetColumn

2.1.16.1.1 Description

From a Worksheet object to get a Column object by index or by name.

2.1.16.1.2 Connector Pane



2.1.16.1.3 Controls and Indicators

	Origin.Worksheet is a reference to Origin.Worksheet.
	Column Name/Index is a variant, accepts either a string of column name or a number of column index
	Error In contains error information that occur before this VI or function runs.
	Origin.Worksheet is a reference to Origin.Worksheet.
	Origin.Column is a reference to Origin.Column.
	Error Out contains error information.

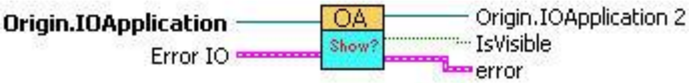
Note: This VI is available since Origin 8 SR2.

2.1.17 OA_IsVisible

2.1.17.1.1 Description

Check if the Origin Application is shown or hidden.

2.1.17.1.2 Connector Pane



2.1.17.1.3 Controls and Indicators

	Origin.IOApplication is a reference to Origin.ApplicationSI.

	Error In contains error information that occur before this VI or function runs.
	Origin.IOApplication.Out
TF	**IsVisible** is TRUE, Origin Application is shown, otherwise, is hidden.
	Error Out contains error information.

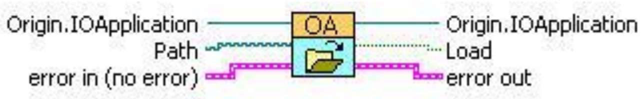
Note: This VI is available since Origin 8 SR3.

2.1.18 OA_Load

2.1.18.1.1 Description

Load an opj file or ogw file, or to start a new empty project if no file is specified.

2.1.18.1.2 Connector Pane



2.1.18.1.3 Controls and Indicators

	Origin.IOApplication is a reference to Origin.ApplicationSI.
	Path accepts a path containing the complete path to the opj or ogw file that you are loading, when not specified, an empty project is loaded
	Error In contains error information that occur before this VI or function runs.
	Origin.IOApplication.Out is a reference to Origin.ApplicationSI.
TF	**Load** is TRUE if file is loaded successfully, otherwise, load failed.
	Error Out contains error information.

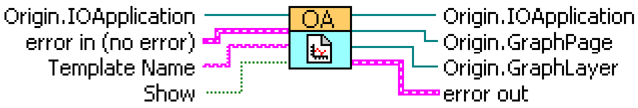
Note: This VI is available since Origin 8 SR2.

2.1.19 OA NewEmptyGraph

2.1.19.1.1 Description

Create an empty graph window.

2.1.19.1.2 Connector Pane



2.1.19.1.3 Controls and Indicators

	Origin.IOApplication is a reference to Origin.ApplicationSI.
	Error In contains error information that occur before this VI or function runs.
abc	**Template Name** is the name of the graph template. Note that tge .otp extention is not needed. The default template is **origin**.
TF	**Show** needs to be set to **TRUE** if you want to the new created Worksheet is shown or **FALSE** to be hidden.
	Origin.IOApplication is a reference to Origin.ApplicationSI.
	Origin.GraphPage output GraphPage refnum.
	Origin.GraphLayer output GraphLayer refnum.
	Error out is a reference to Origin.Column.

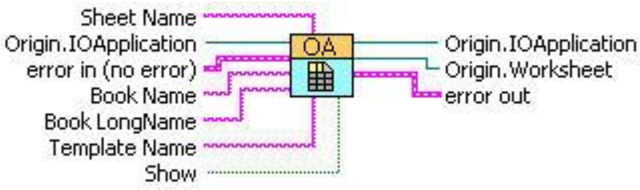
Note: This VI is available since Origin 8 SR3.

2.1.20 OA_NewWorksheet

2.1.20.1.1 Description

Create a new worksheet in a new created workbook using the specified template.

2.1.20.1.2 Connector Pane



2.1.20.1.3 Controls and Indicators

abc	**Sheet Name** specify the desired sheet name of the new workbook.
	Origin.IOApplication is a reference to Origin.ApplicationSI
	Error In contains error information that occur before this VI or function runs.
abc	**Book Name** specify the desired short name of the new workbook
abc	**Book LongName** specify the long name of the new workbook
abc	**Template Name** specify the template to create the book. Template can be full path file name, or just a name and Origin will search User Files, Group Files and then Program folder.

TF	**Show** needs to be set to **TRUE** if you want to the new created Worksheet is shown or **FALSE** to be hidden.
	Origin.IOApplication is a reference to Origin.ApplicationSI
	Origin.WorkSheet output Worksheet refnum.
	Error out contains error information.

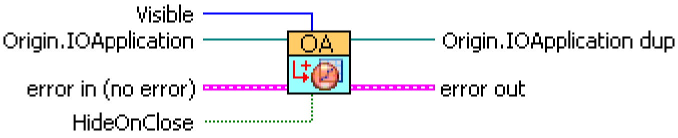
Note: This VI is available since Origin 8 SR3.

2.1.21 OA_OpenNewOrigin

2.1.21.1.1 Description

Start a new instance of Origin.

2.1.21.1.2 Connector Pane



2.1.21.1.3 Controls and Indicators

I32	**Visible** is a property which allows you to change the visible state of the Origin application window. The different states include hidden, shown, minimized, maximized, and bring to front. It must be one of the following value:
	MAINWND_HIDE = 0
	MAINWND_SHOW = 1
	MAINWND_SHOWMINIMIZED = 2
	MAINWND_SHOWMAXIMIZED =3
	MAINWND_SHOW_BRING_TO_FRONT =100
	The default value is **MAINWND_HIDE**.

	Origin.IOApplication is a reference to Origin.ApplicationSI.
	Error In contains error information that occur before this VI or function runs.
TF	**HideOnClose** need to set to **TRUE** (default is **FALSE**) if you want to make Origin hidden when you click the **Close** button to close Origin.
	Origin.IOApplication Dup a reference to Origin.ApplicationSI.
	Error Out contains error information.

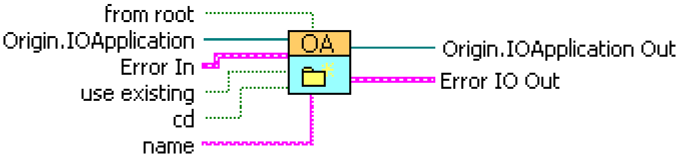
Note: This VI is available since Origin 8 SR2.

2.1.22 OA_PE_mkdir

2.1.22.1.1 Description

Call the pe_mkdir X-Function to make a sub folder in the current folder in Origin.

2.1.22.1.2 Connector Pane



2.1.22.1.3 Controls and Indicators

TF	**from root** need to be set to **TRUE** if you want to create the new folder from root. **FALSE** means to create the folder from the current folder.
	Origin.IOApplication is a reference to Origin.ApplicationSI.
	Error In contains error information that occur before this VI or function runs.
TF	**use existing** means use the existed folder that **name** indicates if it is set to **TRUE**. **FALSE** means to create a new folder using enumerated name.
TF	**cd** need to set **TRUE** if you want to switch to named folder or **FALSE** if not to change active folder.
abc	**name** specifies the new folder name.
	Origin.IOApplication.Out is a reference to Origin.ApplicationSI.
	Error IO Out contains error information.

Note: This VI is available since Origin 8 SR3.

2.1.23 OA_PlotWksCols

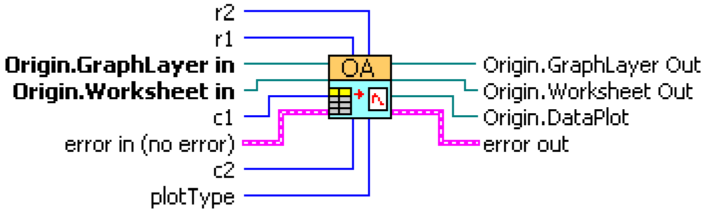
2.1.23.1.1 Description

Plot a range of cells from a Worksheet to a GraphLayer. Only XY type of plots are supported with this sub-VI.

You need to assign the proper column designations (Column.Type) in the worksheet before adding them to a graph layer. Origin make the plots in same way as you would when you select these columns to plot them. See LabVIEW:OA_Col-Setting on how to setup column designations.

Minimum Version Required: Origin 8.5 SR0

2.1.23.1.2 Connector Pane



2.1.23.1.3 Controls and Indicators

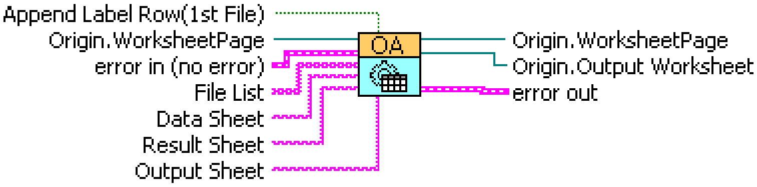
	Origin.GraphLayer in is a reference to Origin.GraphLayer.
	Origin.Worksheet in is a reference to Origin.Worksheet.
	error in contains error information that occur before this VI or function runs.
I32	**r1** is the first row of the range to be plotted. The default value is **0**.
I32	**r2** is the last row of the range to be plotted. The default value is **-1**, which indicates the end of data.
I32	**c1** is the first column of the range to be plotted. The default value is **0**.
I32	**c2** is the last column of the range to be plotted.The default value is **-1**, which indicates the end of data.
I32	**Plot Type** is one of the XY plot type that Origin supports, such as 200 = line 201 = scatter 202 = line + symbol Default value is **200**.
	Origin.GraphLayer Out is the refnum of the GraphLayer.
	Origin.Worksheet Out is the refnum of the Worksheet.
	error out contains error information.
	Origin.DataPlot is the refnum of Dataplot. It is a invalid refnum if the plot failed to be added.

2.1.24 OA_RunBatchProcess

2.1.24.1.1 Description

Perform batch processing with opened analysis template. Please note that this VI has the similar ability as the **Batch Processing** tool in Origin if the **Repeatedly Import into Active Analysis Template Window** mode has been selected.

2.1.24.1.2 Connector Pane



2.1.24.1.3 Controls and Indicators

TF	**Append Label Row(1st File)** specify whether to append label rows of the first **Result Sheet** to **Output Workhsheet**.
	Origin.WorksheetPage is the reference of the workbook that contains the analysis template.
	Error In contains error information that occur before this VI or function runs.
abc	**File List** specify the data files which will be imported into the Data Sheet.
abc	**Data Sheet** specify the source data worksheet in the analysis template. .
abc	**Result Sheet** specify analysis result sheet in the analysis template.
abc	**Origin.Output Sheet** specify the worksheet which results from Result Sheet will be appended to.
	Origin.WorksheetPage is the reference of the workbook that contains the analysis template.
	Origin.Output Worksheet refers to the **Output Sheet**.
	Error Out contains error information.

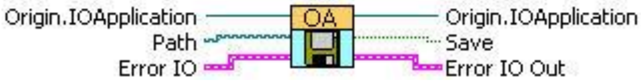
Note: This VI is available since Origin 8.5 SR0.

2.1.25 OA_Save

2.1.25.1.1 Description

Save an .opj file to disk.

2.1.25.1.2 Connector Pane



2.1.25.1.3 Controls and Indicators

	Origin.IOApplication is a reference to Origin.ApplicationSI.
	Path accepts a full path of the opj file that you are saving.
	Error In contains error information that occur before this VI or function runs.
	Origin.IOApplication.Out
TF	**Save** is **TRUE** if file is saved successfully, otherwise, saved unsuccessfully.

	Error Out contains error information.

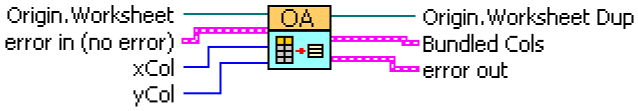
Note: This VI is available since Origin 8 SR2.

2.1.26 OA_Wks-Get-2Cols

2.1.26.1.1 Description

Bundle an x column and a y column of a worksheet to a cluster.

2.1.26.1.2 Connector Pane



2.1.26.1.3 Controls and Indicators

	Origin.Worksheet is a reference to Origin.Worksheet.
	error in(no error) contains error information that occur before this VI or function runs.
I32	**xCol** is the index of a x column.
I32	**yCol** is the index of a y column.
	Origin.Worksheet Dup is a reference to Origin.Worksheet.
	Bundled Cols is a cluster which has bundled two 1D arrays, one of which is gotten from a x column and another is gotten from a y column.
	error out contains error information.

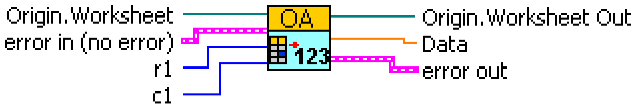
Note: This VI is available since Origin 8 SR4.

2.1.27 OA_Wks-GetCell(numeric)

2.1.27.1.1 Description

Get numeric data from a worksheet cell.

2.1.27.1.2 Connector Pane



2.1.27.1.3 Controls and Indicators

	Origin.Worksheet is a reference to Origin.Worksheet.
	error in(no error) contains error information that occur before this VI or function runs.
I32	**r1** is the row index of the cell.
I32	**c1** is the column index of the cell.
	Origin.Worksheet Dup is a reference to Origin.Worksheet.
DBL	**Data** is numeric data gotten from a worksheet cell.
	error out contains error information.

Note: This VI is available since Origin 8 SR4.

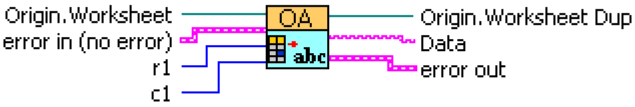
2.1.28 OA_Wks-GetCell(String)

2.1.28.1.1 Description

Get string data from a worksheet cell.

Minimum Version Required: Origin 8.5 SR0

2.1.28.1.2 Connector Pane



2.1.28.1.3 Controls and Indicators

	Origin.Worksheet is a reference to Origin.Worksheet.
	error in(no error) contains error information that occur before this VI or function runs.
I32	**r1** is the row index of the cell.
I32	**c1** is the column index of the cell.
	Origin.Worksheet Dup is a reference to Origin.Worksheet.
abc	**Data** is string data gotten from a worksheet cell.
	error out contains error information.

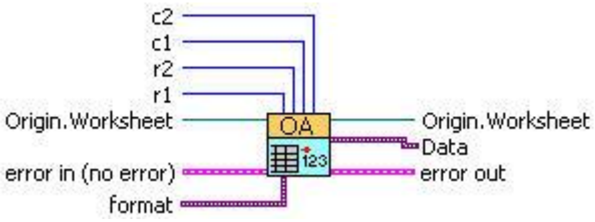
2.1.29 OA_Wks-GetRange

2.1.29.1.1 Description

Get data from a worksheet.

Minimum Version Required: Origin 8.5 SR0

2.1.29.1.2 Connector Pane



2.1.29.1.3 Controls and Indicators

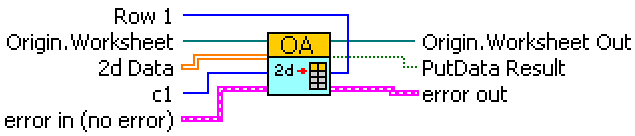
	Origin.Worksheet is a reference to Origin.Worksheet.
I32	**r1** is the start row index.
I32	**c1** is the start column index.
I32	**r2** is the end row index.
I32	**c2** is the end column index.
	format is the format of the data.
	error in(no error) contains error information that occur before this VI or function runs.
	Origin.Worksheet is a reference to Origin.Worksheet.
	Data is the output data
	error out contains error information.

2.1.30 OA_Wks-PutData

2.1.30.1.1 Description

Send a 2D array to an Origin worksheet, starting from the c1 column. Starting row index (0 offset) is specified by Row 1, which you can set to -1 to append to existing data

2.1.30.1.2 Connector Pane



2.1.30.1.3 Controls and Indicators

I32	**Row1** is the index of the start row of the import data. Default value is **0**. Please note that this option is available since Origin 8 SR6.
	Origin.Worksheet is a reference to Origin.ApplicationSI.
SGL	**2d Data** is a 2d array that will be set to a worksheet.

I32	**c1** is the index of the start column of the import data. Default value is **0**.
	error in(no error) contains error information that occur before this VI or function runs.
	Origin.Worksheet Out is a reference to Origin.Worksheet.
TF	**PutData Result** is a bool variable showing whether the "2d Data" has been inputted into the worksheet correctly.
	error out contains error information.

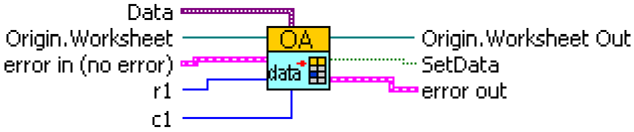
Note: This VI is available since Origin 8 SR4.

2.1.31 OA_Wks-SetCell

2.1.31.1.1 Description

Put data into a worksheet cell.

2.1.31.1.2 Connector Pane



2.1.31.1.3 Controls and Indicators

	Data is a variant, accepting different types of data (numeric or string).
	Origin.Worksheet is a reference to Origin.Worksheet.
	error in(no error) contains error information that occur before this VI or function runs.
I32	**r1** is the row index of the cell.
I32	**c1** is the column index of the cell.
	Origin.Worksheet Out is a reference to Origin.ApplicationSI.
TF	**SetData** is a bool variable which can show whether the "Data" has been inputted into the cell correctly.
	error out contains error information.

Note: This VI is available since Origin 8 SR4.

2.2 OriginAppClassics Sub-VIs

2.2.1 OriginAppClassics Sub-VIs

Note: A newer version is available for some of the Sub-VIs in this palette. Please see details in Recommended Usage for each Sub-VI in the table below. For those Invoke Node is recommended, full access to methods will be provided, while for those Property Node is recommended, full options of properties are provided.

These VIs are kept for compatibility only. You should try to stay away from using these VIs for creating new applications. The newer VIs introduced in Origin 8 are more object oriented and more efficient and much easier to use.

OriginAppClassics Sub-VIs	Description	Recommended Usage
ExponentialDecayNoise	Creates a 2D array of a sample dataset of 100 points.	
OA2DArrayToCluster	2D array found as output of GetWorksheet or GetMatrix is converted to cluster for imput to plotting.	
OABeginSession	Start a communiation session between LabVIEW and Origin .	Invoke Node
OACloseCommunication	Close a communiation session between LabVIEW and Origin .	
OACopyPage	Copy the image of an Origin Graph or Layout page to the clipboard.	Invoke Node
OACreatePage	Create a new Origin worksheet, graph, matrix, or notes page.	Invoke Node
OADestroyPage	Deletes the specified Origin page.	Invoke Node
OAEndSession	Releases the Origin session locked earlier by the BeginSession method, allowing other clients to communicate with that particular Origin session.	Invoke Node
OAExecute	Send a command string to Origin.	Invoke Node
OAExit	Shut down the Origin application.	Invoke Node
OAGetIsModified	Indicates if the Origin project has been modified and needs to be saved.	Property Node

OAGetLTStr	Get LabTalk string variables.	Invoke Node
OAGetLTVar	Get LabTalk numeric variables.	Invoke Node
OAGetMatrix	Get data from an Origin matrix.	Deprecated. Please use sub-VIs in OriginMatrix.llb.
OAGetPageString	Get Origin page textual properties or Origin Notes content.	Invoke Node
OAGetWorksheet	Get data from an Origin worksheet.	Invoke Node
OALoadProject	Open an existing Origin project file.	Please see the newer version OA_Load.vi.
OAOpenCommunication	Open communication with a single instance of the Origin application.	Please see the newer version OA_ConnectToOrigin.vi, and note that OA_ConnectToOrigin.vi would set IsModified to be false.
OAPutMatrix	Put data into an Origin matrix.	Deprecated. Please use sub-VIs in OriginMatrix.llb.
OAPutWorksheet	Place data into an Origin worksheet.	Invoke Node
OAResetProject	Clear all Origin worksheets and matrices.	Invoke Node
OA_Run	Instructs Origin to finish all auto-update calculations.	Invoke Node
OASaveProject	Save current Origin project to disk.	Please see the newer version OA_Save.vi.
OASetIsModified	Set the Origin Modified flag.	Property Node
OASetLTStr	Set LabTalk string variables.	Invoke Node
OASetLTVar	Set LabTalk numeric variables.	Invoke Node
OASetPageString	Set Origin page textual properties or Origin Notes content.	Invoke Node

OAShowHide	Sends LabTalk command to show/hide the Origin application.	Property Node.Full options of the visible state of the Origin application: MAINWND_HIDE, MAINWND_SHOW, MAINWND_SHOW_BRING_TO_FRONT, MAINWND_SHOWMAXIMIZED, and MAINWND_SHOWMINIMIZED.
OAVariant2DToArrayOrCluster	Variant found as output of GetMatrix is converted to 2D array and cluster	
SampleDataNoise	Creates a 2D array of a sample dataset of 250 points	

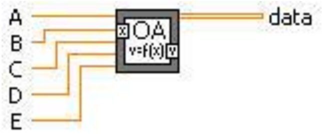
2.2.2 ExponentialDecayNoise

2.2.2.1.1 Description

Creates a 2D array of a sample dataset of 100 points using the following expression:

$$f = A * \exp(-x / B) + C * (D - \text{rand}()) + E$$

2.2.2.1.2 Connector Pane



2.2.2.1.3 Controls and Indicators

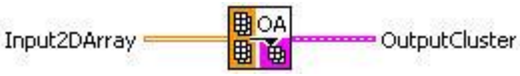
DBL	**A**
DBL	**B**
DBL	**C**
DBL	**D**
DBL	**E**
DBL	**data** is the generated noise data.

2.2.3 OA2DArrayToCluster

2.2.3.1.1 Description

2D array found as output of GetWorksheet or GetMatrix is converted to cluster for imput to plotting.

2.2.3.1.2 Connector Pane



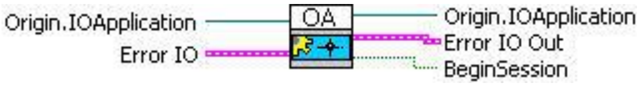
DBL	**Input2DArray** is the input 2D array data.
▤	**Output Cluster** is the output cluster.

2.2.4 OABeginSession

2.2.4.1.1 Description

Start a communiation session between LabVIEW and Origin .

2.2.4.1.2 Connector Pane



2.2.4.1.3 Controls and Indicators

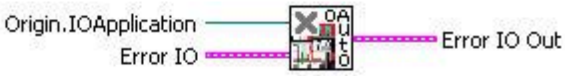
◻	**Origin.IOApplication** is a reference to Origin.ApplicationSI
▤	**Error In** contains error information that occur before this VI or function runs.
◻	**Origin.IOApplication.Out**
TF	**BeginSession** is TRUE if file is saved successfully, otherwise, save failed
▤	**Error Out** contains error information.

2.2.5 OACloseCommunication

2.2.5.1.1 Description

Close a communiation session between LabVIEW and Origin .

