

ADvantageSAS 1.0.2.9 Update Summary

This is a maintenance update, and includes ~45 modifications, primarily for ADvantageDB. These include:

- Several minor changes to ADvantageDB, mostly CAN related, and includes some new Python script functions.
- Similar to ADvantageDE, ADvantageDB now handles relative path locations for file generation, allowing for more portable databases.
- Improved command line options for ADvantageFC.
- In conjunction with new menu options in ADvantageVI, the Event Log Viewer now supports new command line options for opening multiple UDP port pages.
- SIMplotter now uses information from running ADvantageVI sessions to allow “selection” of DAS ports to connect to.
- Some changes and additions have been made to the SIMplotter Python module for better data access.

Note: several of these changes are related to logical device and ADvantageVI changes included in ADvantage update 8.4.0.12.

Major Enhancements

This update includes some major changes and enhancements to SIMplotter.

Data Storage

The underlying data storage used by SIMplotter has been changed.

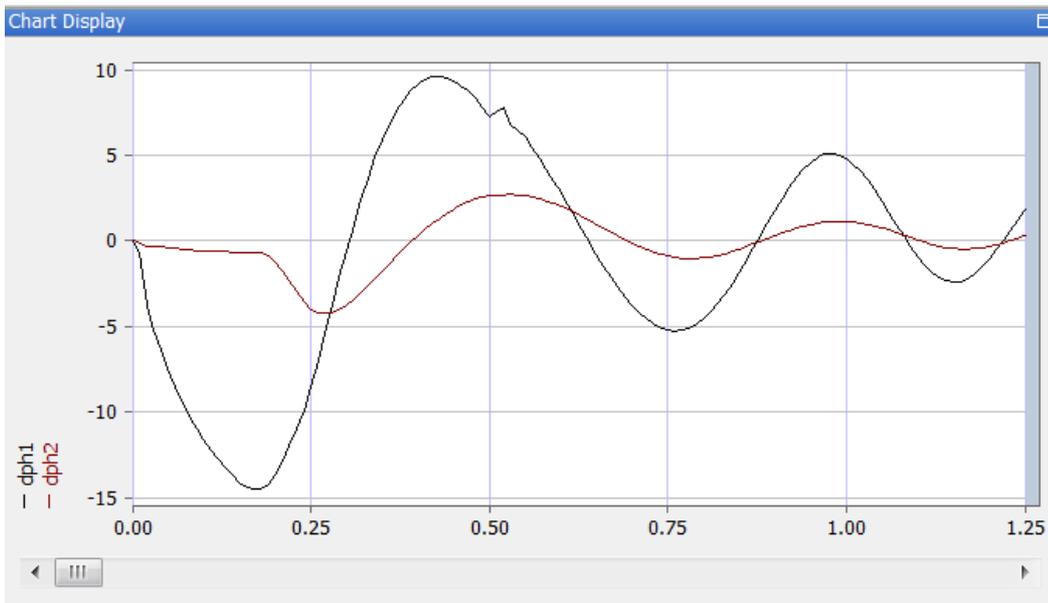
Previously, SIMplotter used a memory buffering scheme where an initial buffer was allocated based on the size of the incoming data frame and a System Preference for Max Dataset Buffer Frames. As data filled the buffer it would also be spooled to disk. When the buffer got full, new data would overwrite existing data in the buffer, allowing at most only the number of frames specified by the setting to be available for viewing. When saving the dataset, the spool files were used, so there was not actual loss of data, just a limitation on what could be viewed or used from scripting for analysis.

With this new version, SIMplotter makes use of a memory caching page file scheme, such that data is still spooled to disk as needed, and the entire spool file is still used when saving the dataset. However, now the viewing of data for charts and processing by Python script has access to the entire dataset by reading in “pages” of frame data from the disk file as needed. The previous system setting has been replaced with a new Dataset Page File (Megabytes) setting. This value determines the total memory size of the page to use for processing data (per dataset). The smaller the value, the more memory pages must be swapped in and out of memory from disk. The larger the value, the faster the performance, but care must be taken to not exceed the available system memory if multiple datasets are in use.