2.2.5.1.2 Connector Pane



2.2.5.1.3 Controls and Indicators

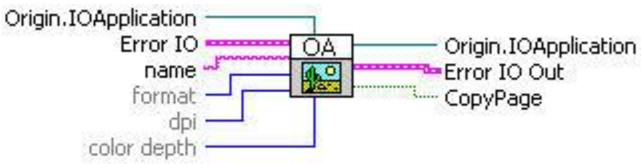
◻	**Origin.IOApplication** is a reference to Origin.ApplicationSI.
▤	**Error In** contains error information that occur before this VI or function runs.
▤	**Error Out** contains error information.

2.2.6 OACopyPage

2.2.6.1.1 Description

Copy the image of an Origin Graph or Layout page to the clipboard.

2.2.6.1.2 Connector Pane



2.2.6.1.3 Controls and Indicators

	Origin.IOApplication is a reference to Origin.ApplicationSI.
abc	**name** is the name of the page to copy.

I32	**format** is the format of the Graph. The possible values are:
	`Metafile = 0`
	`EMetafile = 1`
	`Bitmap = 2`
	`DIB = 3`

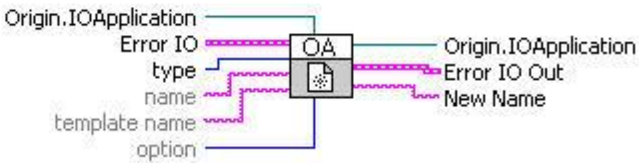
I32	**dpi** is the dpi of the Graph.
I32	**color depth** is the color depth of the Graph, which can be 1, 4, 8, 16, 24, 32.
	Error In contains error information that occur before this VI or function runs.
	Origin.IOApplication.Out
TF	**CopyPage** is TRUE if file is saved successfully, otherwise, save failed.
	Error Out contains error information.

2.2.7 OACreatePage

2.2.7.1.1 Description

Create a new Origin worksheet, graph, matrix, or notes page.

2.2.7.1.2 Connector Pane



2.2.7.1.3 Controls and Indicators

	Origin.IOApplication is a reference to Origin.ApplicationSI.

I32	**type** is the type of the page. The possible values are:
	Worksheet = 2
	Graph = 3
	Matrix = 5
	Notes = 9

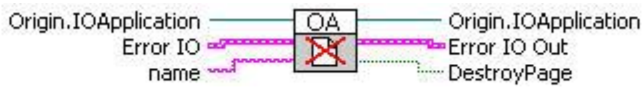
abc	**name** is the name of the page to be created.
abc	**template name** is the template used to create the page.
I32	**option** controls whether to show the created page, 1 for show and 0 for hidden.
	Error In contains error information that occur before this VI or function runs.
	Origin.IOApplication.Out is a reference to Origin.ApplicationSI.
abc	**New Name** returns the name of the new page
	Error Out contains error information.

2.2.8 OADestroyPage

2.2.8.1.1 Description

Deletes the specified Origin page.

2.2.8.1.2 Connector Pane



2.2.8.1.3 Controls and Indicators

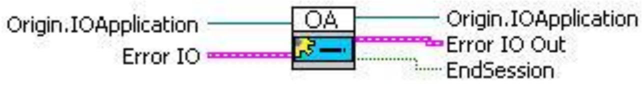
	Origin.IOApplication is a reference to Origin.ApplicationSI
abc	**name** is the name of the page to delete
	Error In contains error information that occur before this VI or function runs.
	Origin.IOApplication.Out
TF	**DestroyPage** is TRUE, page is destroyed successfully, otherwise, destroy failed
	Error Out contains error information.

2.2.9 OAEndSession

2.2.9.1.1 Description

Releases the Origin session locked earlier by the BeginSession method, allowing other clients to communicate with that particular Origin session.

2.2.9.1.2 Connector Pane



2.2.9.1.3 Controls and Indicators

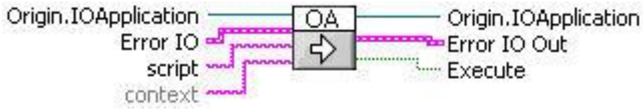
	Origin.IOApplication is a reference to Origin.ApplicationSI.
	Error In contains error information that occur before this VI or function runs.
	Origin.IOApplication.Out
TF	**EndSession** is TRUE, the operation is successfully, otherwise, the operation failed.
	Error Out contains error information.

2.2.10 OAExecute

2.2.10.1.1 Description

Send a command string to Origin.

2.2.10.1.2 Connector Pane



2.2.10.1.3 Controls and Indicators

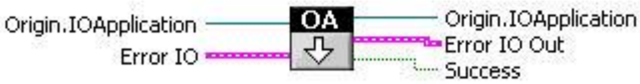
	Origin.IOApplication is a reference to Origin.ApplicationSI.
abc	**script** is the script to execute.
abc	**context** is the context of the script.
	Error In contains error information that occur before this VI or function runs.
	Origin.IOApplication.Out
TF	**Execute** is TRUE if the operation is successfully, otherwise, the operation failed.
	Error Out contains error information.

2.2.11 OAExit

2.2.11.1.1 Description

Shut down the Origin application.

2.2.11.1.2 Connector Pane



2.2.11.1.3 Controls and Indicators

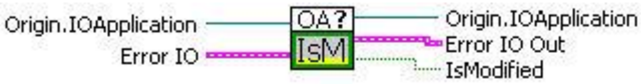
	Origin.IOApplication is a reference to Origin.ApplicationSI.
	Error In contains error information that occur before this VI or function runs.
	Origin.IOApplication.Out is a reference to Origin.ApplicationSI.
TF	**Success** is TRUE, Origin shuts down successfully, otherwise, Origin fails to shut down.
	Error Out contains error information.

2.2.12 OAGetIsModified

2.2.12.1.1 Description

Indicates if the Origin project has been modified and needs to be saved.

2.2.12.1.2 Connector Pane



2.2.12.1.3 Controls and Indicators

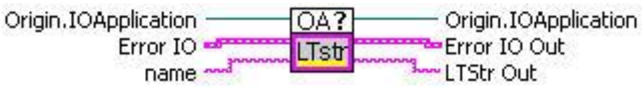
	Origin.IOApplication is a reference to Origin.ApplicationSI.
	Error In contains error information that occur before this VI or function runs.
	Origin.IOApplication.Out is a reference to Origin.ApplicationSI.
TF	**IsModified** Indicates if the Origin project has been modified and needs to be saved
	Error Out contains error information.

2.2.13 OAGetLTStr

2.2.13.1.1 Description

Get LabTalk string variables.

2.2.13.1.2 Connector Pane



2.2.13.1.3 Controls and Indicators

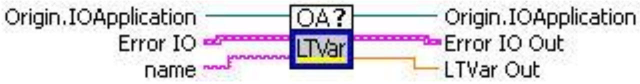
	Origin.IOApplication is a reference to Origin.ApplicationSI.
abc	**name** is the name of the string variable.
	Error In contains error information that occur before this VI or function runs.
	Origin.IOApplication.Out is a reference to Origin.ApplicationSI.
Abc	**LTStr Out** is the value of the string variable specified by **name**
	Error Out contains error information.

2.2.14 OAGetLTVar

2.2.14.1.1 Description

Get LabTalk numeric variables.

2.2.14.1.2 Connector Pane



2.2.14.1.3 Controls and Indicators

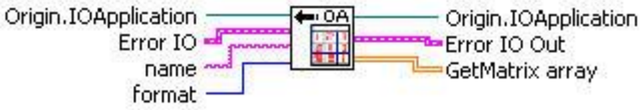
	Origin.IOApplication is a reference to Origin.ApplicationSI.
abc	**name** is the name of the string variable.
	Error In contains error information that occur before this VI or function runs.
	Origin.IOApplication.Out is a reference to Origin.ApplicationSI.
DBL	**LTVar Out** is the value of the numeric variable specified by **name**
	Error Out contains error information.

2.2.15 OAGetMatrix

2.2.15.1.1 Description

Get data from an Origin matrix.

2.2.15.1.2 Connector Pane



2.2.15.1.3 Controls and Indicators

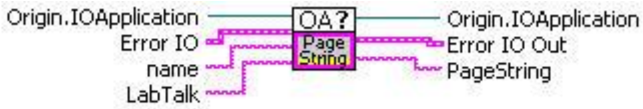
	Origin.IOApplication is a reference to Origin.ApplicationSI.
abc	**name** is the name of the matrix.
I32	**format** is the format of the data.
	Error In contains error information that occur before this VI or function runs.
	Origin.IOApplication.Out is a reference to Origin.ApplicationSI.
DBL	**GetMatrix array** is the data get from the matrix.
	Error Out contains error information.

2.2.16 OAGetPageString

2.2.16.1.1 Description

Get Origin page textual properties or Origin Notes content.

2.2.16.1.2 Connector Pane



2.2.16.1.3 Controls and Indicators

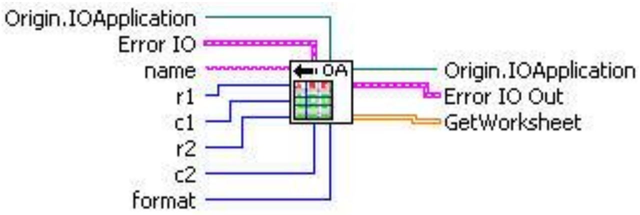
	Origin.IOApplication is a reference to Origin.ApplicationSI.
abc	**name** is the name of the page.
abc	**Labtalk** is the labtalk script.
	Error In contains error information that occur before this VI or function runs.
	Origin.IOApplication.Out is a reference to Origin.ApplicationSI.
abc	**PageString** is the Origin page textual properties or Origin Notes content.
	Error Out contains error information.

2.2.17 OAGetWorksheet

2.2.17.1.1 Description

Get data from an Origin worksheet.

2.2.17.1.2 Connector Pane



2.2.17.1.3 Controls and Indicators

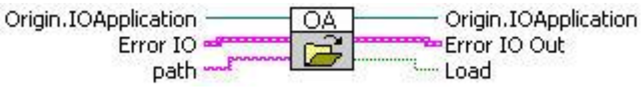
	Origin.IOApplication is a reference to Origin.ApplicationSI.
abc	**name** is the name of the worksheet.
I32	**r1** is the start row index.
I32	**c1** is the start column index.
I32	**r2** is the end row index.
I32	**c2** is the end column index.
I32	**format** is the format of the data.
	Error In contains error information that occur before this VI or function runs.
	Origin.IOApplication.Out is a reference to Origin.ApplicationSI
DBL	**GetWorksheet** is the data get from the worksheet.
	Error Out contains error information.

2.2.18 OALoadProject

2.2.18.1.1 Description

Open an existing Origin project file.

2.2.18.1.2 Connector Pane



2.2.18.1.3 Controls and Indicators

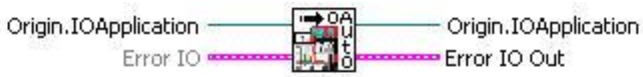
	Origin.IOApplication is a reference to Origin.ApplicationSI.
abc	**path** is the path of the project to load.
	Error In contains error information that occur before this VI or function runs.
	Origin.IOApplication.Out is a reference to Origin.ApplicationSI.
TF	**Load** is TRUE, project is loaded successfully, otherwise, load failed.
	Error Out contains error information.

2.2.19 OAOpenCommunication

2.2.19.1.1 Description

Open communication with a single instance of the Origin application.

2.2.19.1.2 Connector Pane



2.2.19.1.3 Controls and Indicators

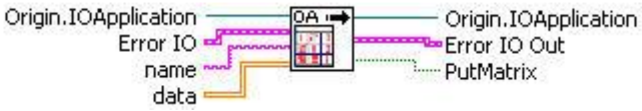
	Origin.IOApplication is a reference to Origin.ApplicationSI.
	Error In contains error information that occur before this VI or function runs.
	Origin.IOApplication.Out is a reference to Origin.ApplicationSI.
	Error Out contains error information.

2.2.20 OAPutMatrix

2.2.20.1.1 Description

Put data into an Origin matrix.

2.2.20.1.2 Connector Pane



2.2.20.1.3 Controls and Indicators

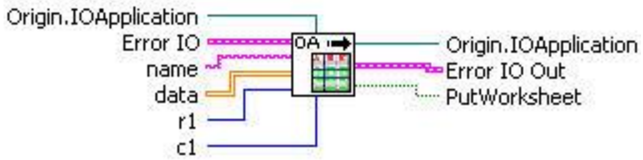
	Origin.IOApplication is a reference to Origin.ApplicationSI.
abc	**name** is the name of the matrix.
DBL	**data** is the data to put in matrix.
	Error In contains error information that occur before this VI or function runs.
	Origin.IOApplication.Out
TF	**PutMatrix** is TRUE, the data are put into the matrix successfully, otherwise failed.
	Error Out contains error information.

2.2.21 OAPutWorksheet

2.2.21.1.1 Description

Place data into an Origin worksheet.

2.2.21.1.2 Connector Pane



2.2.21.1.3 Controls and Indicators

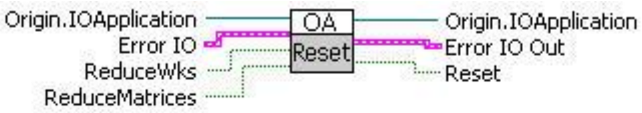
	Origin.IOApplication is a reference to Origin.ApplicationSI.
abc	**name** is the name of the worksheet.
DBL	**data** is the data to put in worksheet.
I32	**r1** is the start row index to put the data.
I32	**c1** is the start column index to put the data.
	Error In contains error information that occur before this VI or function runs.
	Origin.IOApplication.Out is a reference to Origin.ApplicationSI.
TF	**PutWorksheet** is TRUE, the data are put into the worksheet successfully, otherwise failed.
	Error Out contains error information.

2.2.22 OAResetProject

2.2.22.1.1 Description

Clear all Origin worksheets and matrices.

2.2.22.1.2 Connector Pane



2.2.22.1.3 Controls and Indicators

	Origin.IOApplication is a reference to Origin.ApplicationSI.
TF	**ReduceWks** controls whether to reduce the size of the worksheet.
TF	**ReduceMatrices** controls whether to reduce the size of the matrix.
	Error In contains error information that occur before this VI or function runs.
	Origin.IOApplication.Out is a reference to Origin.ApplicationSI.
TF	**Reset** is TRUE, the project is reset successfully, otherwise failed.

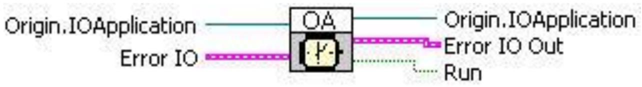
	Error Out contains error information.

2.2.23 OARun

2.2.23.1.1 Description

Instructs Origin to finish all auto-update calculations.

2.2.23.1.2 Connector Pane



2.2.23.1.3 Controls and Indicators

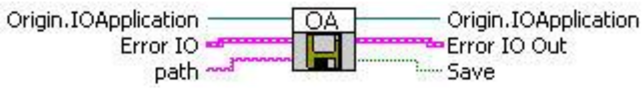
	Origin.IOApplication is a reference to Origin.ApplicationSI.
	Error In contains error information that occur before this VI or function runs.
	Origin.IOApplication.Out is a reference to Origin.ApplicationSI.
TF	**Run** is TRUE, the operation is successful, otherwise failed.
	Error Out contains error information.

2.2.24 OASaveProject

2.2.24.1.1 Description

Save current Origin project to disk.

2.2.24.1.2 Connector Pane



2.2.24.1.3 Controls and Indicators

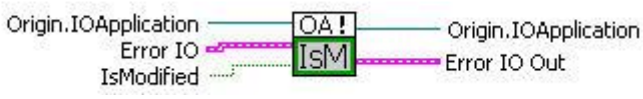
	Origin.IOApplication is a reference to Origin.ApplicationSI.
abc	**path** is the path to save project in.
	Error In contains error information that occur before this VI or function runs.
	Origin.IOApplication.Out
TF	**Save** is TRUE, the project is saved successfully, otherwise failed.
	Error Out contains error information.

2.2.25 OASetIsModified

2.2.25.1.1 Description

Set the Origin Modified flag.

2.2.25.1.2 Connector Pane



2.2.25.1.3 Controls and Indicators

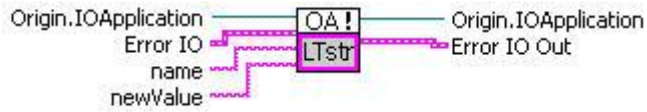
	Origin.IOApplication is a reference to Origin.ApplicationSI.
TF	**IsModified** set Origin Modified flag.
	Error In contains error information that occur before this VI or function runs.
	Origin.IOApplication.Out is a reference to Origin.ApplicationSI.
	Error Out contains error information.

2.2.26 OASetLTStr

2.2.26.1.1 Description

Set LabTalk string variables.

2.2.26.1.2 Connector Pane



2.2.26.1.3 Controls and Indicators

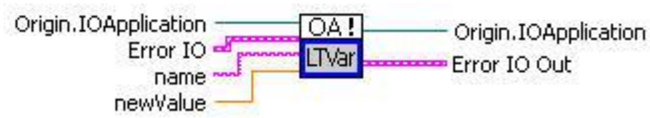
	Origin.IOApplication is a reference to Origin.ApplicationSI.
abc	**name** is the name of labtalk string variable.
abc	**newValue** is the value to set to the variable.
	Error In contains error information that occur before this VI or function runs.
	Origin.IOApplication.Out is a reference to Origin.ApplicationSI.
	Error Out contains error information.

2.2.27 OASetLTVar

2.2.27.1.1 Description

Set LabTalk numeric variables.

2.2.27.1.2 Connector Pane



2.2.27.1.3 Controls and Indicators

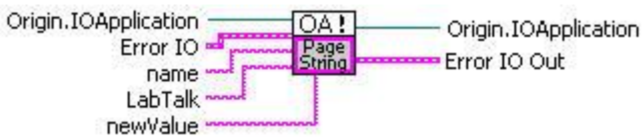
	Origin.IOApplication is a reference to Origin.ApplicationSI.
abc	**name** is the name of labtalk numeric variable.
DBL	**newValue** is the value to set to the variable.
	Error In contains error information that occur before this VI or function runs.
	Origin.IOApplication.Out is a reference to Origin.ApplicationSI.
	Error Out contains error information.

2.2.28 OASetPageString

2.2.28.1.1 Description

Set Origin page textual properties or Origin Notes content.

2.2.28.1.2 Connector Pane



2.2.28.1.3 Controls and Indicators

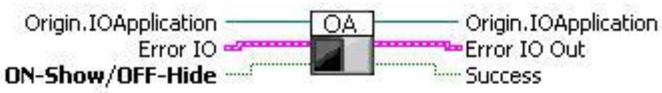
	Origin.IOApplication is a reference to Origin.ApplicationSI.
abc	**name** is the name of the string to set.
abc	**LabTalk** is the labtalk script to set.
abc	**newValue** is the value to set to the string.
	Error In contains error information that occur before this VI or function runs.
	Origin.IOApplication.Out is a reference to Origin.ApplicationSI.
	Error Out contains error information.

2.2.29 OAShowHide

2.2.29.1.1 Description

Sends LabTalk command to show/hide the Origin application.

2.2.29.1.2 Connector Pane



2.2.29.1.3 Controls and Indicators

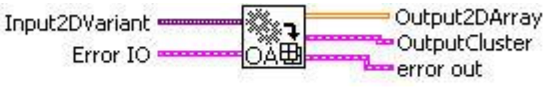
	Origin.IOApplication is a reference to Origin.ApplicationSI。
TF	**ON-Show/OFF-Hide** set Origin Show or Hide.
	Error In contains error information that occur before this VI or function runs.
	Origin.IOApplication.Out
TF	**Success** is TRUE, the operation is successful, otherwise failed.
	Error Out contains error information.

2.2.30 OAVariant2DToArrayOrCluster

2.2.30.1.1 Description

Variant found as output of GetMatrix is converted to 2D array and cluster

2.2.30.1.2 Connector Pane



2.2.30.1.3 Controls and Indicators

	Input2DVariant is the input variant matrix.
	Error In contains error information that occur before this VI or function runs.
DBL	**Output2DArray** is the output matrix data.
	OutputCluster is the output cluster.
	Error Out contains error information.

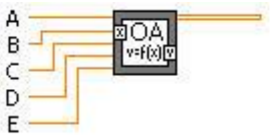
2.2.31 SampleDataNoise

2.2.31.1.1 Description

Creates a 2D array of a sample dataset of 250 points, using the following expression:

f = A * exp(-(x - B)^2 / C) + D * (E + rand())

2.2.31.1.2 Connector Pane



2.2.31.1.3 Controls and Indicators

DBL	**A**
DBL	**B**
DBL	**C**
DBL	**D**
DBL	**E**
DBL	**data** is the sample dataset generated.

2.3 OriginMatrix Sub-VIs

2.3.1 OriginMatrix Sub-VIs

These VIs are for setting and getting data from Origin matrix objects in an Origin project.

OriginMatrix Sub-VIs	Description
OA_FindMatrixSheet	Find a MatrixSheet from the range string.
OA_GetMatrix	Get a Matrix object by index(0,1,2..) from a MatrixSheet.
OA_Mat-GetData	Get data as 2D array from an Origin Matrix Object.
OA_Mat-SetData	Send a 2D numeric array to an Origin Matrix Object.
OA_NewMatrixsheet	Create a new Matrix Book with a single MatrixSheet with a single MatrixObject.

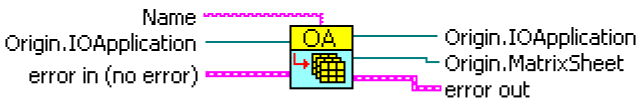
2.3.2 OA FindMatrixSheet

2.3.2.1.1 Description

Find a MatrixSheet from the range string, like [MBook1]MSheet1. If matrix book has only one sheet, then can just use the book name, like Mbook1. If the active window in Origin is a matrix book, you can also use an empty string to get the active matrix book's active sheet.

Minimum Version Required: Origin 8.5 SR0

2.3.2.1.2 Connector Pane



2.3.2.1.3 Controls and Indicators

abc	**Name** specifies the name of the sheet that needs to be found.
	Origin.IOApplication is a reference to Origin.ApplicationSI.
	error in(no error) contains error information that occur before this VI or function runs.
	Origin.IOApplication is a reference to Origin.ApplicationSI.
	error out contains error information.
	Origin.MatrixSheet is the sheet that has been found.

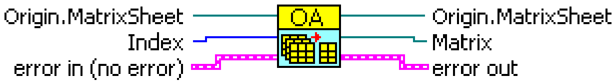
2.3.3 OA GetMatrix

2.3.3.1.1 Description

Get a Matrix object by index(0,1,2..) from a MatrixSheet.

Minimum Version Required: Origin 8.5 SR0

2.3.3.1.2 Connector Pane



2.3.3.1.3 Controls and Indicators

	Origin.MatrixSheet is a reference to Origin.MatrixSheet.
I32	**Index** indicates which matrix object will be accessed.
	error in(no error) contains error information that occur before this VI or function runs.
	Origin.MatrixSheet is a reference to Origin.MatrixSheet.
	Matrix is the matrix object that has been found.
	error out contains error information.

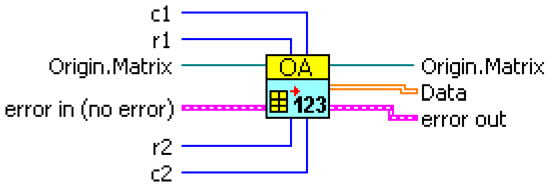
2.3.4 OA Mat-GetData

2.3.4.1.1 Description

Get data as 2D array from an Origin Matrix Object.

Minimum Version Required: Origin 8.5 SR0

2.3.4.1.2 Connector Pane



2.3.4.1.3 Controls and Indicators

I32	**c1** is the index of first column to get, using 0 to start from the 1st column.
I32	**r1** is the index of first row to get, using 0 to start from the 1st row.
	Origin.Matrix is the matrix object from which we will access data.
	error in(no error) contains error information that occur before this VI or function runs.
I32	**c2** uses -1 to indicate last column, or a 0-offset column index as the last column to get.
I32	**r2** uses -1 to indicate last row, or a 0-offset row index as the last row to get.
	Origin.Matrix is a reference to Origin.Matrix.
DBL	**Data** is the 2D array accessed from matrix object.
	error out contains error information.

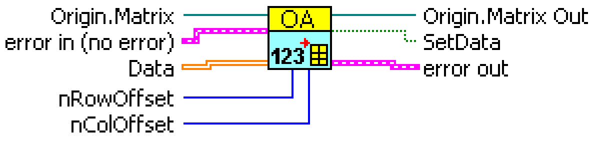
2.3.5 OA Mat-SetData

2.3.5.1.1 Description

Send a 2D numeric array to an Origin Matrix Object.

Minimum Version Required: Origin 8.5 SR0

2.3.5.1.2 Connector Pane



2.3.5.1.3 Controls and Indicators

	Origin.Matrix is the matrix object which we will send data to.
	error in(no error) contains error information that occur before this VI or function runs.

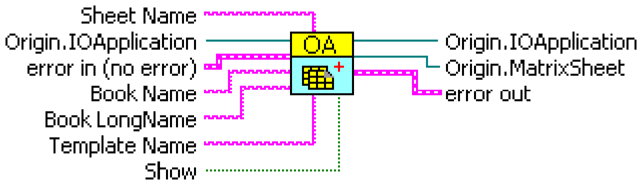
DBL	**Data** is a 2D array.
I32	**nRowOffset** uses 0-offset index to start, or 0 for entire matrix.
I32	**nColOffset** uses 0-offset index to start, or 0 for entire matrix.
	Origin.Matrix Out is a reference to Origin.Matrix.
TF	If **SaveData** is TRUE, the data is set successfully, otherwise, set failed
	error out contains error information.

2.3.6 OA NewMatrixsheet

2.3.6.1.1 Description

Create a new Matrix Book with a single MatrixSheet with a single MatrixObject. By default, the origin.otm template will be used.

2.3.6.1.2 Connector Pane



2.3.6.1.3 Controls and Indicators

abc	**Sheet Name** is the name of the new sheet.
	Origin.IOApplication is a reference to Origin.ApplicationSI.
	error in(no error) contains error information that occur before this VI or function runs.
abc	**Book Name** is the short name of the book that the new sheet belongs to.
abc	**Book LongName** specifies the long name of the book that the new sheet belongs to.
abc	**Template Name** is the name of the template.
TF	**Show** means whether to activate the created matrix sheet.
	Origin.IOApplication is a reference to Origin.ApplicationSI.
	Origin.MatrixSheet is the sheet that has been created.
	error out contains error information.

Note: This VI is available since Origin 8 SR4.

2.4 OriginWave Sub-VIs

2.4.1 OriginWave Sub-VIs

This group of sub-VIs is for handling LabVIEW Waveform data, the same type of data that you would typically wire into a Chart. Origin has native waveform support in worksheet columns, so these VIs allows you to directly connect a waveform to Origin.

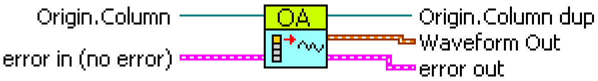
OriginWave Sub-VIs	Description
OA Col-GetWave	Get a waveform from a column.
OA Col-SetWaveInfo	Extract a wave's attributes: NI_ChannelName, NI_UnitDescription, NI_DeviceNumber and set them as a column's long name, Unit, User Parameter.
OA Col-NewWaveFile	Connect to Origin and start a new project with a single worksheet and optionally plot all the columns in a graph.
OA Wks-GetWave	Retrieve continues data from a worksheet to create dynamic data.
OA Wks-PutWave	Send 1D array of waveforms to an Origin worksheet starting from 1st column.

2.4.2 OA Col-GetWave

2.4.2.1.1 Description

Get a waveform from a column. If the column has sampling interval info, then the waveform attributes will also be properly set.

2.4.2.1.2 Connector Pane



2.4.2.1.3 Controls and Indicators

	Origin.Column is a reference to Origin.Column.
	error in(no error) contains error information that occur before this VI or function runs.
	Origin.Column Dup is a reference to Origin.Column.
	Waveform Out returns waveform which transfered from a column.
	error out contains error information.

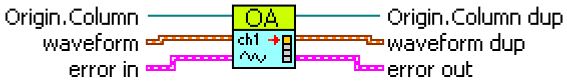
Note: This VI is available since Origin 8 SR4.

2.4.3 OA Col-SetWaveInfo

2.4.3.1.1 Description

Extract a wave's attributes: NI_ChannelName, NI_UnitDescription, NI_DeviceNumber and set them as a column's long name, Unit, User Parameter.

2.4.3.1.2 Connector Pane



2.4.3.1.3 Controls and Indicators

	Origin.Column is a reference to Origin.Column.
	waveform is the wavform data.
	error in contains error information that occur before this VI or function runs.
	Origin.Column Dup is a reference to Origin.Column.
	Waveform Dup is the duplicate waveform data.
	error out contains error information.

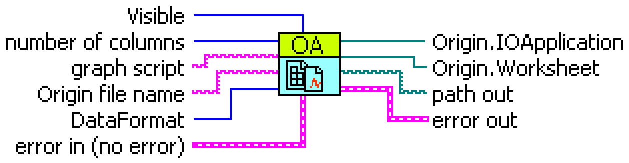
Note: This VI is available since Origin 8 SR4.

2.4.4 OA NewWaveFile

2.4.4.1.1 Description

Connect to Origin and start a new project with a single worksheet and optionally plot all the columns in a graph.

2.4.4.1.2 Connector Pane



2.4.4.1.3 Controls and Indicators

I32	**Visible** is to decide which status of Origin will be, like hidden, show, etc.
I32	**number of columns** is the number of columns of the new creating worksheet.

abc	**graph script** is the labtalk script which will plot all the columns in a graph.
abc	**Origin file name** is a string of your filename or file path.

I32	**DataFormat** is the data type of the column. It must be one of the following values:
	`DF_NO_CHANGE =-1`
	`DF_DOUBLE =0`
	`DF_TEXT =1`
	`DF_TIME = 2`
	`DF_DATE = 3`
	`DF_TEXT_NUMERIC = 9`
	`DF_FLOAT = 32`
	`DF_SHORT = 33`
	`DF_LONG =34`
	`DF_CHAR =35`
	`DF_BYTE = 38`
	`DF_USHORT = 39`
	`DF_ULONG =40`
	`DF_COMPLEX = 41`
	The default value is **DF_DOUBLE**.

	error in(no error) contains error information that occur before this VI or function runs.
	Origin.IOApplication is a reference to Origin.ApplicationSI.
	Origin.Worksheet is the new created worksheet.
	path out is a full path of the file.
	error out contains error information.

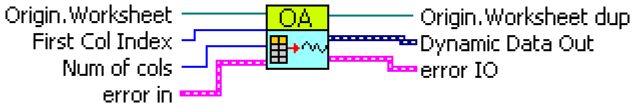
Note: This VI is available since Origin 8 SR4.

2.4.5 OA Wks-GetWave

2.4.5.1.1 Description

Retrieve continues data from a worksheet to create dynamic data. First Col Index is zero offset.

2.4.5.1.2 Connector Pane



2.4.5.1.3 Controls and Indicators

	Origin.Worksheet is a reference to Origin.Worksheet.
I32	**First Col Index** is the index of the first column that will be used to create dynamic data. It is zero offset. And default value is **0**.
I32	**Num of cols** means how many columns involved into creating dynamic data. Default value is **-1**, which indicates that all of the columns will be used to create dynamic data.
	error in contains error information that occur before this VI or function runs.
	Origin.Worksheet Dup is a reference to Origin.Worksheet.
	Dynamic Data Out returns the dynamic data which transferred from the data of several columns of a worksheet.
	error IO contains error information.

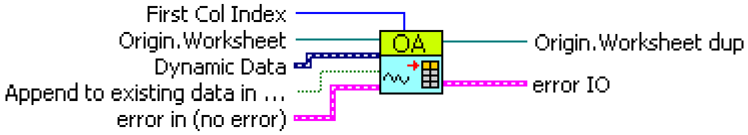
Note: This VI is available since Origin 8 SR4.

2.4.6 OA Wks-PutWave

2.4.6.1.1 Description

Send 1D array of waveforms to an Origin worksheet starting from 1st column. Waveform attributes are extracted to put into Column labels and Sampling Intervals.

2.4.6.1.2 Connector Pane



2.4.6.1.3 Controls and Indicators

I32	**First Col Index** is the index of the first column that will be used to put the dynamic data. It is zero offset. And default value is **0**.
	Origin.Worksheet is a reference to Origin.Worksheet.
	Dynamic Data is the data will put into Origin worksheet.
TF	**Append to exsiting data in...** decides whether the **Dynamic Data** will be appended to esiting data in the worksheet.

	error in(no error) contains error information that occur before this VI or function runs.
	Origin.Worksheet Dup is a duplication of the inputed worksheet.
	error IO contains error information.

Note: This VI is available since Origin 8 SR4.

3 LabVIEW Tutorials

These tutorials serve as a step-by-step guide for a new LabVIEW user to create simple VIs, such as to generate some simulated data and save them into an Origin OPJ file.

More experienced LabVIEW users can look directly at the samples provided in the following Origin installation folder:

```
<exe>\Samples\COM Server and Client\LabVIEW\
```

Notes: All of these Tutorials are written with LabVIEW 7 Express.

Tutorials included in this section:

3.1 Creating a Simple VI to Save Data to Origin

In this tutorial, you will create a VI to send data from LabVIEW to an Origin column of an Origin worksheet.

1. Run Origin. If Origin is already running, start a new project.

2. Run LabVIEW and new a blank VI.

3. Activate the **Block Diagram** window and right-click inside the window.

4. From the **Functions** palette, choose **All Functions**, then **User Libraries**, then **paletteMenu**.

5. Choose **OA_ConnectToOrigin(App)** VI and place it in the window.

6. Right-click in the window again, and select **OA_FindAddWorksheet** VI from the same palette as in the previous step.

7. Wire the **Origin.IOApplication** and **Error IO** connectors to the two VIs. Make sure to choose output side connectors for the OA_ConnectToOrigin(App) VI and input side connectors for the OA_FindAddWorksheet VI.

8. Add a string constant and assign to it the string **[Book1]Sheet2**.

9. Wire the string constant to the **Worksheet Name** connector of **OA_FindAddWorksheet** VI. (Hint: You can click on any Origin VI you place in the window, then go to the Help menu and select Show Context Help. This will open the contextual help window with specifics on the connectors to the VI).

10. Add an **OA_GetColumn** VI from the Origin VI palette, and wire the **Origin.Worksheet** and **Error IO** connectors to the corresponding connectors of the **OA_FindAddWorksheet VI**.

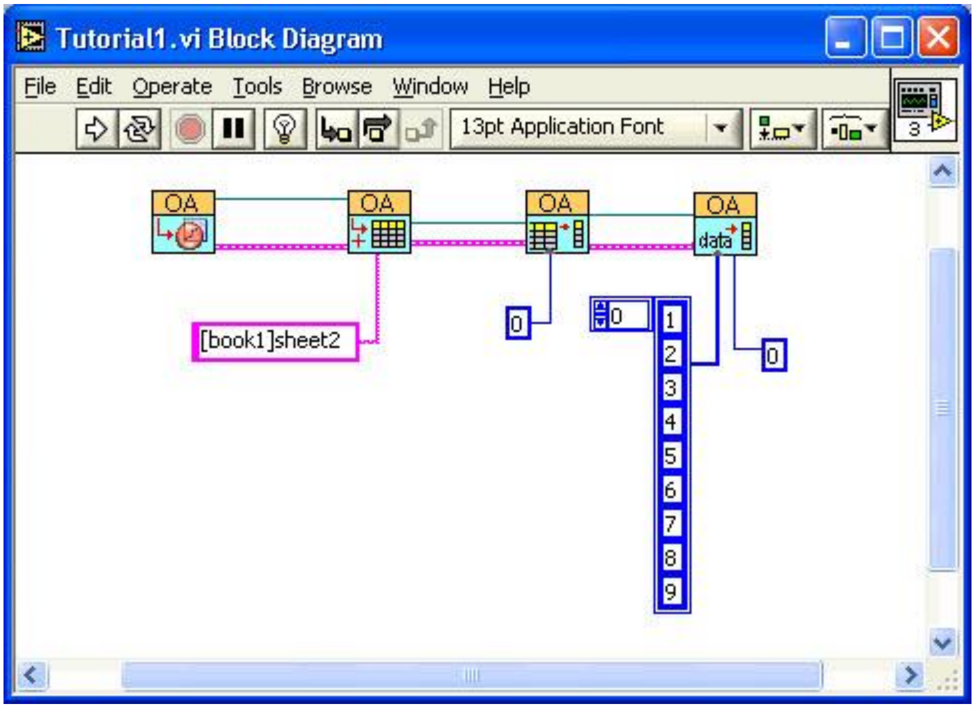
11. Add a numeric constant and set it to 0, and wire it to the **Column Name/Index** connector of **OA_GetColumn** VI.

12. Add an **OA_Col-SetData** VI from the Origin VI palette, and wire the **Origin.Column** and **Error IO** connectors to the corresponding connectors of the **OA_GetColumn** VI.

13. Add a numeric constant and set it to 0, and wire it to the **nOffset** connector of **OA_Col-SetData** VI.

14. Add an array constant and set it as you want, and wire it to the **data** connector of **OA_Col-SetData** VI.

15. When the wiring is completed, your Block Diagram window should look like this:

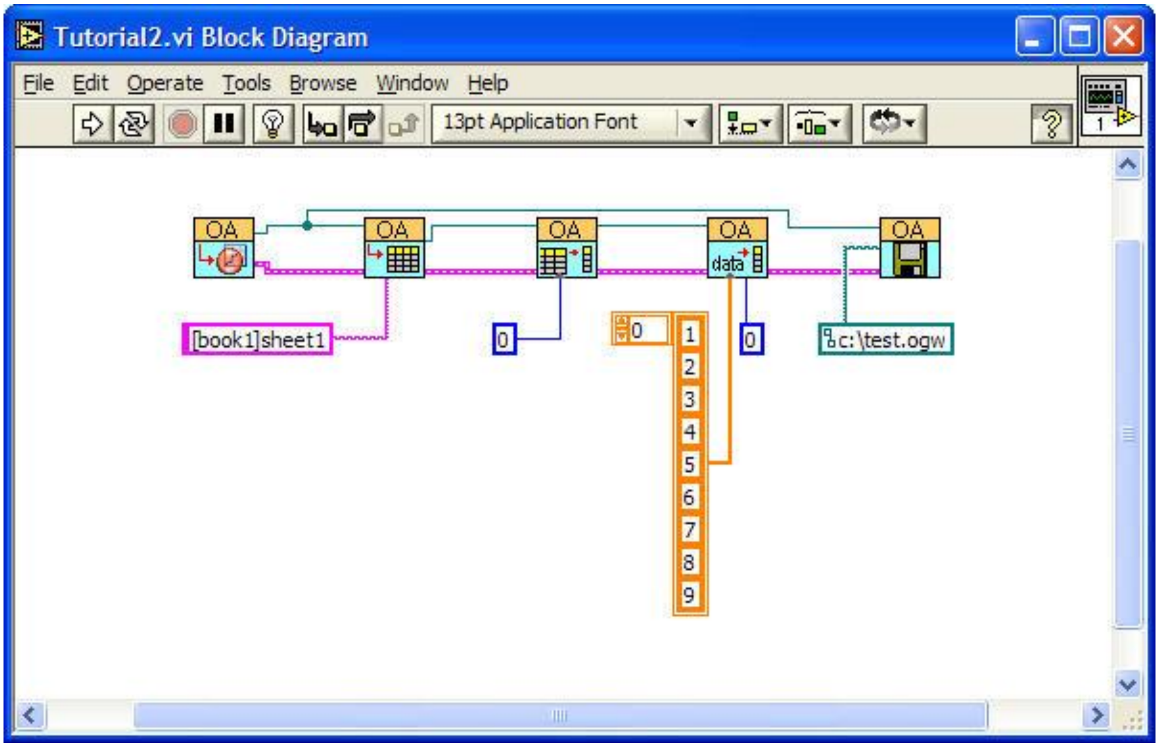


16. Click the **Run** button to send the data to Origin. You should see the data in the [Book1]Sheet2 worksheet. Try changing the **Column** number controls to place data in a different location in the worksheet.

3.2 Sending Data to Save into OGW

In this tutorial, you will continue from **Creating a Simple VI to Save Data to Origin** to save the workbook as an OGW file.

1. Perform operations in Creating a Simple VI to Save Data to Origin.
2. Add an **OA_Save** VI from the Origin VI palette, and wire the **Origin.IOApplication** connector to the corresponding connector of **OA_ConnectToOrigin** VI and the **Error IO** connector to the corresponding connector of **OA_Col-SetData** VI.
3. Add a path constant and set the path as **c:\test.ogw**, and wire it to the **Path** connector of **OA_Save** VI.
4. When the wiring is complete, your Block Diagram window should look like this:



5. Click the **Run** button to send the data to Origin. You should see the data in the [Book1]Sheet1 worksheet and find that a file named test.ogw appears under c:\.

3.3 Plotting with a User-Defined Template

In this tutorial, we will show you how to plot with a user-defined graph template in Labview. Make sure there is a graph template named **MyGraphTemplate** under path "C:\". (Hint: you can use the the graph template in **Tutorial: Customizing Graphs: Customizing a Graph**)

1. Run Origin. Make sure that there is a Workbook with the default "Book1" and "Sheet1". Fill the first two column with row numbers by selecting the two columns and right click, choose **Row Numbers** from **Fill Columns With**.

2. Select **File: Import: Single ASCII** from the menu, then find **Linear Fit.dat** from **<Origin installation folder>\Samples\Curve Fitting** and make sure the **Show Options Dilalog** checkbox has been selected. Select **Start New Column** for **Import Options: Import Mode**. Unwrap the **(Re)Naming Worksheet and Workbook** branch, disable the **Rename Sheet with(Partial) Filename** and the **Rename Book with(Partial) Filename** checkboxes. Then click **OK** button to import the file.

3. Run Labview, create a new VI and active the Block Diagram window. Then choose **OA_ConnectToOrigin** VI and place it to the window.

4. Select and place **OA_FindWorksheet** VI , then wire the **Origin.IOApplication** and **Error IO** connectors to the corresponding connectors of the **OA_ConnectToOrigin** VI.

5. Add a string constant and assign to it the string **[Book1]Sheet1**. Wire the string constant to the **Name** connector of **OA_FindWorksheet** VI.

6. Select and place **OA_GetColumn** VI, then wire the **Origin.Worksheet** and **Error IO** connectors to the corresponding connectors of the **OA_FindWorksheet** VI.

7. Add a numeric constant and set it to 2, and wire it to the **Column Name/Index** connector of **OA_GetColumn** VI.

8. Select and place **OA_Col-Setting** VI , then wire the **Origin.Column** and **Error IO** connectors to the corresponding connectors of the **OA_GetColumn** VI.

9. Add a numeric constant and set it to 3, and wire it to the **Type** connector of **OA_Col-Setting** VI to set the type of column 2 to X, since Only XY type of plots are supported with **OA_PlotWksCols** VI.

10. Select and place **OA_FindWorksheet** VI again, then wire the **Origin.IOApplication** connector to the corresponding connectors of the first**OA_FindWorksheet** VI, and wire the **Error IO** connector to the corresponding connectors of the **OA_Col-Setting** VI.

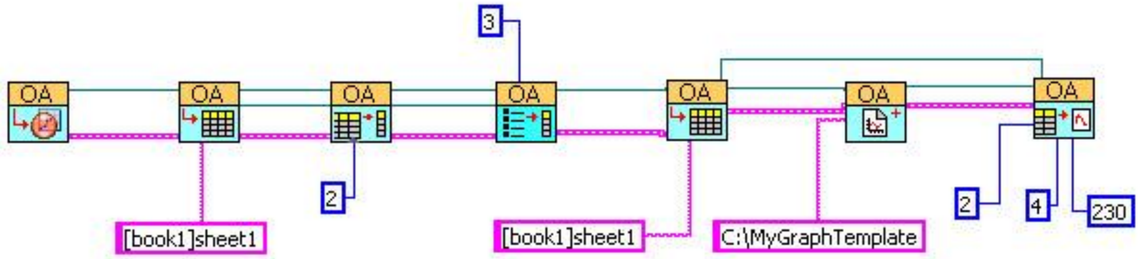
11. Add a string constant and assign to it the string **[Book1]Sheet1**. Wire the string constant to the **Name** connector of the second **OA_FindWorksheet** VI.

12. Select and place **OA_NewEmptyGraph** VI, then wire the **Origin.IOApplication** and the **Error IO** to the corresponding connector of the second **OA_FindWorksheet** VI.

13. Add a string constant and assign to it the string **C:\MyGraphTemplate**. Wire the string constant to the **Template Name** connector of the second **OA_NewEmptyGraph** VI.

14. Select and place **OA_PlotWksCols** VI, then wire the **Origin.Worksheet in** connector to the corresponding connector of the Second **OA_FindWorksheet** VI. Wire the **Origin.GraphLayer in** and **Error IO** connectors to the corresponding connector of **OA_NewEmptyGraph** VI. Then create three numeric constant and set them to **2**, **4**, **230**. Connect **2**, **4**, **230** to **c1**, **c2**, **plotType** connectors separately.

15. When all of these are finished, the Block Diagram should look like:



16. Click **Run** button to run this VI and then you can turn to Origin to see the result.

3.4 Working with Origin Analysis Template

Origin Template is very useful to perform automatic analysis just like a macro. In this tutorial, you will learn how to work with an Origin Analysis Template for interpolation in Labview.

1. Open Origin, import **<exe>\Samples\Import and Export\ASCII Simple.dat** into Sheet1 of Book1.

2. Add one new column and fill some data in it, and select **Analysis: Mathematics: Interpolate/Extrapolate Y From X** to do **Cubic-Spline** interpolating. Please set **Recalculate** to **Auto**.

3. Clear the data in Column C and save the worksheet as **interpolate.ogw** under the folder where you want to place this VI.

4. Start Labview, and new a blank VI. Active the Block Diagram window.

5. From the **Functions** palette, choose **All Functions**, then **User Libraries**, then **paletteMenu**.

6. Choose **OA_ConnectToOrigin** VI and place it in the window.

7. Place **OA_Load** , **OA_FindWorksheet** , **OA_GetColumn** , and **OA_Col-SetData** as the previous step.

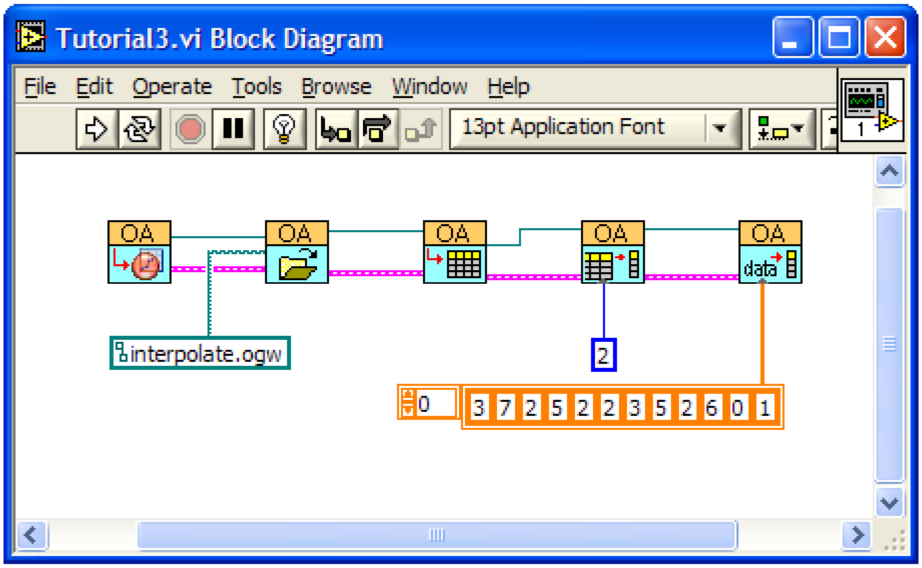
8. Wire the corresponding **Reference** connectors and **Error In/Out** connectors.

9. Add a path constant and set it as **interpolate.ogw**, and wire it to **Path** connector of **OA_Load**.

10. Add a numeric constant and set it to 2, and wire it to the **Column Name/Index** connector of **OA_GetColumn** VI.

11. Add an array constant and set it as you want, and wire it to the **data** connector of **OA_Col-SetData**.

12. When the wiring is complete, your Block Diagram window should look like this:



13. Click the **Run** button to send the data to Origin. You will see the interpolating results appear in Column D automatically.

3.5 Getting Data from Worksheet Range

This tutorial shows how to get data to Labview from a rectangle range of a worksheet in Origin.

1. Run Origin. If Origin is already running, start a new project. Make sure that there is a Workbook with the default "Book1" and "Sheet1".

2. Select **File: Import: Single ASCII** from the menu, then find **Gaussian.dat** from **<Origin installation folder>\Samples\Curve Fitting**. And make sure the **Show Options Dilalog** checkbox has been selected. Unwrap the **(Re)Naming Worksheet and Workbook** branch, disable the **Rename Sheet with(Partial) Filename** and the **Rename Book with(Partial) Filename** checkboxes. Then click **OK** button to import the file.

3. Run LabView and new a blank VI.

4. Put four integer controls to specify r1, r2, c1, and c2, which mean start row index, end row index, start column index, and end column index, respectively.

5. Put a 2D array with double indicators to show the data get from Origin.

6. Turn to **Block Diagram** window, and choose **OA_ConnectToOrigin** VI and place it to the window.

7. Select and place **OA_FindWorksheet** VI , and then wire the **Origin.IOApplication** and **Error IO** connectors to the two VIs.

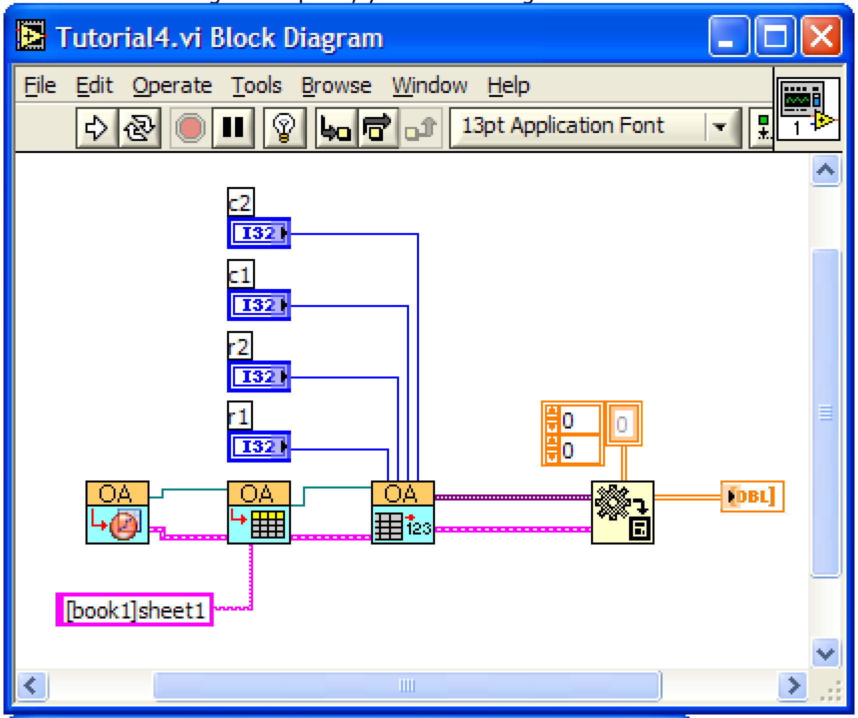
8. Add a string constant and assign to it the string **[Book1]Sheet1**, and then wire it to the **Name** connector of **OA_FindWorksheet** VI.

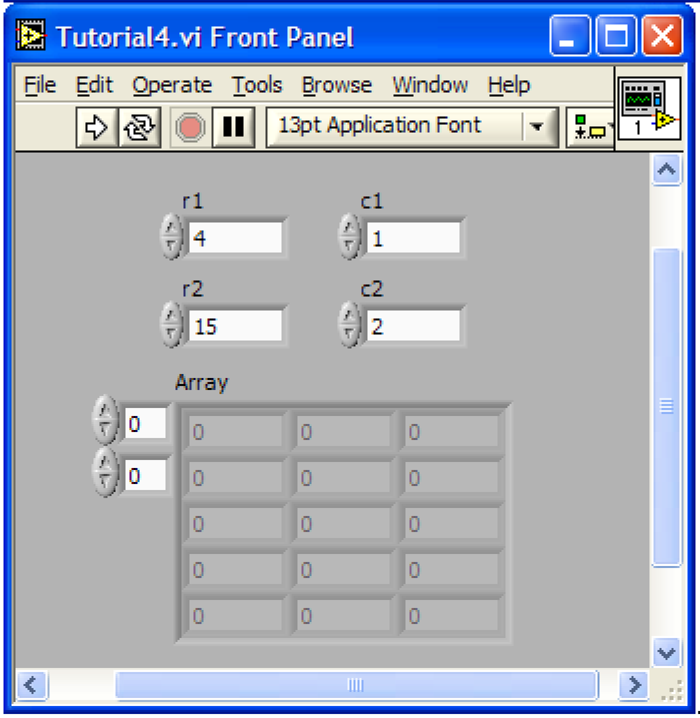
9. Select and place **OA_Wks-GetRange** VI , and then wire the **Origin.Worksheet** and **Error IO** connectors to the corresponding connectors of the **OA_FindWorksheet** VI.

10. Wire the four controls r1, r2, c1, and c2 to corresponding connectors of the **OA_Wks-GetRange** VI.

11. Since **OA_Wks-GetRange** VI get the data as Variant type, we use a **Variant to Data** SubVI to convert it to fit the output 2D Array, which is under **Advanced\Data Manipulation\Variant** palette.

12. When the wiring is complete, your Block Diagram and Front window should look like this:





r1 4

c1 1

r2 15

c2 2

Array

0

0

0	0	0
0	0	0
0	0	0
0	0	0
0	0	0

13. Change the r1, r2, c1, c2 and then click **Run** button to get a range of worksheet from Origin to Labview.

3.6 Setting and Getting Matrix Data

In this tutorial, we will show you how to operate on Origin matrix in Labview. We first create two matrixes in Origin and set value to one of the matrix. Then transfer the transposed matrix data to another matrix.

Note that the subVis related to matrix access are in OriginMatrix_LV7.llb. If you have followed the instruction, you can simply get them through **All Functions/ User Libraries/ OriginMatrix** from **Functions** palette.

1. Run Labview, create a new VI and active the Block Diagram window. Then choose **OA_ConnectToOrigin** VI and place it to the window.

2. Select and place **OA_NewMatrixsheet** VI , then wire the **Origin.IOApplication** and **Error IO** connectors to the corresponding connectors of the **OA_ConnectToOrigin** VI.

3. Select **All Functions/ Application Control** from **Functions** palette and place **Invoke Node** VI on the diagram. Then wire **reference** connector to **Origin.MatrixSheet** connector of **OA_NewMatrixsheet** VI. And wire **Error IO** connector to the corresponding connector of the **OA_NewMatrixsheet** VI too. After wired the connectors, this VI will be like , click on **Method** and choose **Execute** from the fly-out menu. Then add a string constant and set it to **matrix -v 10*i + j**, and then wire it to the **LabTalkStr** connector.

4. Select and place another **OA_NewMatrixsheet** VI , then wire the **Origin.IOApplication** to the corresponding connector of the first **OA_NewMatrixsheet** VI. And wire the **Error IO** to the corresponding connector of the **Invoke Node** VI.

5. Select and place **OA_GetMatrix** VI , then wire the **Origin.MatrixSheet** connector to the corresponding connector of the **Invoke Node** VI. And wire the **Error IO** to the corresponding connector of the second **OA_NewMatrixsheet** VI.

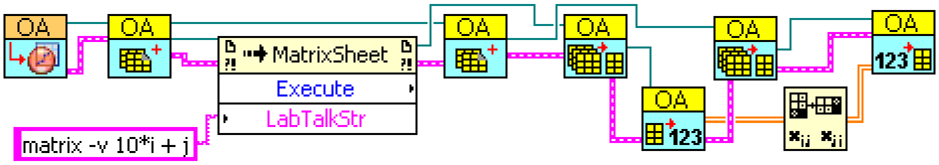
6. Select and place **OA_Mat-GetData** VI , and then wire the **Origin.Matrix** and **Error IO** connectors to the corresponding connectors of the **OA_GetMatrix** VI.

7. Select and place **OA_GetMatrix** VI , then wire the **Origin.MatrixSheet** to the corresponding connector of the second **OA_NewMatrixsheet** VI. And wire the **Error IO** to the corresponding connector of the **OA_Mat-GetData** VI.

8. Select and place **OA_Mat-SetData** VI , and then wire the **Origin.Matrix** and **Error IO** connectors to the corresponding connectors of the **OA_GetMatrix** VI.

9. Select **All Function/ Array** from **Functions** palette and place **Transpose 2D Array** VI , then wire the **2D array** and **transposed array** connectors to the **Data** connectors of **OA_Mat-GetData** VI and **OA_Mat-SetData** VI separately.

10. When all of these are finished, the Block Diagram should look like:



11. Click **Run** button to run this VI and then you can turn to Origin to see the result.

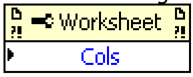
3.7 Working with Origin Com And LabVIEW SubVIs

Origin can function as an automation server whereby other applications communicate with Origin using methods and properties exposed by Origin. LabVIEW can support COM programming, so it can easily communicate with Origin using Origin COM. Origin has equipped with some LabVIEW SubVIs, which can do some basic operations on worksheet, matrix, etc. But they are not enough when you are facing some complicated problems. In these situations, you may need to use Origin Com and the existed LabVIEW SubVIs together. This tutorial will show you how to work with Origin Com and LabVIEW SubVIs.

1. Run Labview, create a new VI and active the Block Diagram window. Then choose **OA_ConnectToOrigin** VI and place it to the window.

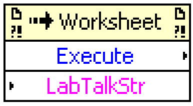
2. Select and place **OA_NewWorksheet** VI , then wire the **Origin.IOApplication** and **Error IO** connectors to the corresponding connectors of the **OA_ConnectToOrigin** VI.

3.Move the Mouse Cursor to the **Origin.Worksheet** connector of the **OA_NewWorksheet** VI. When the shape of the Cursor become to , right click and choose **Create/ Property/ Cols** to create a **Property Node**. Then right click on it and choose **Change All To Write**. After that, the **Property Node** will be like



. Then wire **reference** connector to **Origin.Worksheet** connector of **OA_NewWorksheet** VI. And wire **Error IO** connector to the corresponding connector of the **OA_NewWorksheet** VI too.

4. Move the mouse to the **Origin.Worksheet** connector of the **Property Node** . When the shape of the mouse cursor is like , right click and choose **Create/ Method/ Execute** to create another **Property Node**. The **Property Node** will be like

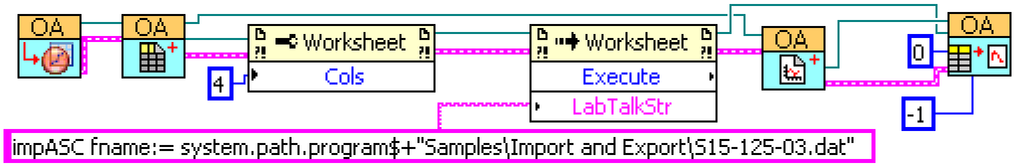


. Then wire **reference** connector to **Origin.Worksheet** connector of previous **Property Node**. And wire **Error IO** connector to the corresponding connector too. Then add a string constant and set it to **impASC fname:= system.path.program$+"Samples\Import and Export\S15-125-03.dat"**, and then wire it to the **LabTalkStr** connector.

5. Select and place **OA_NewEmptyGraph** VI , then wire the **Origin.IOApplication** to the corresponding connector of the **OA_ConnectToOrigin** VI. And wire the **Error IO** to the corresponding connector of the second **Property Node** VI.

6. Select and place **OA_PlotWksCols** VI , then wire the **Origin.Worksheet in** connector to the corresponding connector of the Second **Property Node**. Wire the **Origin.GraphLayer in** and **Error IO** connectors to the corresponding connector of **OA_NewEmptyGraph** VI. Then create two integer constant and set them to **0**, **-1**. Connect **0**,**-1** to **c1**,**c2** connectors separately.

7. When all of these are finished, the Block Diagram should look like:


OA
OA
4
Worksheet
Cols
Worksheet
Execute
LabTalkStr
OA
0
OA
-1
impASC fname:= system.path.program$+"Samples\Import and Export\S15-125-03.dat"

8. Click **Run** button to run this VI and then you can turn to Origin to see the result.

3.8 Using LabTalk to Get Worksheet Information

LabTalk is a high-level, full-featured programming language which has access to most of Origin's operation. So you can take the advantage of LabTalk to use it in LabVIEW to facilitate your work. This tutorial will show you how to run LabTalk scripts in LabVIEW to return the maximum numbers of the column and row that have data.

1. Run Origin. If Origin is already running, start a new project. Make sure that there is a Workbook with the name **Book1** and a worksheet with the name **Sheet1**.

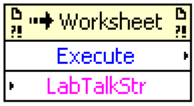
2. Add several columns in the worksheet, for example add 3 columns. Select a range of cells and right-click on it to select **Fill Range With: Row Number**.

3. Run LabVIEW and create a blank VI.

4. Turn to **Block Diagram** window, and choose **OA_ConnectToOrigin** VI and place it to the window.

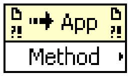
5. Select and place **OA_FindWorksheet** VI , and then wire the **Origin.IOApplication** and **Error IO** connectors to the **OA_ConnectToOrigin** VI.

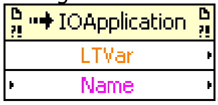
6. Add a string constant and assign to it the string **[Book1]Sheet1**, and then wire it to the **Name** connector of **OA_FindWorksheet** VI.

7. Move the mouse to the **Origin.Worksheet** connector of the **OA_FindWorksheet.vi**. When the shape of the mouse cursor looks like , right click on and choose **Create/ Method/ Execute** to create a **Property Node**. The **Property Node** will look like 

. Then wire **reference** connector to **Origin.Worksheet** connector. And wire **Error IO** connector to the corresponding connector too. Then add a string constant and input the following scripts into it. And then wire it to the **LabTalkStr** connector.

```
maxRows =wks.maxrows;
j=wks.ncols;
for(i=wks.ncols;i>0;i--)
{
  get wcol(i) -e v1;
  if(v1>0) break;
  j=i-1;
};
maxCols = j;
```

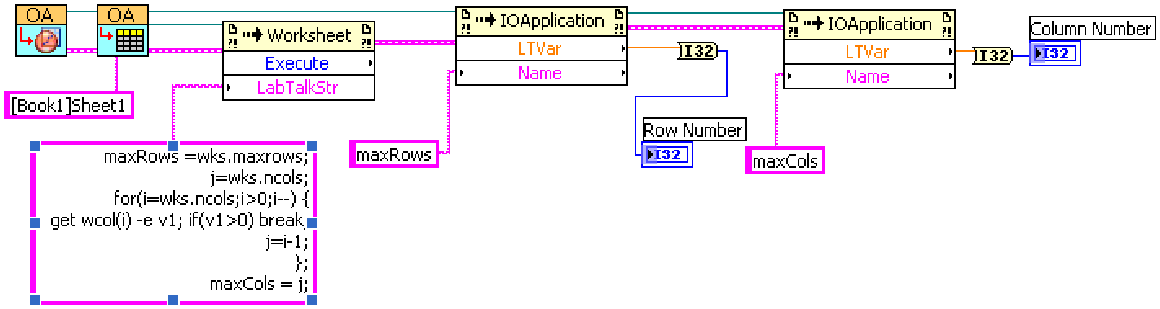
8. Then we will try to get the values of two variables: **maxRows** and **maxCols**. Select **All Functions/ Application Control** from **Functions** palette and place **Invoke Node** VI 

on the diagram. Then wire **reference** connector to **Origin.IOApplication** connector of **OA_FindWorksheet** VI. And wire **Error IO** connector to the corresponding connector of the previous **Invoke Node** VI too. Right click on **Method** and choose **Methods/LTVar(get)** from the fly-out menu. Then add a string constant and set it to **maxRows**, and then wire it to the **LabTalkStr** connector. This VI should look like 

.

9. Select **All Function/Numeric/ Conversion/ To Long Integer** to add the **To Long Integer** vi I32 into the diagram. Wire its **Number** connector to the **LVar** connector of the **InvodeNode**. And remember to create an Indicate to its another connector.

10. Then repeat the steps 8~9 to get the maximum number of columns (Please remember to change the string value to **maxCols**.). The Block Diagram should look like:



```
maxRows =wks.maxrows;
j=wks.ncols;
for(i=wks.ncols;i>0;i--) {
get wcol(i) -e v1; if(v1>0) break;
j=i-1;
};
maxCols = j;
```

11. Click **Run** button to run this VI and then you can turn to Front Panel to see the results.

4 LabVIEW Examples

Origin ships with a collection of examples to show LabVIEW users how to create VIs to communicate between LabVIEW and Origin

The examples provided in the following Origin installation folder:

```
<exe>\Samples\COM Server and Client\LabVIEW\
```

Notes: These examples require Origin 8 SR4 or later builds.

Examples included in this section:

- Origin Column Set Get Data
- Batch Linear Regression
- Simple NLFit
- On The Fly Analysis
- Simulate DAQ
- 6009 Simple LV71
- Realtime Curve Fitting
- 6009 LV71
- Plotting Data
- Plotting XY Data with the Same X
- Sending XY Data To Origin
- Send and Plot XYYY Data
- Simulated Peak
- Matrix Access
- Sending Waveforms to Origin
- Analysis Template
- Waveform Envelope

4.1 Origin Column Set Get Data

This example shows how to access Origin Worksheet Columns. The VI demonstrates the following:

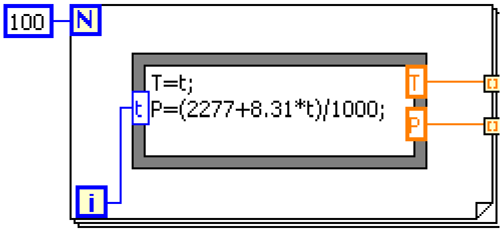
1. Connecting to a running Origin with OA_ConnectToOrigin.vi which maps to the ApplicationSI method in Origin.
2. Loading an Origin OPJ file under Origin's program folder.
3. Configuration of worksheet column objects, including setting them as even sampling (no separate X columns are needed).

4. Sending data to columns in a sheet named "Data" and getting data from another sheet named "Basic Stats".
5. Trigger recalculation in Origin before getting data so as to show how Origin can be used as an Analysis Template.

Note that the VI using an .opj file named **Basic Stats on Data** is shipped with Origin, which contains an Analysis Template to do simple basic statistics computation.

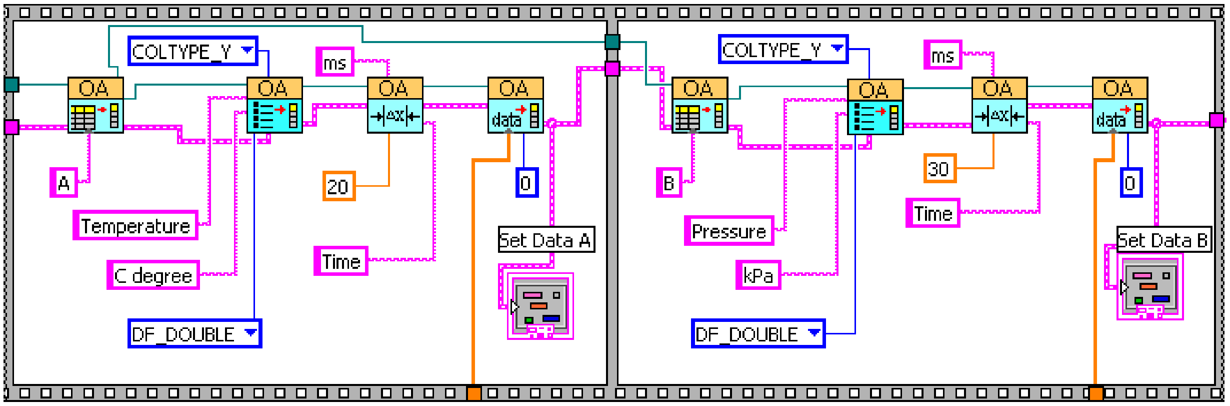
Some important blocks are as below showing:

This block is for generating some data to put into the two columns in our sample:

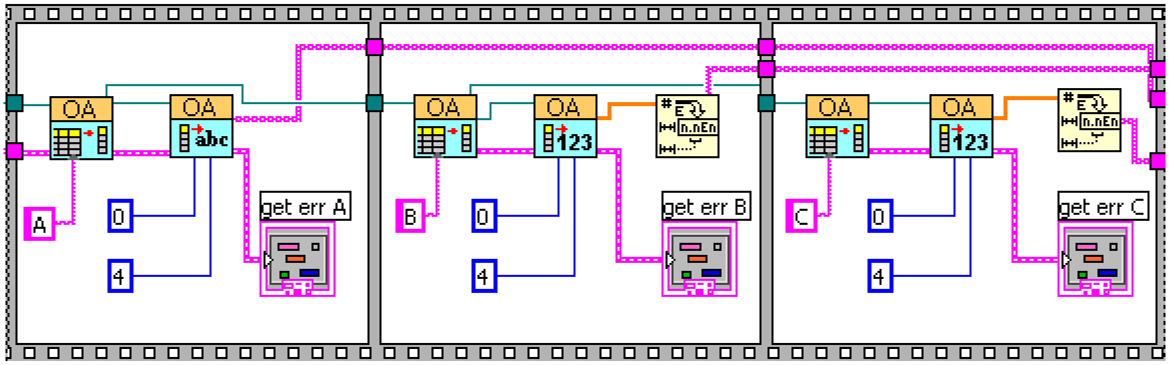


```
T=t;
P=(2277+8.31*t)/1000;
```

The blocks below are for setting up column labels and sending data to 2 columns in Origin:



The blocks below are for reading data from specified columns of result worksheet:



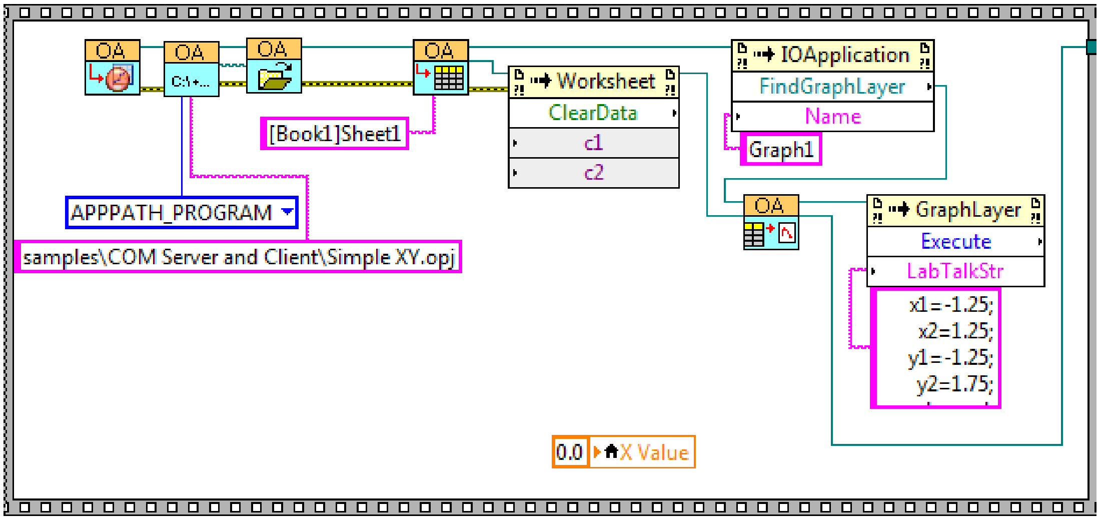
4.2 Plotting XY Data with the Same X

This example shows how to make a XY data plot with the same X in LabVIEW with Origin.

1. It is to demonstrate the following:

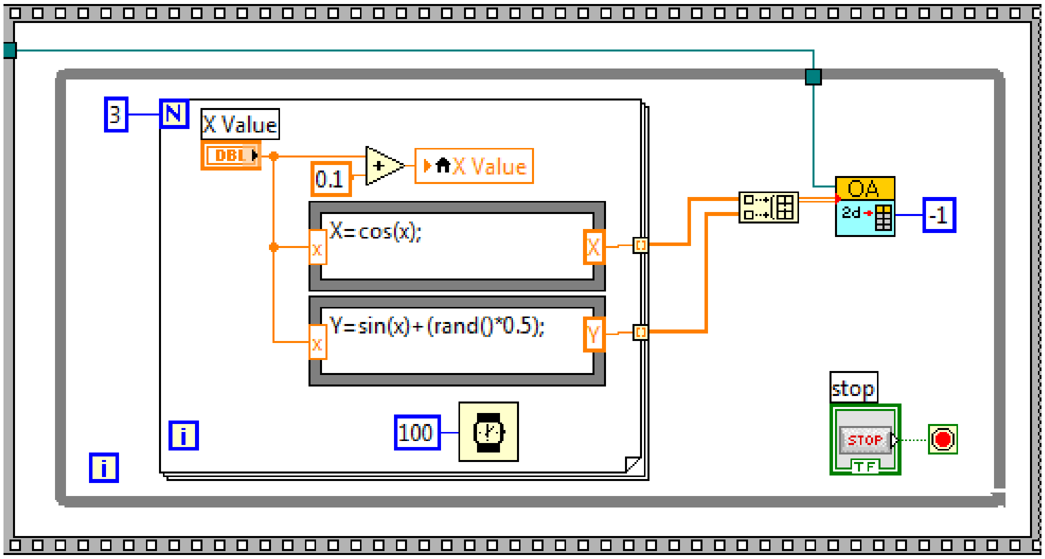
2. Calculate the expressions *X(x)* and *Y(x)* with the same variable *x*, *x* is increasing continuously when the program running, until the Stop button pressed down.
3. Make a 2D line plot by using X and Y dynamically on a graph layer.

- The block diagram below is starting from an Origin project file named Simple XY.opj. The axis scale of the Graph in project was set by LabTalk script.



```
x1=-1.25;
x2=1.25;
y1=-1.25;
y2=1.75;
```

- Parametric function *X(x)* and *Y(x)* was generated, and the interval of parameter x is 0.1. The resulting values was sended to Worksheet and GraphLayer with a data update period of 100ms.



```
X=cos(x);
```

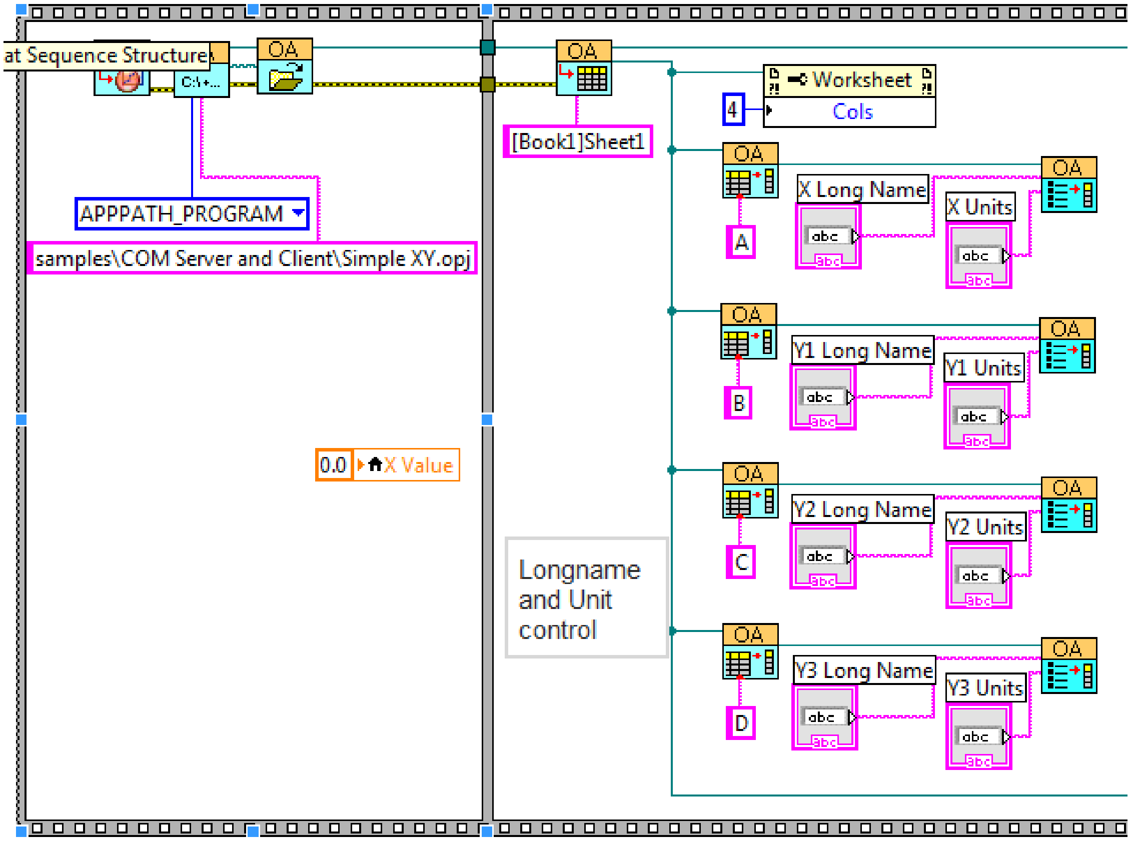
```
Y=sin(x)+(rand()*0.5);
```

4.3 Send and Plot XYYY Data

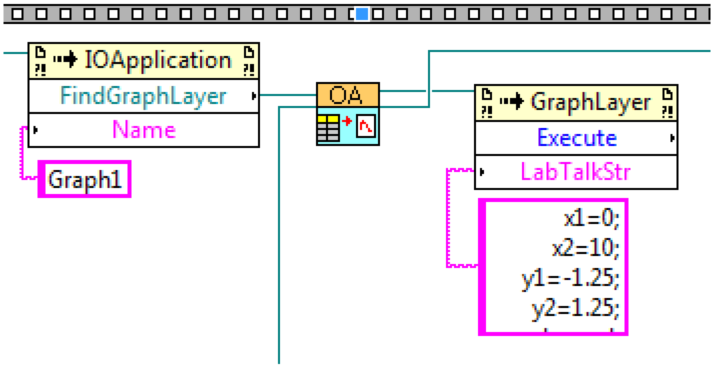
This example shows how to generate sine and square signal and make a 2D line plot in LabVIEW with Origin. It is to demonstrate the following:

1. Generate sine and square signal with the same sequence x, x is increasing continuously when program running, until the Stop button pressed down.
2. Make a 2D line plot for signal waveform dynamically on a graph layer. In addition, you can specify the Long name and Unit for the line plot.

- The block diagram below is starting from an Origin project file named Simple XY.opj and add a worksheet with 2 columns designated as Y, the Long Name and Units of columns are controllable.

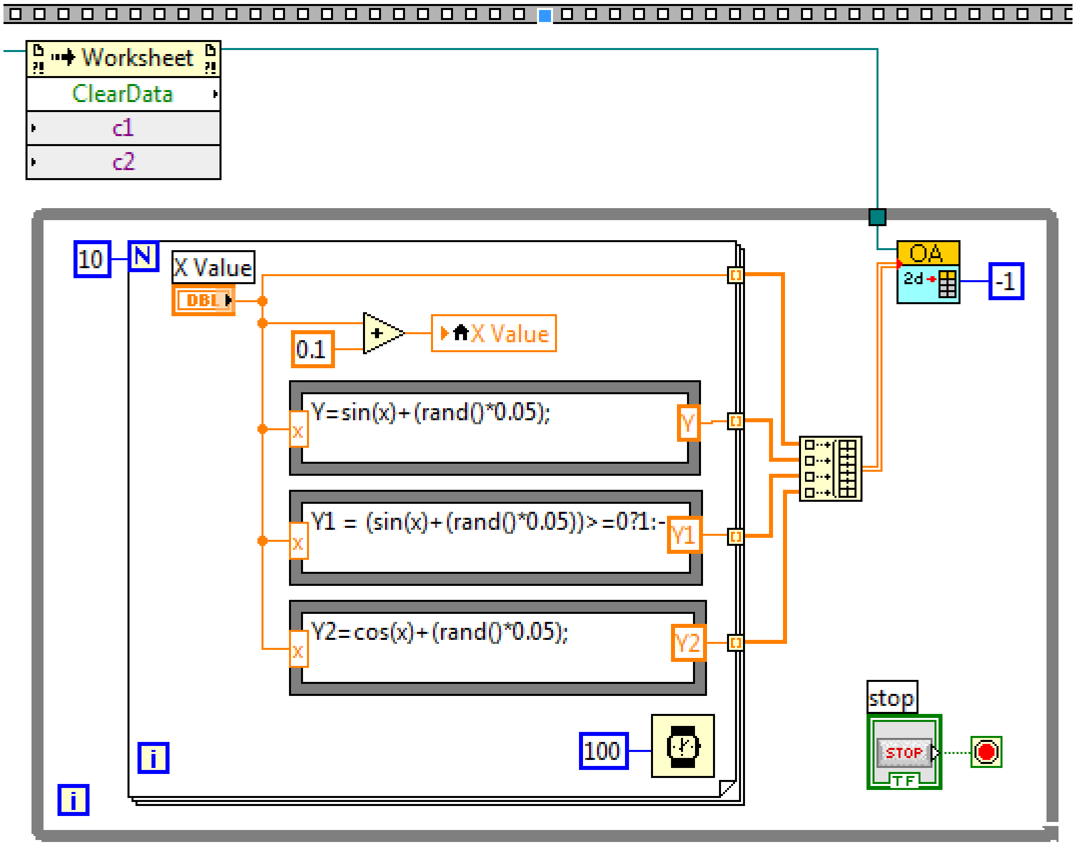


- The block diagram below plot data in worksheet into Graph1 and set the scale for graph via LabTalk script.



```
x1=0;
x2=10;
y1=-1.25;
y2=1.25;
```

- Y, Y1 and Y2 are sinusoidal and square wave generated, the wave data was output to the Worksheet and GraphLayer.



```
Y=sin(x)+(rand()*0.05);
Y1 = (sin(x)+(rand()*0.05))>=0?1:-
Y2=cos(x)+(rand()*0.05);
```

4.4 Batch Linear Regression

This example shows how to use the Batch Processing tool to perform batch linear regression.

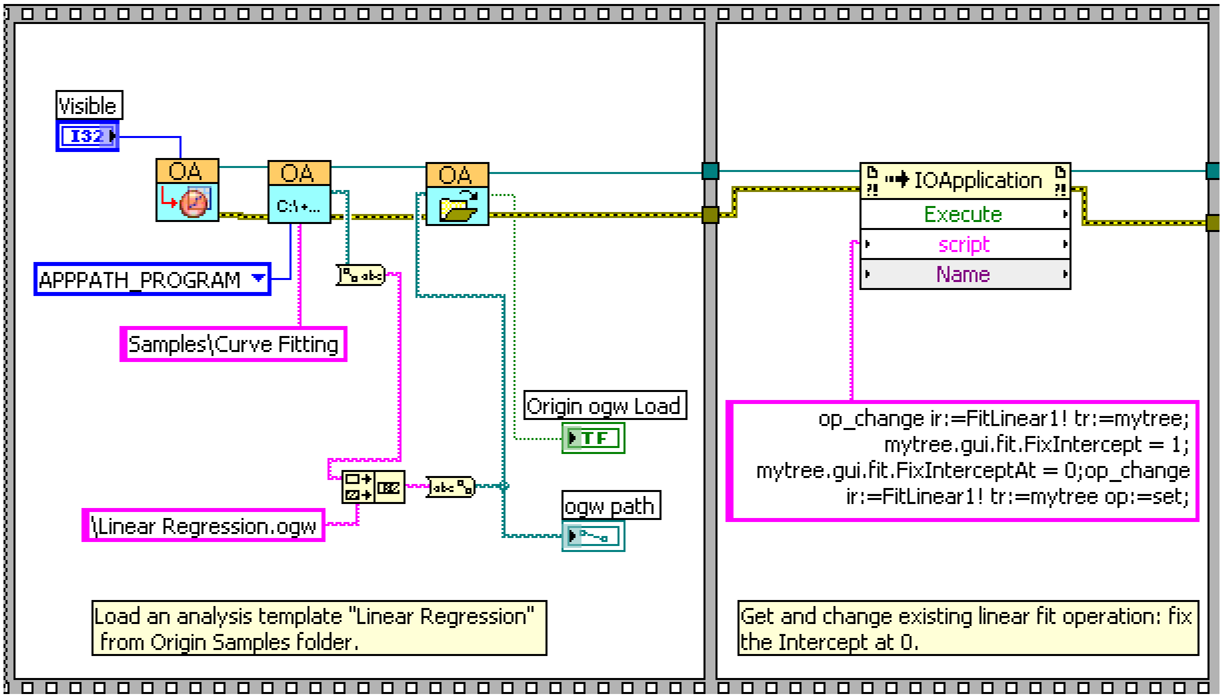
It is to demonstrate the following:

1. Connect to a running Origin with OA_ConnectToOrigin.vi which maps to the ApplicationSI method in Origin.

2. Specify a path to load the analysis template "Linear Regression.ogw", which is under <Origin Program Folder>\Samples\Curve Fitting\ folder.
3. Get and change existing linear regression setting that stored in the analysis template, which intends to fix the Intercept at 0.
4. Prompt a dialog to ask you to select one or multiple data files. To be compatible with the customized analysis template, you should select the "Sensor01.dat", "Sensor02.dat" and "Sensor03.dat" together (or just select one of them) that in the <Origin Program Folder>\Samples\Curve Fitting folder.
5. Perform batch processing and output the results into table.

This example is mainly composed of 3 parts.

- The figure below is to load a template and change the linear regression settings.

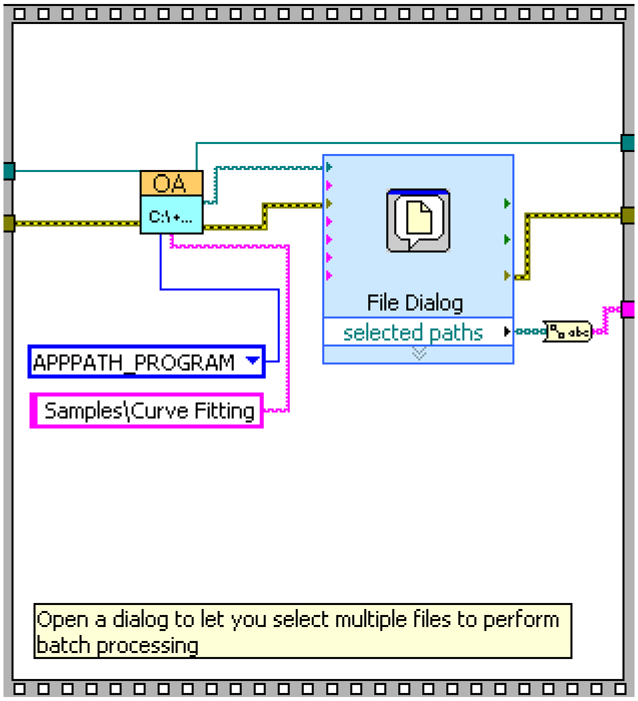


```
op_change ir:=FitLinear1! tr:=mytree;
mytree.gui.fit.FixIntercept = 1;
mytree.gui.fit.FixInterceptAt = 0;op_change
ir:=FitLinear1! tr:=mytree op:=set;
```

Load an analysis template "Linear Regression" from Origin Samples folder.

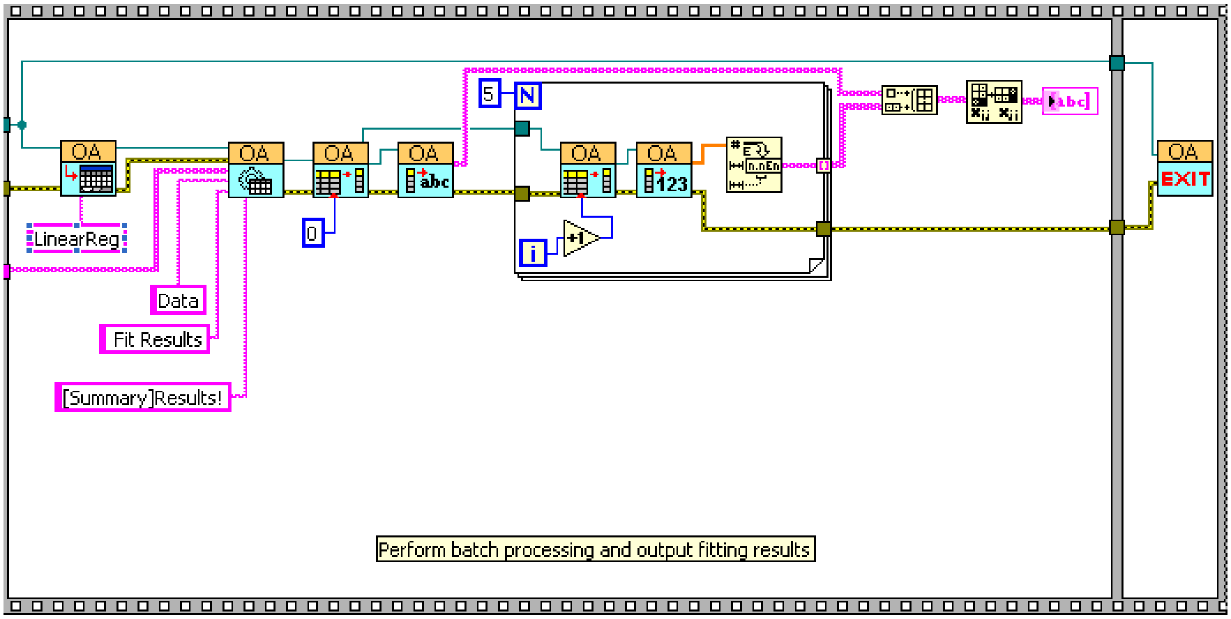
Get and change existing linear fit operation: fix the Intercept at 0.

- The figure below is to let you select multiple files.



Open a dialog to let you select multiple files to perform batch processing

- The figure below is to perform batch linear regression.



Perform batch processing and output fitting results

Note:

This VI is written in LabVIEW 8.5. Please use LabVIEW 8.5 or more newer version of LabVIEW to run it.

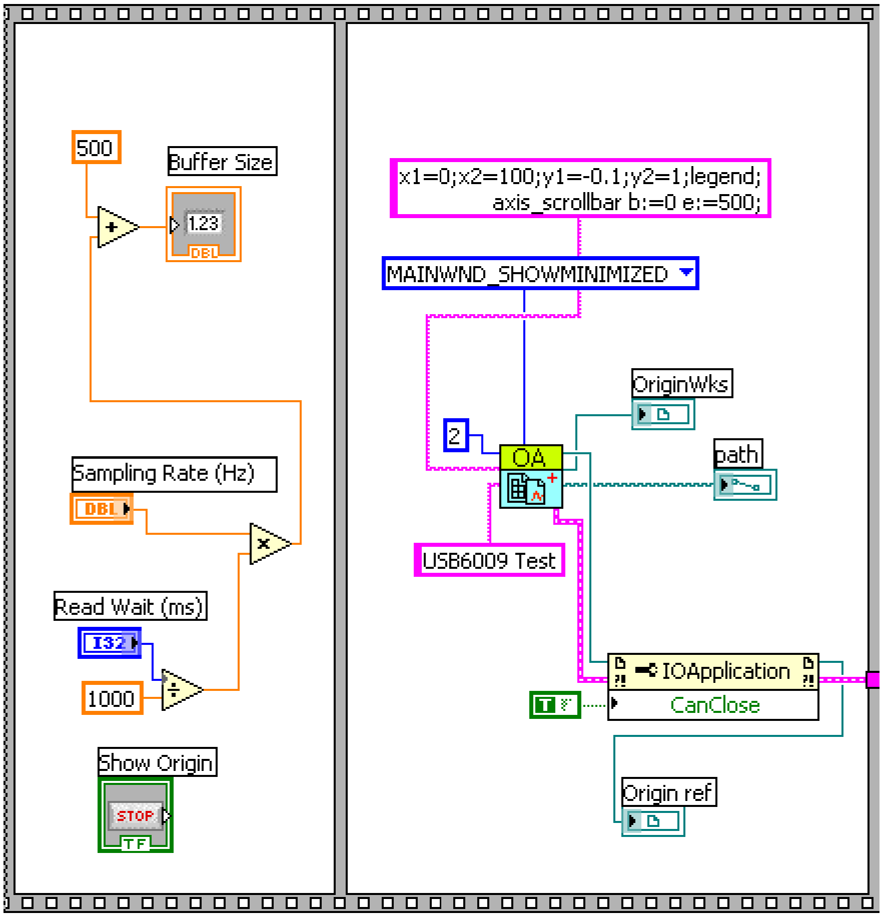
4.5 6009 LV71

This example shows you how to acquire data using National Instruments USB-6009 multifunction I/O devices. It will show how to plot the acquired data dynamically both in Origin and LabView. Compared to 6009 Simple LV71, this example can allow you to set the sampling rate and to change the showing status of Origin 8. And it will also show how the .opj file containing the acquired data will be saved.

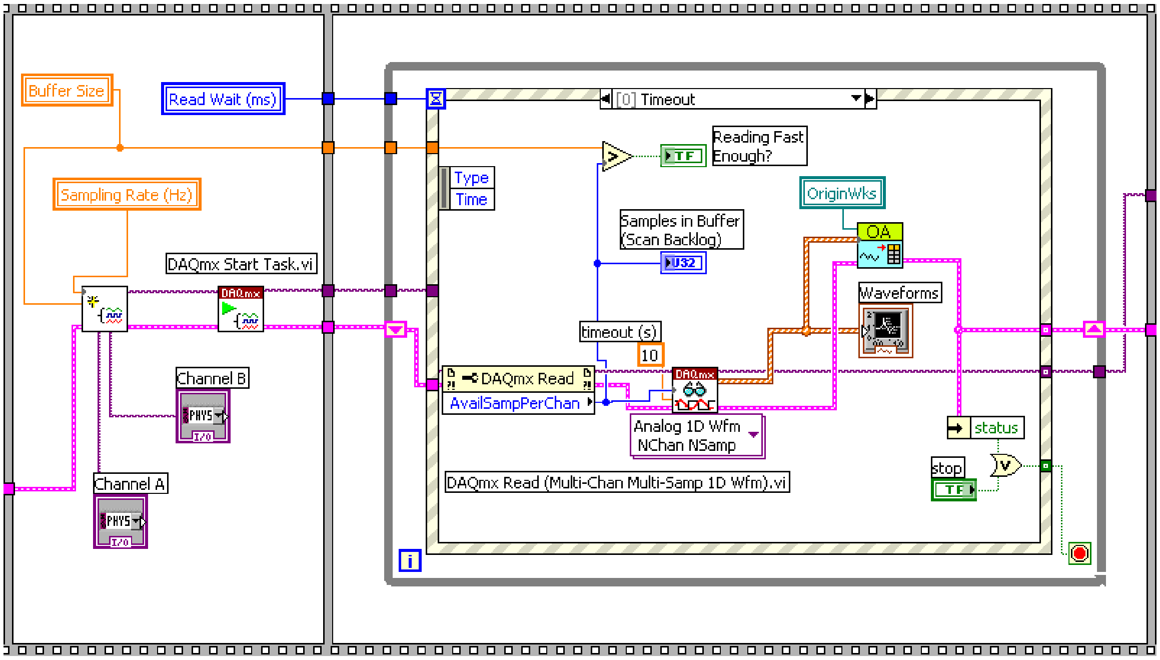
> **Notes:** This VI is created with LabView 7.1.
>
> Make sure that you have installed NA-DAQmx in order to open these DAQ related VIs, and the Labview version you're running is supported by the installed version of NA_DAQmx. e.g. For Labview 2012, it is required to first install NA-DAQmx 9.6.
>
> For detailed information, please refer to the website of National Instruments.

The VI is mainly composed of three parts.

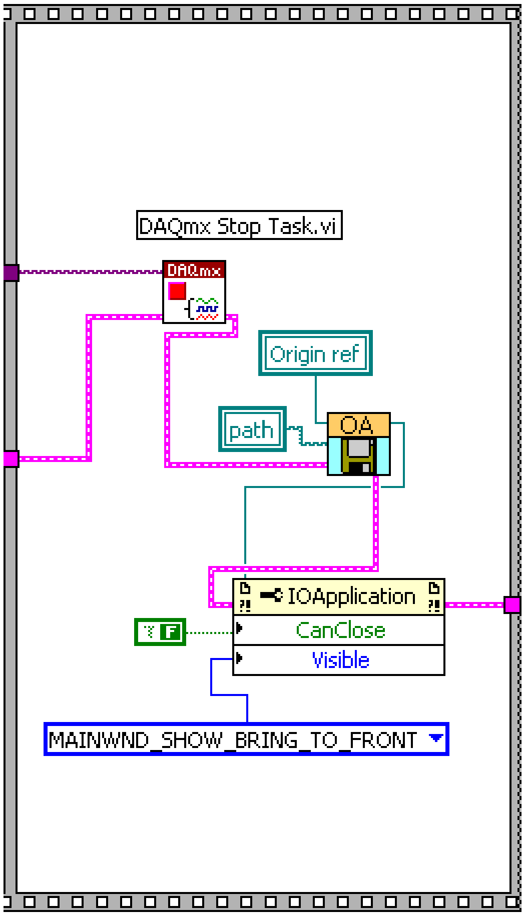
- The part below is to set values of some parameters which are used in data acquisition.



- The figure below is to acquire the data using USB 6009 and plot it in LabView and Origin together. Also the acquired data is saved in an Origin worksheet.



- The figure below is to end the task and save the Origin project.



4.6 6009 Simple LV71

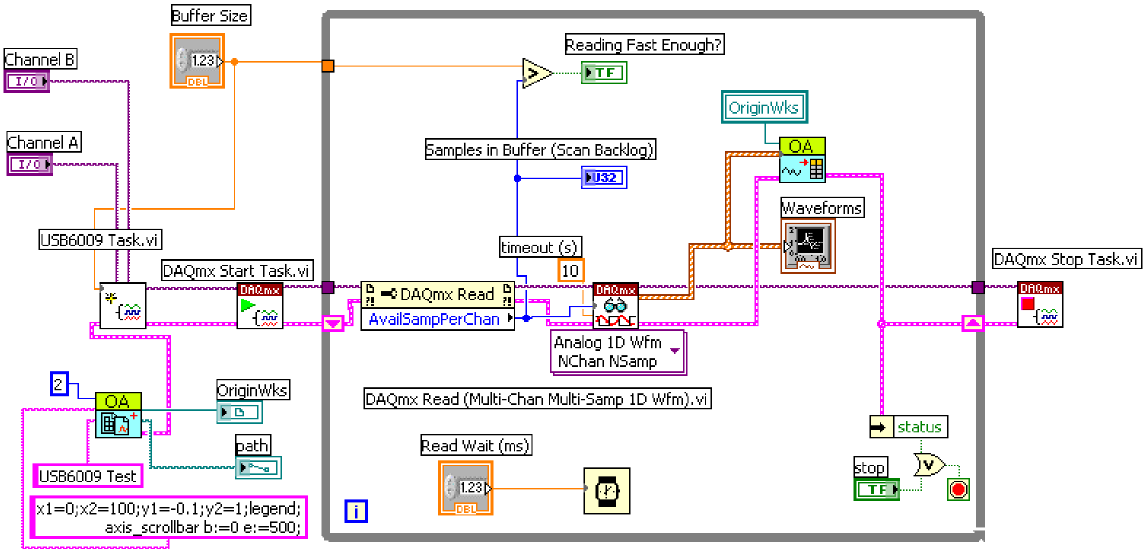
This example shows you how to acquire data using National Instruments USB-6009 multifunction I/O devices. It will show how to plot the acquired data dynamically both in Origin and LabView.

Notes: This VI is created with LabView 7.1.

Make sure that you have installed NA-DAQmx in order to open these DAQ related VIs, and the Labview version you're running is supported by the installed version of NA_DAQmx. e.g. For Labview 2012, it is required to first install NA-DAQmx 9.6.

For detailed information, please refer to the website of National Instruments.

The following is the diagram of this VI.



4.7 On The Fly Analysis

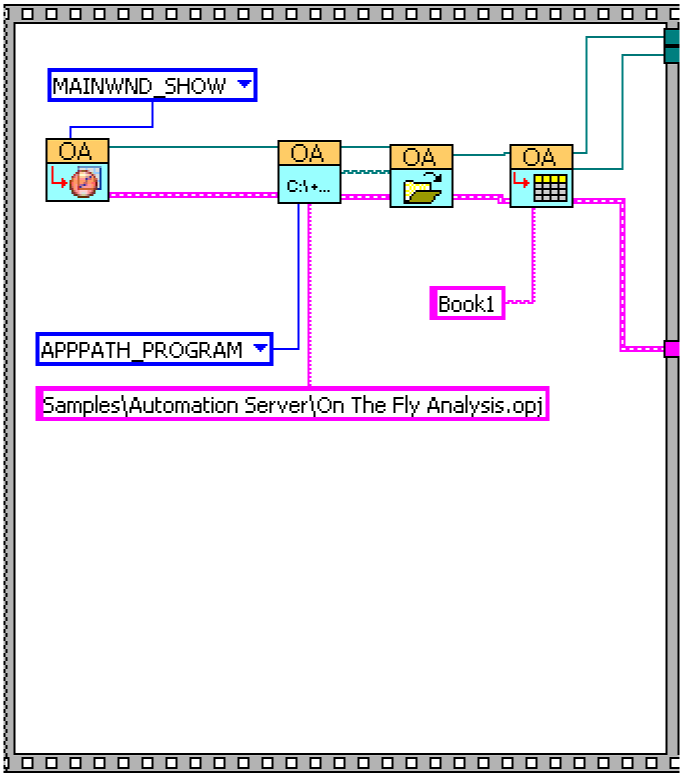
This example shows how to do on the fly analysis in LabView with Origin.

It is to demonstrate the following:

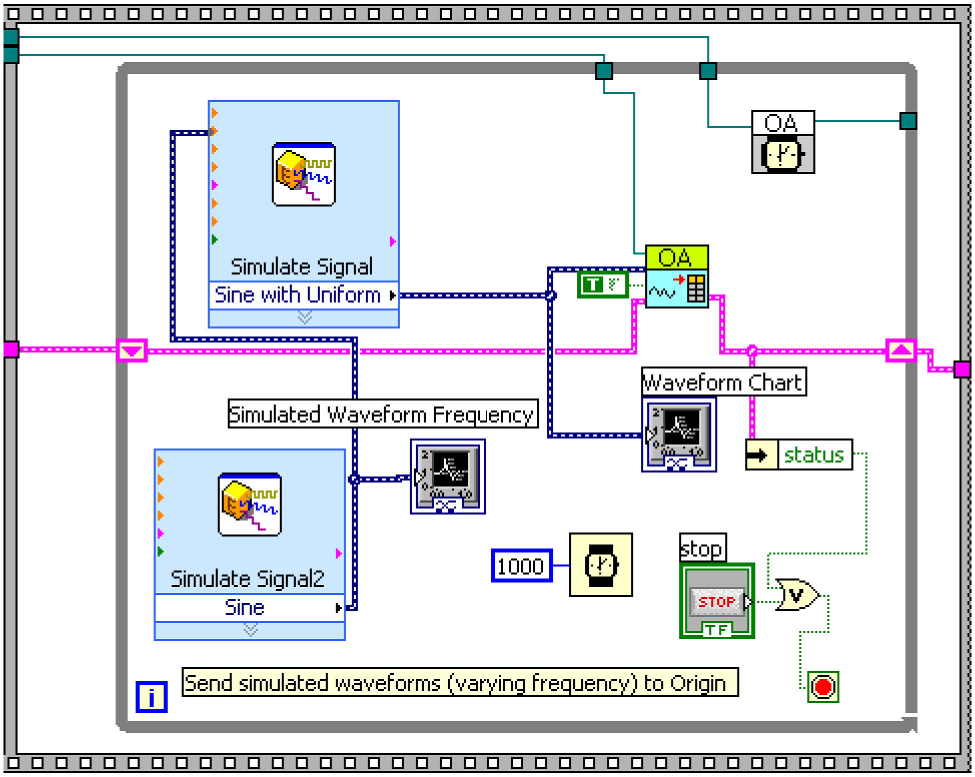
1. Connect to a running Origin with OA_ConnectToOrigin.vi which maps to the ApplicationSI method in Origin.
2. Specify a path to load On The Fly Analysis.opj, which is under <Origin Program Folder>\Samples\COM Server and Client\.
3. Send a simulated data to Origin and trigger Recalculate in Origin to do FFT analysis.

This example is mainly composed of two parts.

- The figure below is to load an .opj.



- The figure below is to perform the on the fly analysis.



4.8 Realtime Curve Fitting

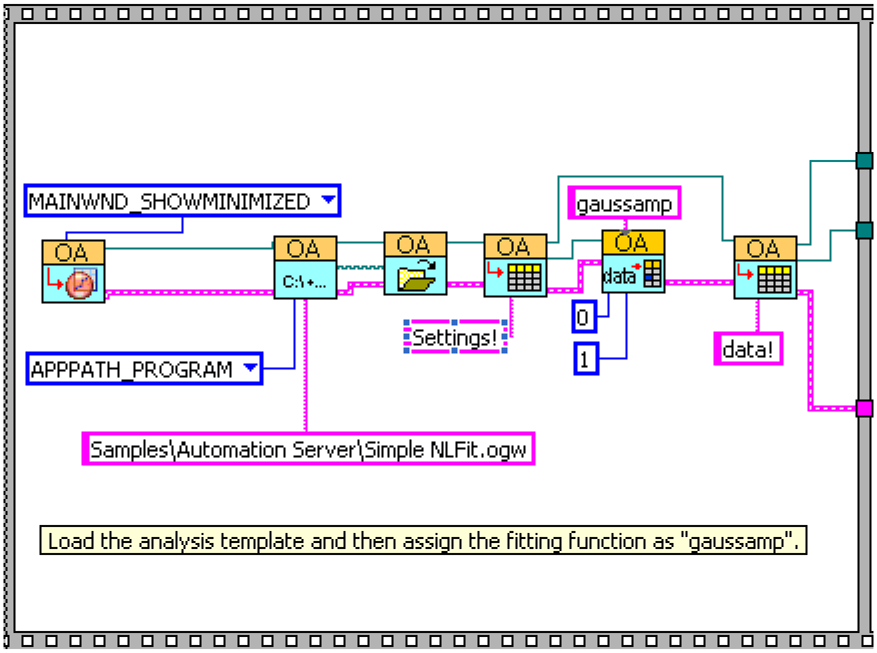
This example shows how to do realtime curve fitting in LabView with Origin.

It is to demonstrate the following:

1. Connect to a running Origin with OA_ConnectToOrigin.vi which maps to the ApplicationSI method in Origin.
2. Specify a path to load the analysis template "Simple NLFit.ogw", which is under <Origin Program Folder>\Samples\COM Server and Client\ folder.
3. Generate the Gaussian peaks with variable widths and import it to analysis tempate to do realtime fitting.
4. Show the dynamic changed widths on the waveform chart.

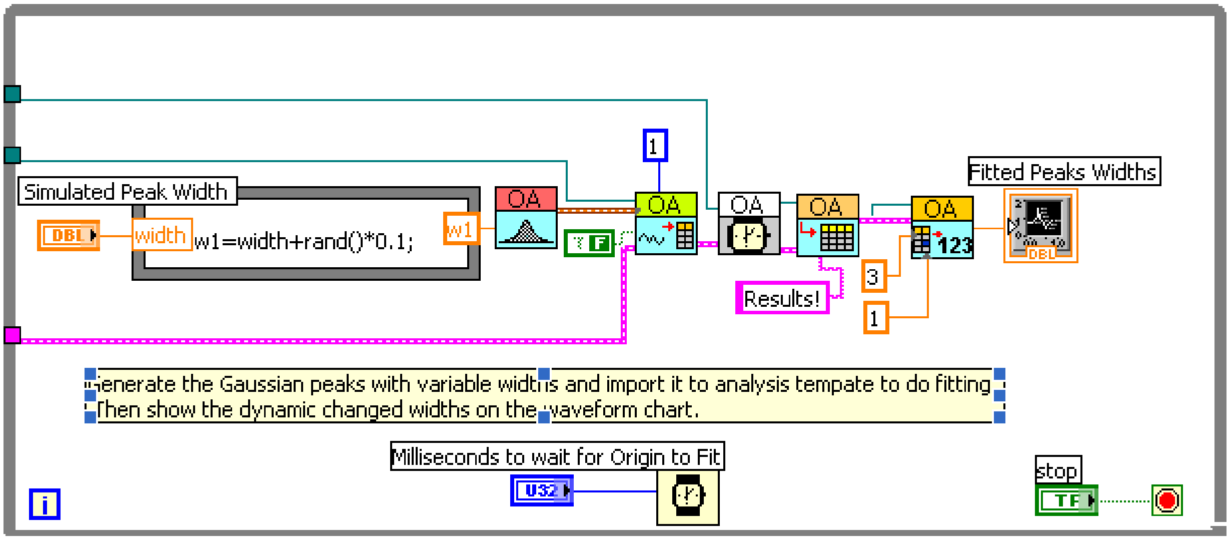
This example is mainly composed of two parts.

- The figure below is to load a template.



Load the analysis template and then assign the fitting function as "gaussamp".

- The figure below is to perform the realtime curve fitting.



```
w1=width+rand()*0.1;
```

Generate the Gaussian peaks with variable widths and import it to analysis tempate to do fitting
Then show the dynamic changed widths on the waveform chart.

4.9 Simple NLFit

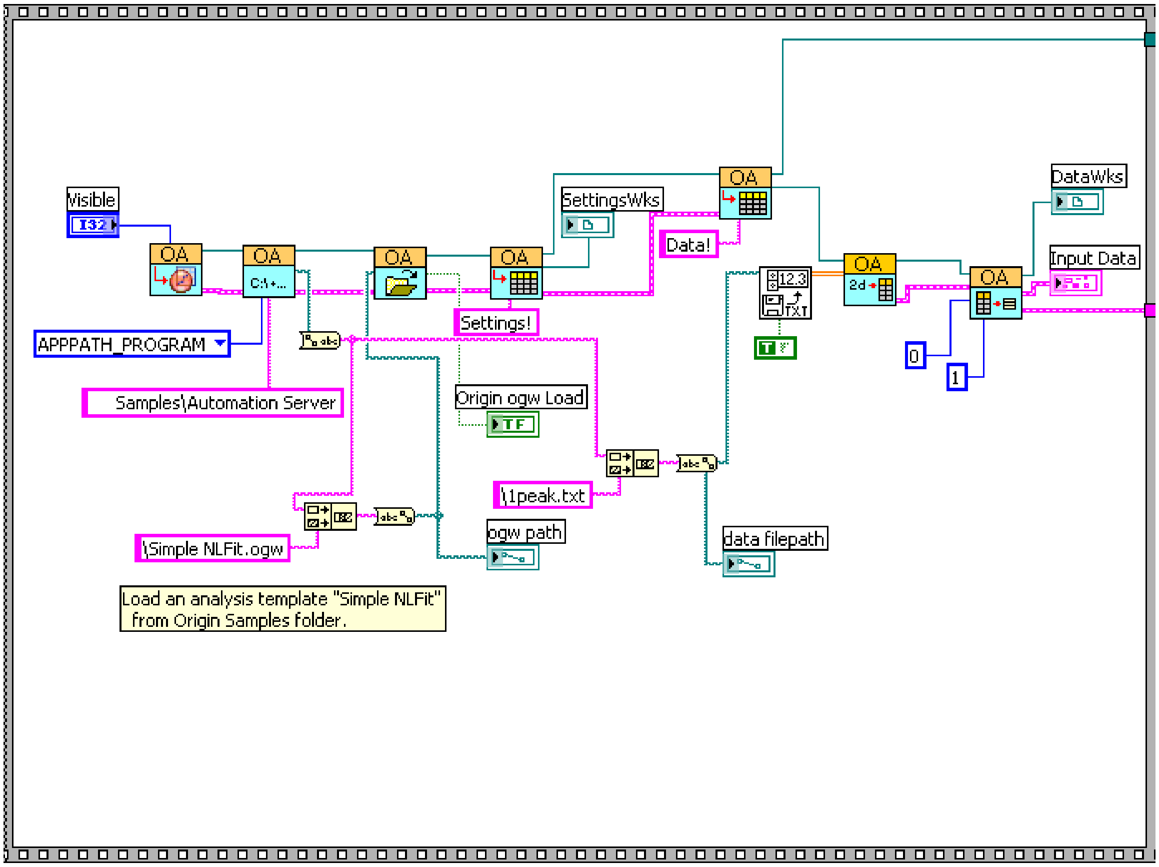
This example shows how to do different kinds of Nonlinear fitting with the same data. And to access the fitting results in Origin.

It is to demonstrate the following:

1. Connect to Origin with OA_ConnectToOrigin.vi which maps to the ApplicationSI method in Origin.
2. Load an Origin Analysis Template (an OGW file) under Origin program folder.
3. Specify a path to load 1peak.txt, which is under <Origin Program Folder>\Samples\COM Server and Client\.
4. Send the data to a sheet named "Data" in the Analysis Template.
5. And then you can select a nonlinear fitting function in the dropdown list.
6. Trigger Recalculate in Origin. The Analysis Template will automatically update the "WaveformEnvelope" sheet.
7. Send the fitting curve and the fitting results from Origin to LabView.
8. Repeat the step 5 and step 7 to do different kinds of fitting until you press the Stop button.
9. Exit the Origin.

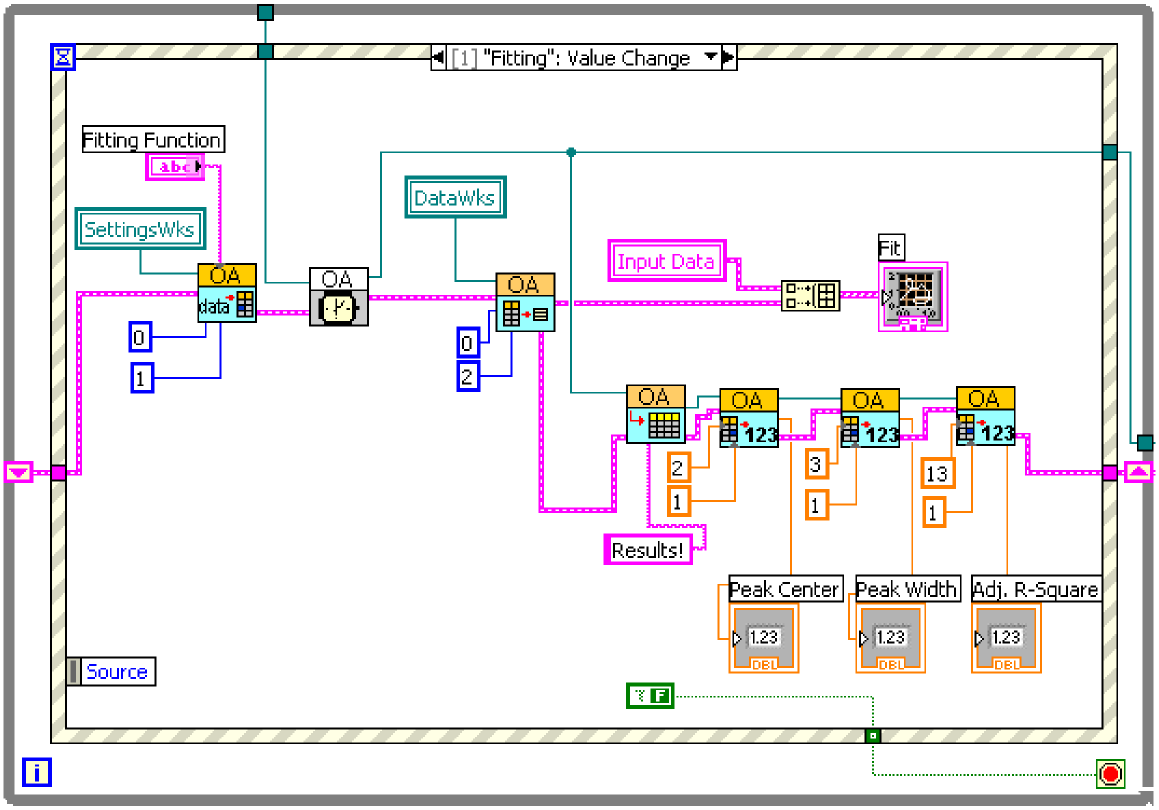
This example is mainly composed of two parts.

- The figure below is to read data from an ASCII file and send the data to the Origin worksheet



Load an analysis template "Simple NLFit" from Origin Samples folder.

- The figure below is to perform the fitting process. And then send the fitting results to LabView.

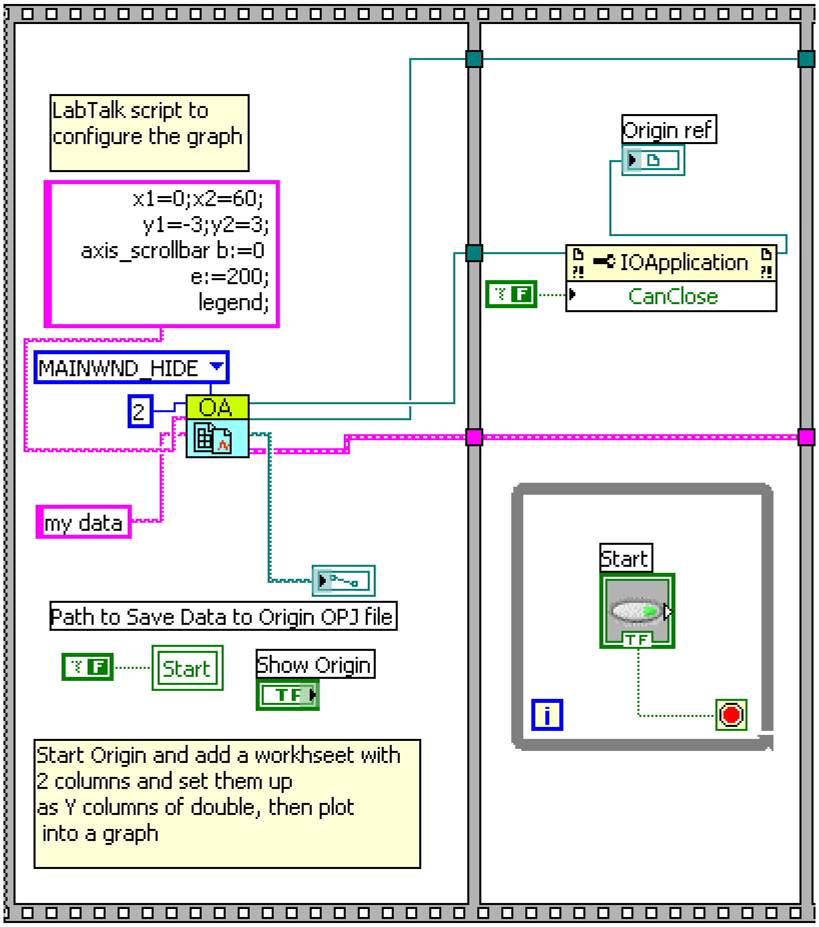


4.10 Simulate DAQ

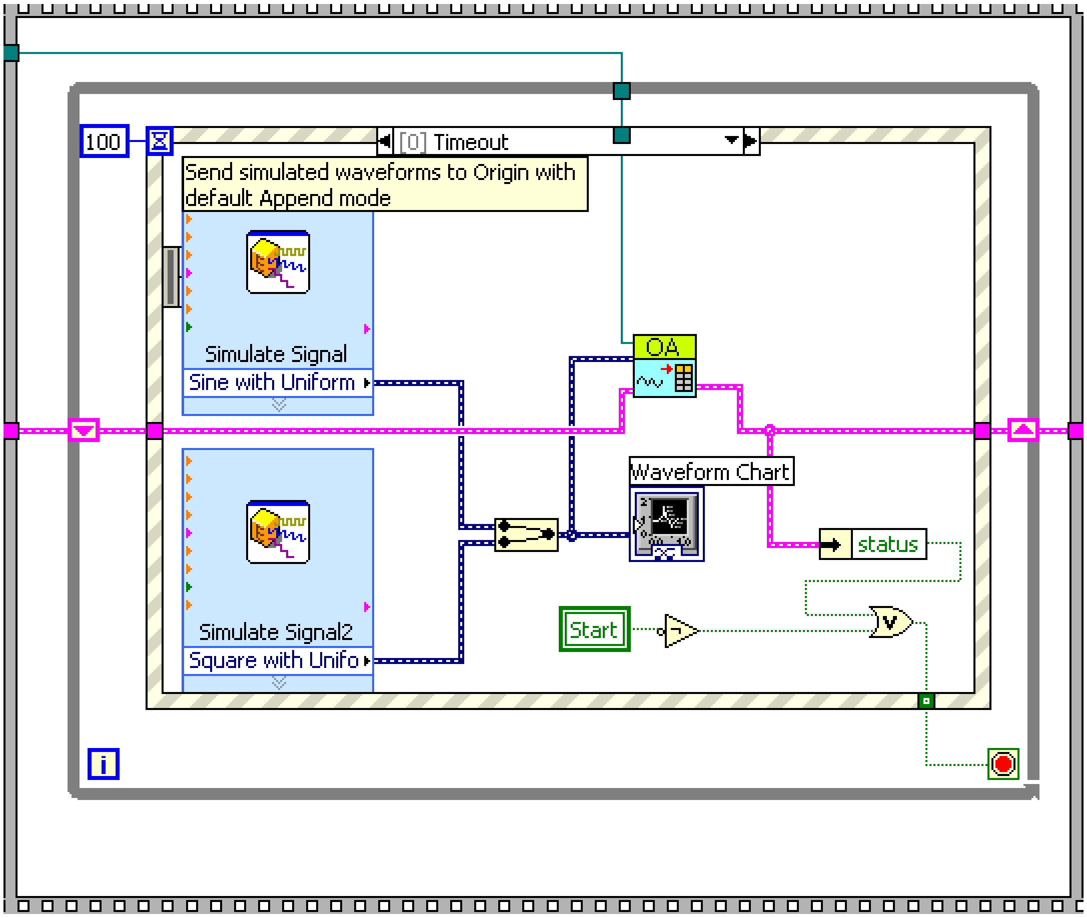
This example shows you how to send the simulated data to Origin worksheet and plot it dynamically both in Origin and LabVIEW.

This example is composed of three parts.

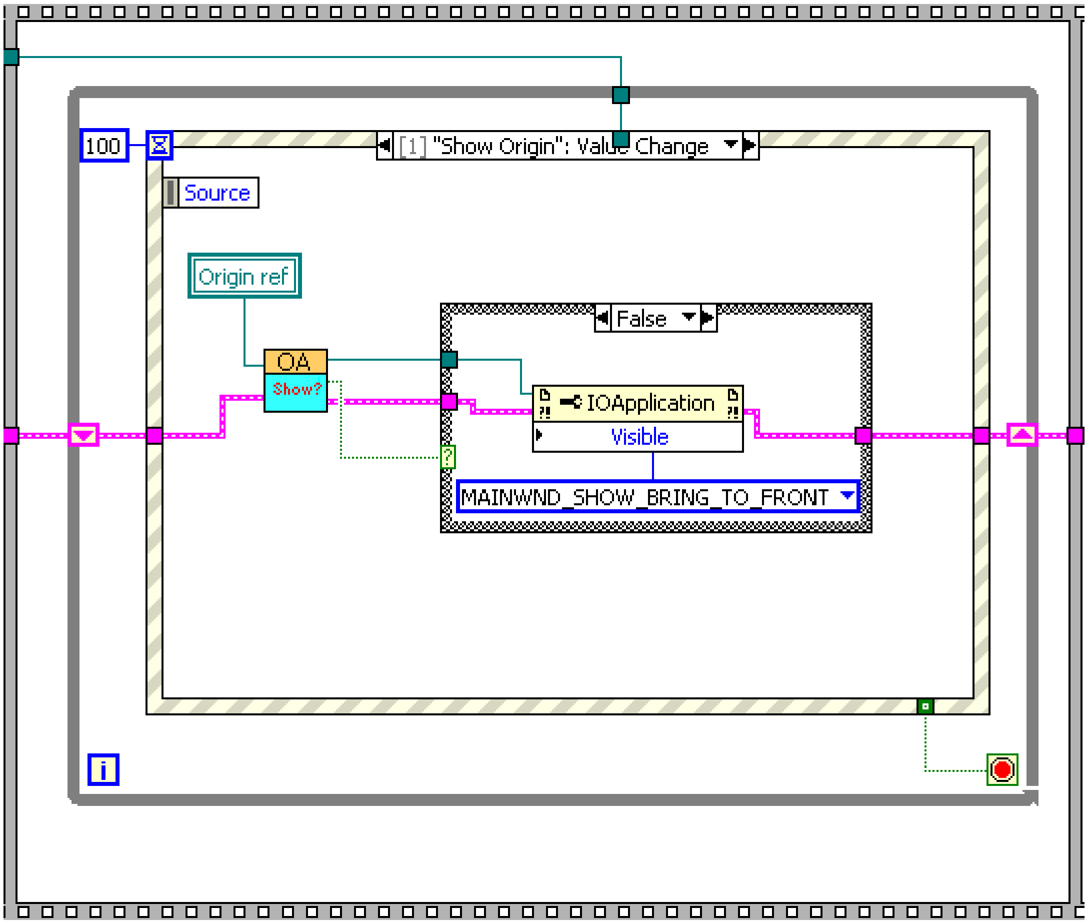
- The following part is to start Origin and add a workshseet with 2 columns. Then set the 2 columns' plot designation as Y and specify the data type of to double and plot them into a graph.


LabTalk script to configure the graph
x1=0;x2=60;
y1=-3;y2=3;
axis_scrollbar b:=0
e:=200;
legend;
MAINWND_HIDE
2
OA
my data
Path to Save Data to Origin OPJ file
F
Start
Show Origin
TF
Start Origin and add a workhseet with 2 columns and set them up as Y columns of double, then plot into a graph
Origin ref
IOApplication
CanClose
F
Start
TF
i

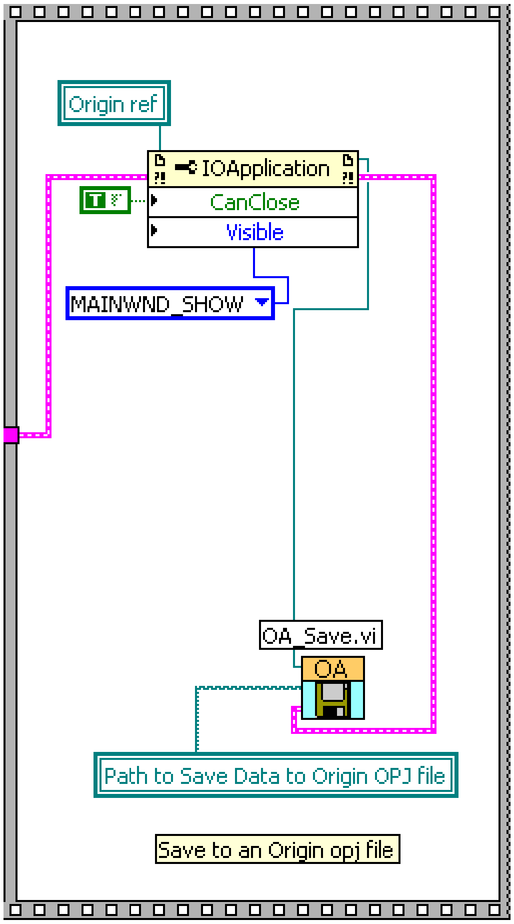
- The following part is an event structure which contains two events.
    - Timeout event: If this event is triggered, the LabView will generate the data and send it to an Origin worksheet and a LabView chart.



- "Show Origin" Value Change event: If this event is triggered, Origin application will be shown. And the simulated data will be show in graph simultaneously.



- The following part is to show final data in Origin and save the data.

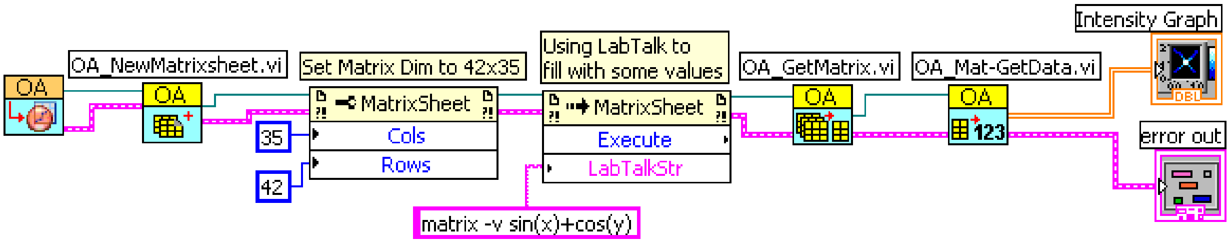

Origin ref
IOApplication
CanClose
Visible
MAINWND_SHOW
OA_Save.vi
OA
Path to Save Data to Origin OPJ file
Save to an Origin opj file

4.11 Matrix Access

This example shows how to operate on a matrix. More specifically, the VI demonstrates the following:

1. Create a new matrix sheet with a single matrix object.
2. Set the properties of the matrix sheet and access it using some subVI.

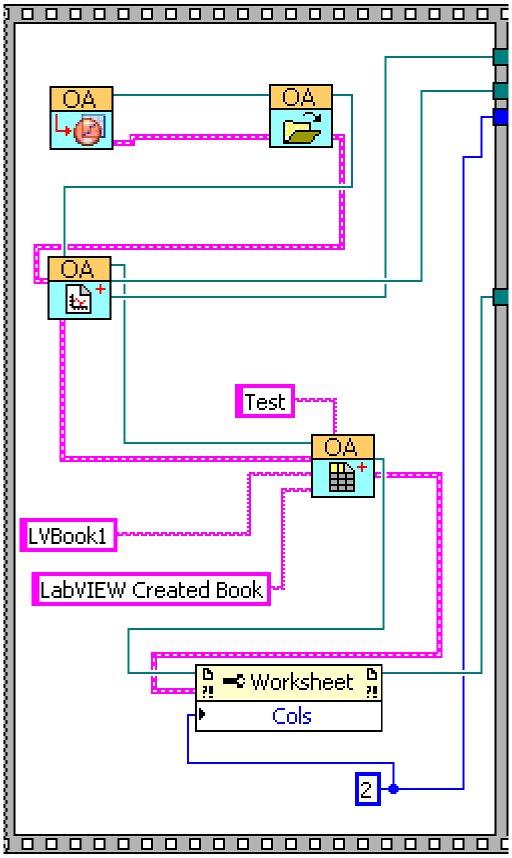
The following is the diagram of the VI. Note that this example has used some OriginCom features.



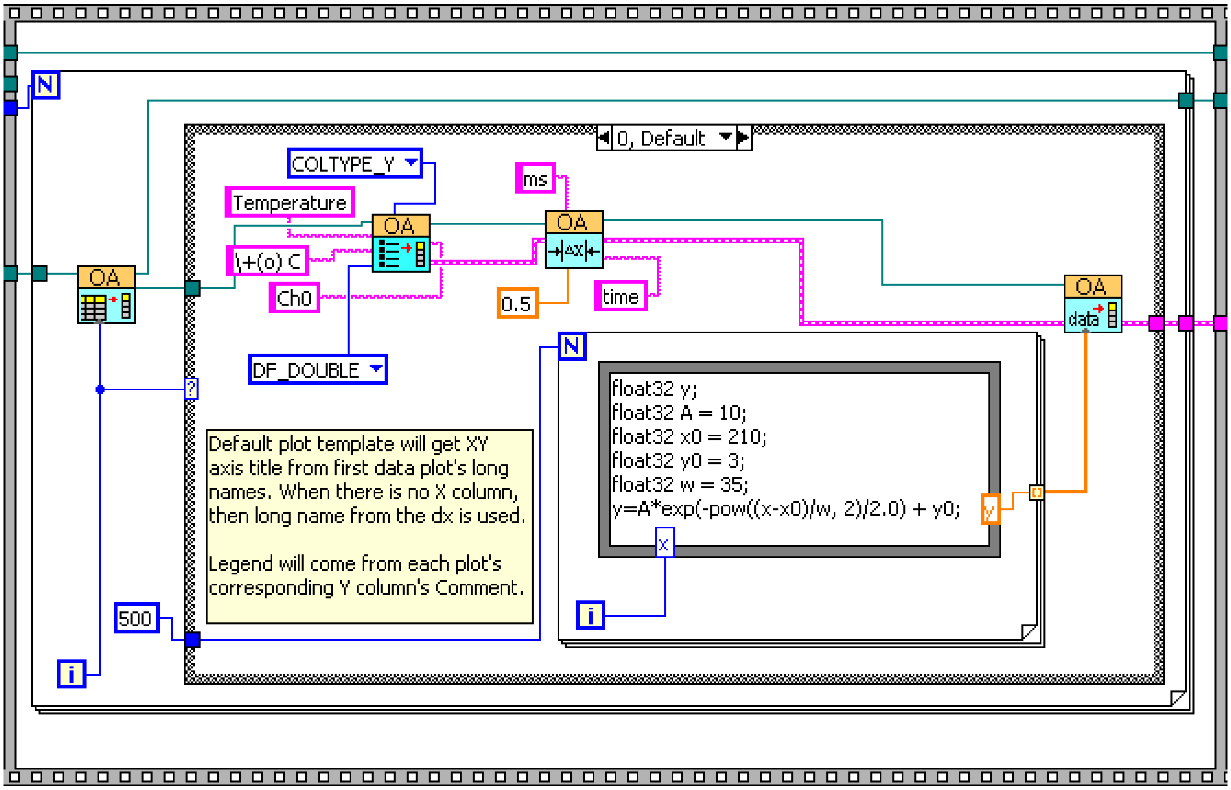
4.12 Plotting Data

This example shows how to create a graph and plot worksheet data to it.

The block below is for creating a new project with a empty graph (OA_NewEmptyGraph), then create a new empty worksheet (OA_NewWorksheet). The Worksheet refnum is then used to call Origin COM property (Worksheet.Cols) to set the 2-column worksheet.



A FOR loop is then used to set both two Y columns (OA_Col-Setting.vi), and with Even Sampling (OA_Col-SetEvenSampling.vi) so that we don't need X columns to make plots. Also it fills both columns with some sample data, in the form of Guassian peak.

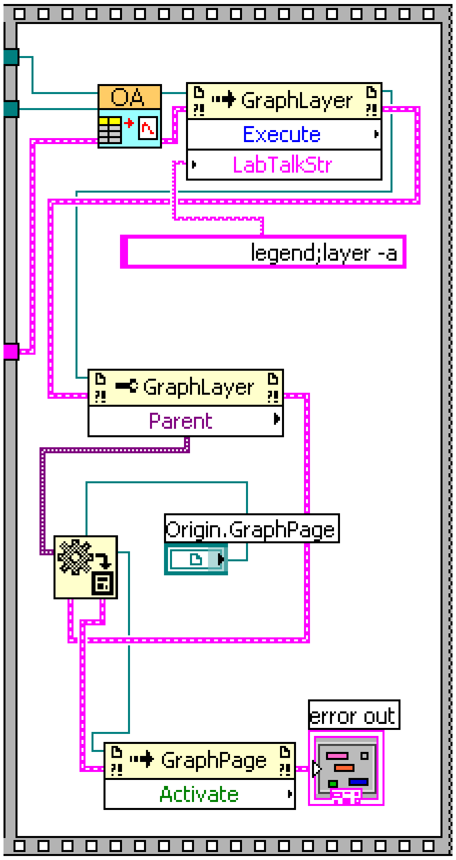


```
float32 y;
float32 A = 10;
float32 x0 = 210;
float32 y0 = 3;
float32 w = 35;
y=A*exp(-pow((x-x0)/w, 2)/2.0) + y0;
```

Default plot template will get XY axis title from first data plot's long names. When there is no X column, then long name from the dx is used.

Legend will come from each plot's corresponding Y column's Comment.

The block below is for plotting(OA_PlotWksCols) the sample data into the graph layer created earlier and to get the GraphPage refnum to use invoke node to call the Activate method, so as to bring the graph to the front.

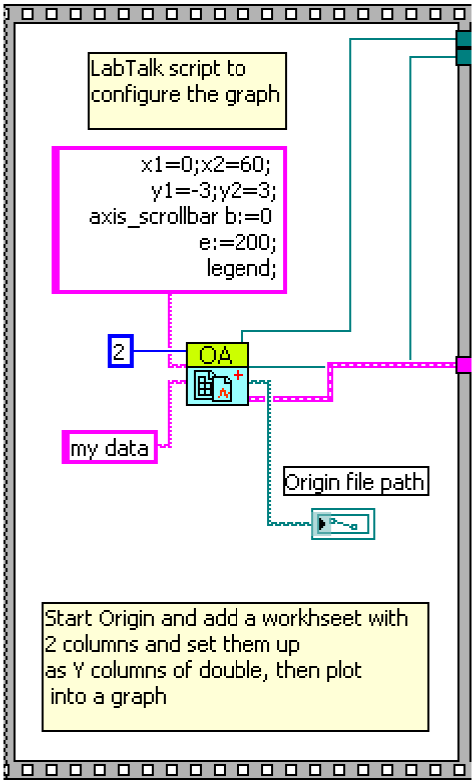


4.13 Sending Waveforms to Origin

This example shows how to send waveform data to Origin. More specifically, the VI demonstrates the following:

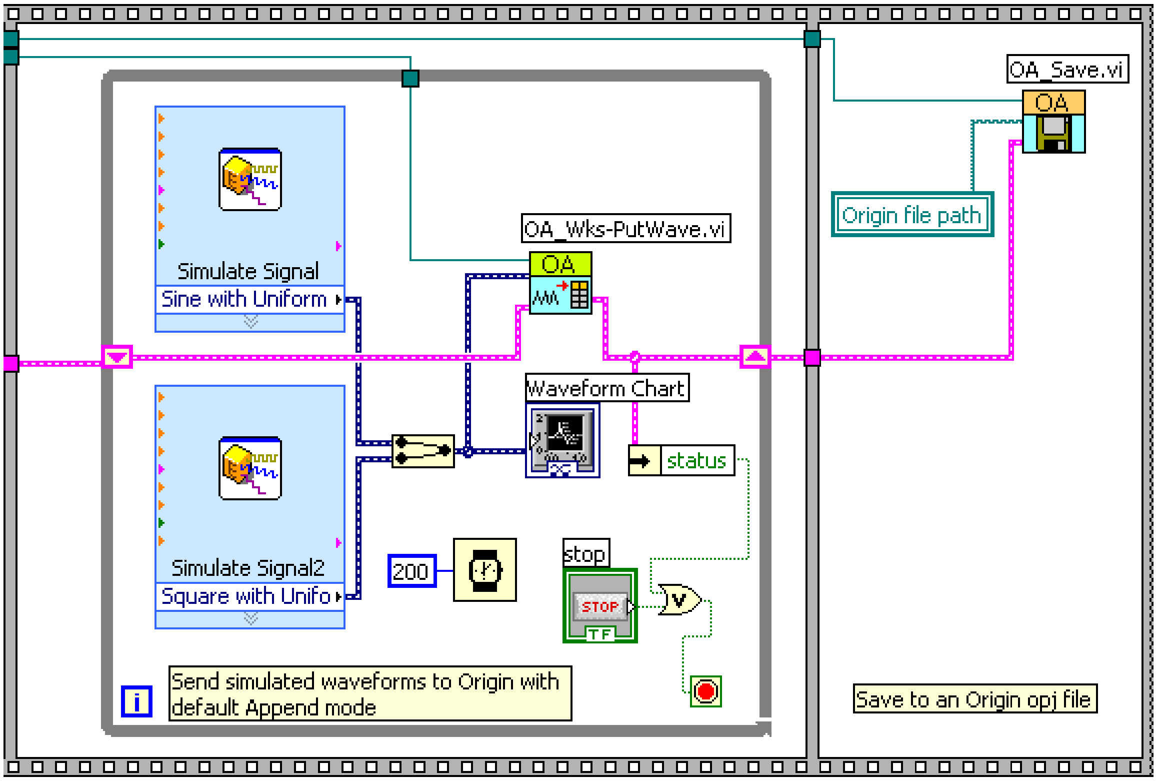
1. Start a new named project and add a worksheet with 2 columns and set them up as Y columns of double, then plot into a graph.
2. Sending the simulated dynamic data to Origin and save the project as a .ogw file.

- The following block shows how to use the OA_NewWaveFile.vi to start a new Origin project named **my data** and add a worksheet with 2 columns. Then set the 2 columns' plot designation as Y and specify the data type of to double and plot them into a graph.



LabTalk script to configure the graph
x1=0;x2=60;
y1=-3;y2=3;
axis_scrollbar b:=0
e:=200;
legend;
2
OA
my data
Origin file path
Start Origin and add a workhseet with 2 columns and set them up as Y columns of double, then plot into a graph

- The following block shows how to send simulated dynamic data to Origin via the OA_PutWave.vi and then save the project as opj file.



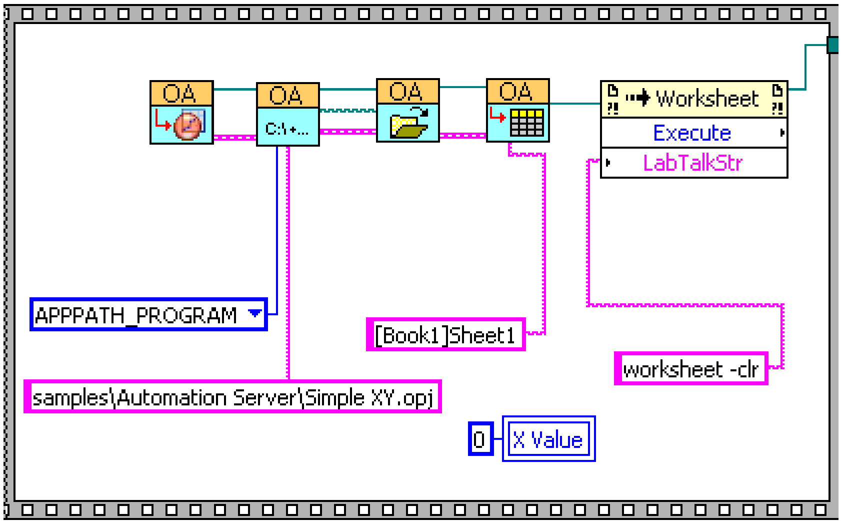
4.14 Sending XY Data To Origin

This example is to complement the Sending Waveforms example. Instead of waveforms, which is essentially just Y columns of data with sample rate, this example shows how to send XY data to Origin. More specifically, the VI demonstrates the following:

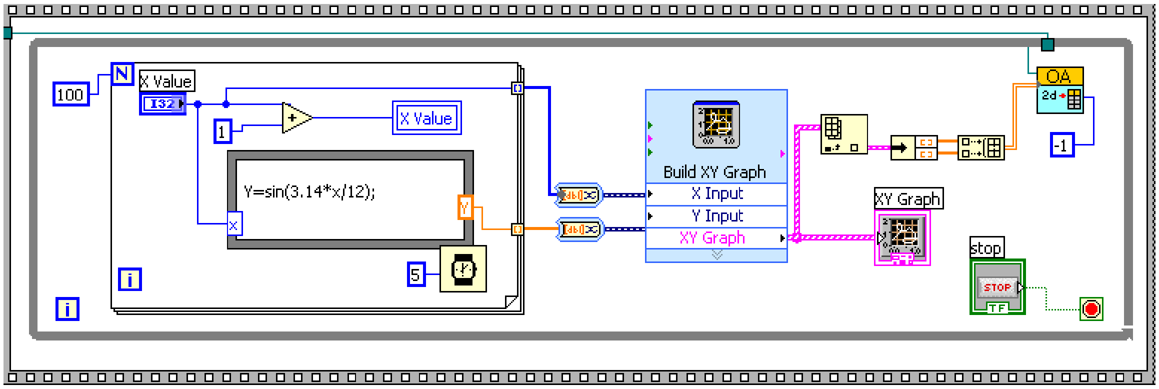
1. Load an opj which contains a worksheet and a graph.
2. Generate two datasets (XY data) and send them to Origin.

This example is mainly composed of two parts.

1. The following block shows how to load an opj file and find the specified worksheet. The found worksheet has also been cleared.



2. The following block shows how to generate two datasets (XY data) and send them to Origin. Please note that a row index of -1 is used to indicate to append data to worksheet, which is a new connected added in SR6.

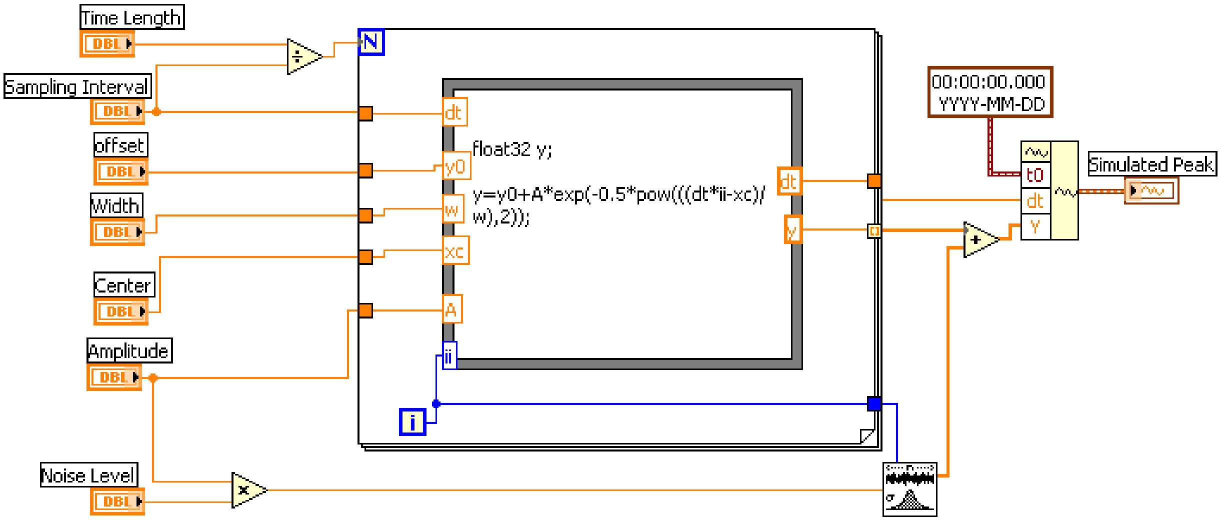


```
Y=sin(3.14*x/12);
```

Note: This VI is available since Origin 8 SR6.

4.15 Simulated Peak

This VI can be used to simulate Gaussian peak. You can set the simulated peak's properties, such as **Amplitude**, **Center**, **Width**, **Offset** etc. The following is the diagram of the VI.



```
float32 y;
y=y0+A*exp(-0.5*pow(((dt*ii-xc)/
w),2));
```

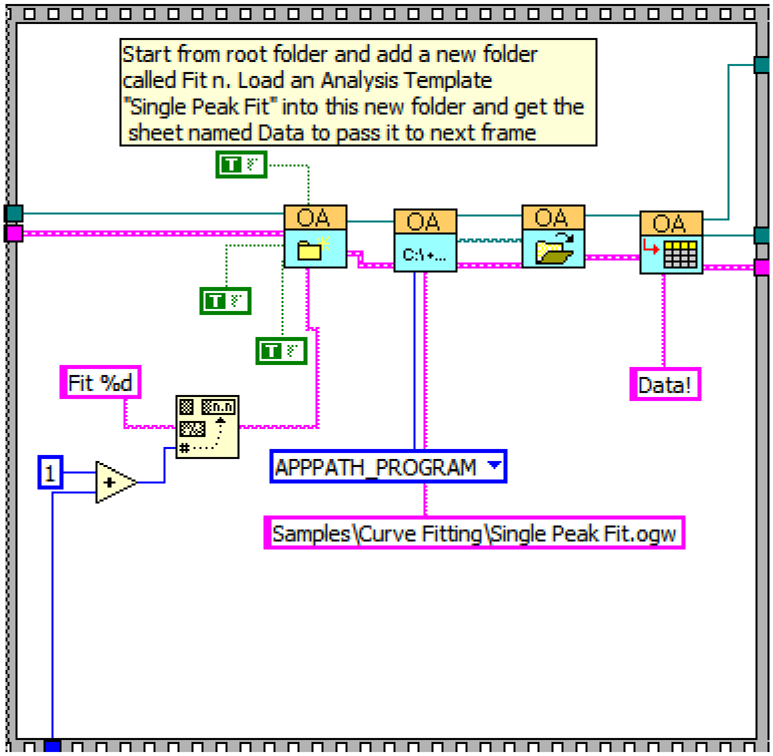
4.16 Analysis Template

This example shows how to load an analysis template into multiple subfolders in Origin and then for each, we put in some simulated data and have the non-linear-fit(Gaussian in this case) to be automatically recalculated. More specifically, the VI demonstrates the following:

1. Creating subfolders in Project Explorer and load an analysis template into it
2. Sending XY data (float32 array in this case) to named sheet (Data in this case) in the template
3. Run the analysis template and active a named sheet(Summary in this case) to show fit results

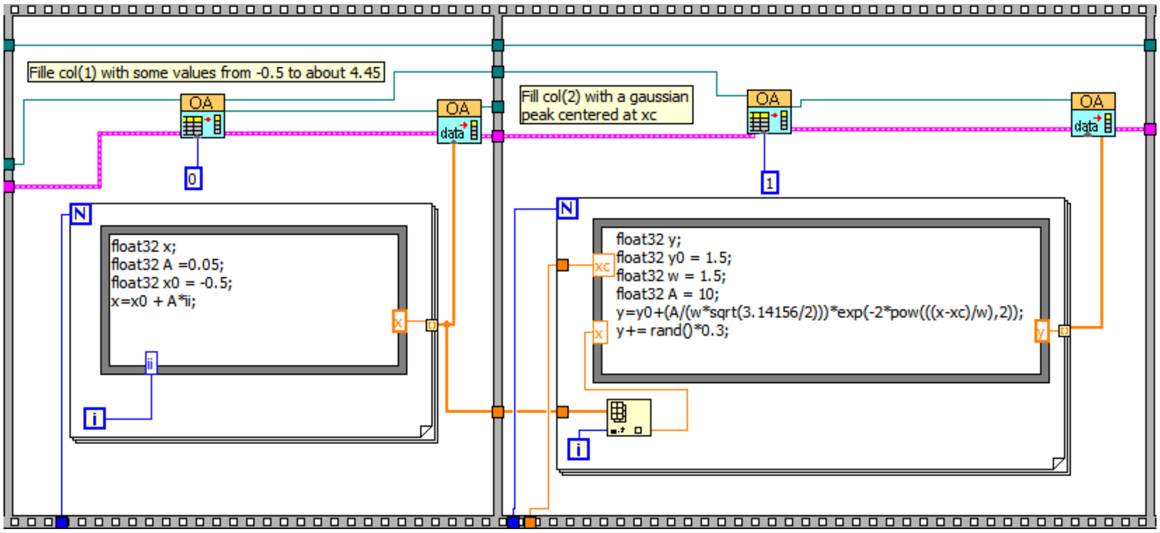
At the end of running the VI, you will have an OPJ file with 3 subfolder(Fit1, Fit2, Fit3), each contains a fit to different data.

- The following block shows how to create a new subfolder from the root and load an analysis template from a relative path into the newly added subfolder



Start from root folder and add a new folder called Fit n. Load an Analysis Template "Single Peak Fit" into this new folder and get the sheet named Data to pass it to next frame

- The following block shows how to send data to Origin, with X data (**float32 1D array** in this example) to col(1) and Y data to col(2) in the sheet named **Data** obtained from the previous frame.



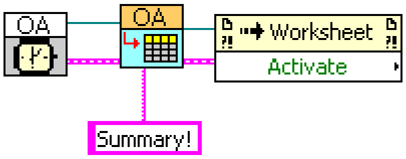
Fille col(1) with some values from -0.5 to about 4.45

```
float32 x;
float32 A =0.05;
float32 x0 = -0.5;
x=x0 + A*ii;
```

Fill col(2) with a gaussian peak centered at xc

```
float32 y;
float32 y0 = 1.5;
float32 w = 1.5;
float32 A = 10;
y=y0+(A/(w*sqrt(3.14156/2)))*exp(-2*pow(((x-xc)/w),2));
y+= rand()*0.3;
```

- The following block shows how to run an analysis template, and then how to activate a sheet (**Summary**) to show fit results



4.17 Waveform Envelope

This example shows you how to access waveform data (.wdt file) and to find an upper envelope with an Analysis Template.

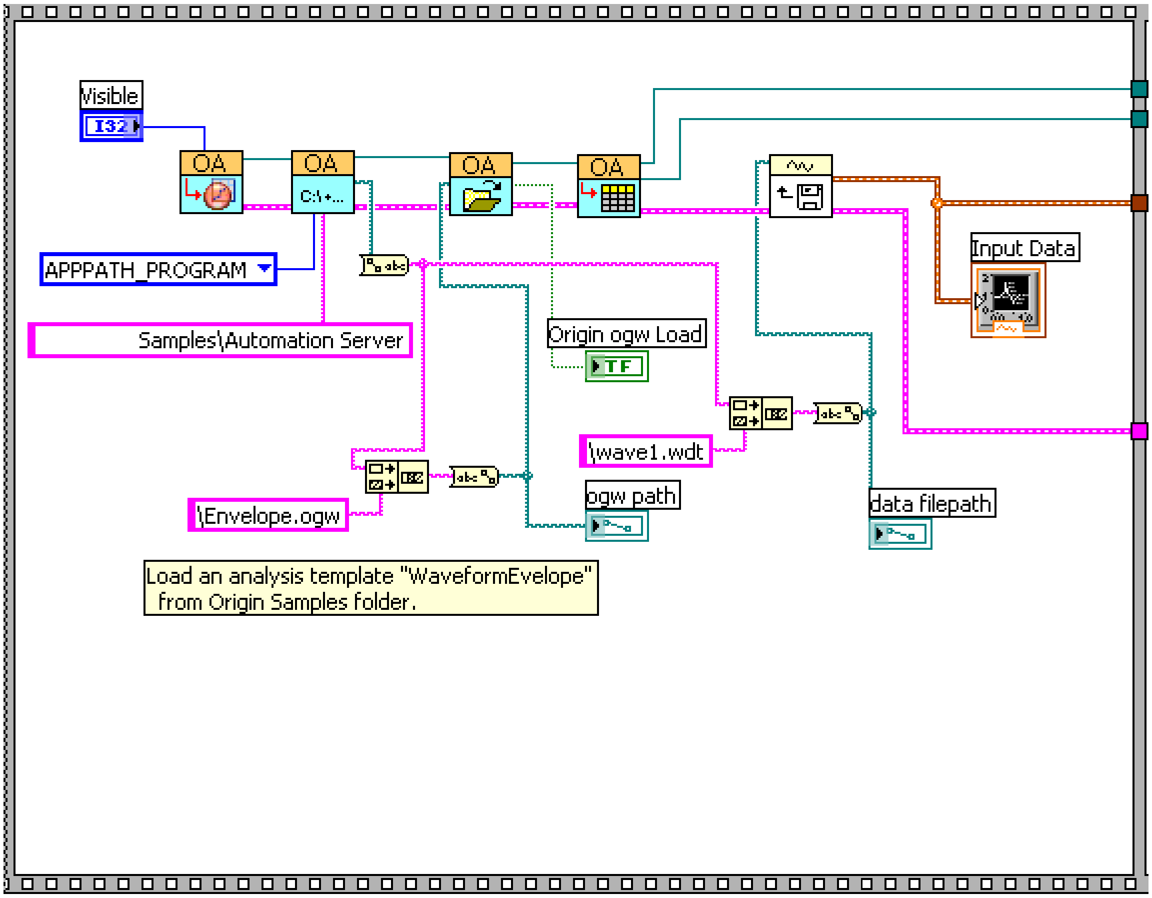
> **Note:** If you want to try this example in Windows 10, please run Labview as administrator at the first time. You should:
>
> 1. Right-click on the Labview program icon and select **Run As Administrator** in the context menu.
> 2. Click **Open Existing** to open the Waveform Envelope.vi
> 3. Click **Convert** button in the pop-up message box after clicking **run** button ⇨.

It is to demonstrate the following:

1. Connect to Origin with OA_ConnectToOrigin.vi which maps to the ApplicationSI method in Origin.
2. Load an Origin Analysis Template (an OGW file) under Origin program folder.
3. Specify a path to load wave1.wdt, which is under <Origin Program Folder>\Samples\COM Server and Client\.
4. Send the data to a sheet named "WaveformEnvelope" in the Analysis Template.
5. Trigger Recalculate in Origin. The Analysis Template will automatically update the "WaveformEnvelope" sheet.
6. Send the found upper envelope and the original data from Origin to LabVIEW, and then plot them.

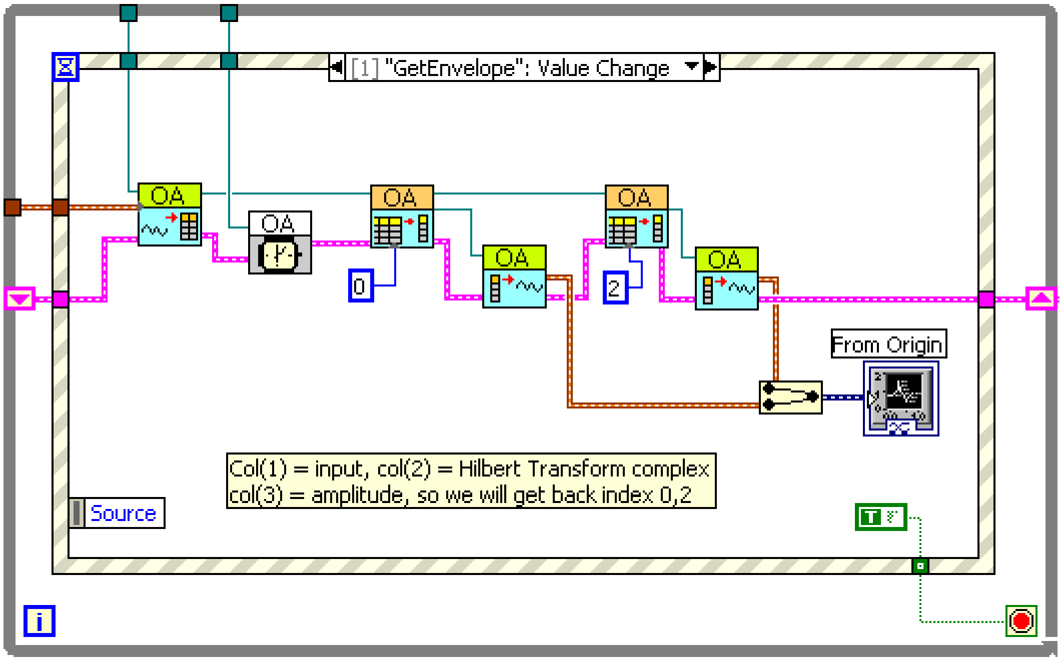
This example is composed of two parts.

- The figure below is to read data from an ASCII file and send the data to the Origin worksheet



Load an analysis template "WaveformEvelope"
from Origin Samples folder.

- The figure below is to perform the envelope finding in Origin, and then plot the envelope and original data in LabView.



Col(1) = input, col(2) = Hilbert Transform complex
col(3) = amplitude, so we will get back index 0,2