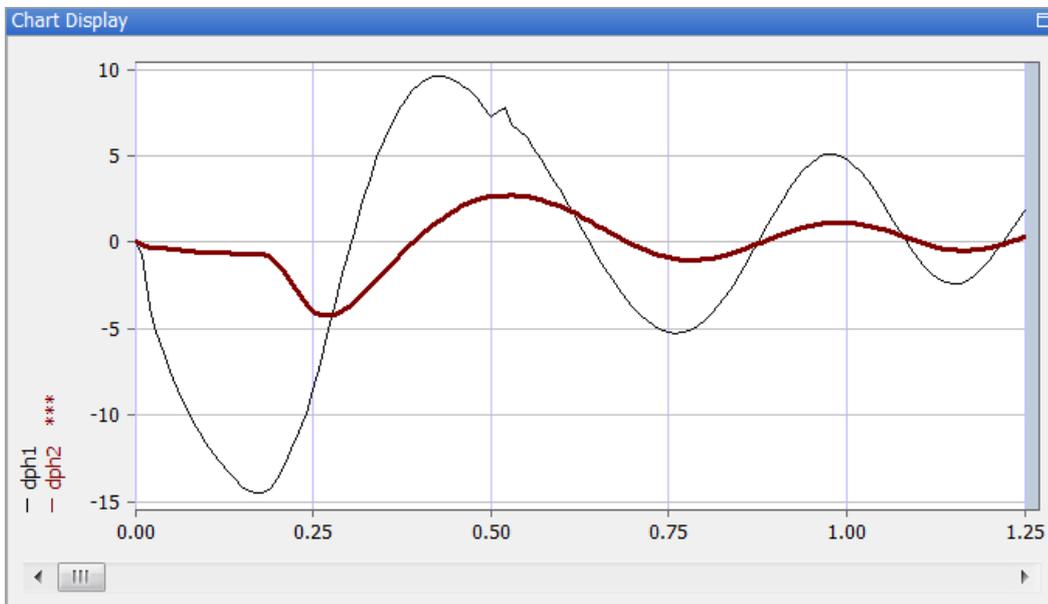
This change should be transparent to the user, other than now being able to access ALL received data.

Selectable Traces

Both the StripChart and XY Chart views now support selectable traces. The following shows a simple strip chart view window:



The chart legend is used to select a trace. Left-click on the legend text for the desired trace to select it:



The selected trace shows with asterisks in the legend, and is drawn in a broader pen in the view. Only one trace per chart may be selected, but note that a view may contain multiple charts. To un-select the trace, simply left-click in empty space in the legend area.

With a trace(s) selected, pressing the Delete key will remove the selected trace(s). If this is done accidentally, pressing Ctrl-Z will Undo the delete.

Support for working with the selected trace has been added to the simplot Python module:

```
nTraceId = simplot.Get(simplot.Chart, nChartId, simplot.ChartSelectedTraceId)
```

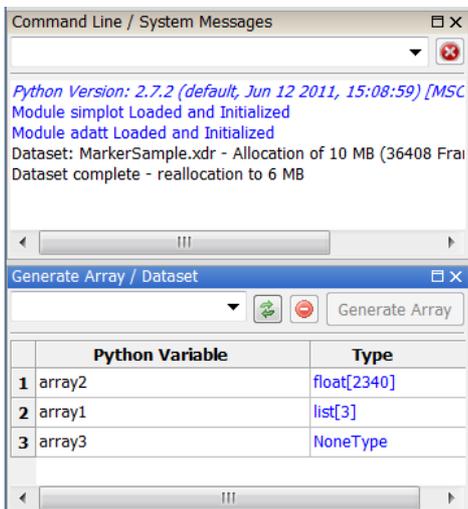
Additionally, selected traces may be used by the new Python Variables View window (see below), and for adding chart Data Markers (see below).

Python Variables View

Functionality has been added to allow manual creation of Python variables, and the translation of Python array variable data to/from Dataset or Trace variables. A new dockable view is now available from the View menu:



Selecting Show Python Variables window will open this new docked window:



This window has the following components:

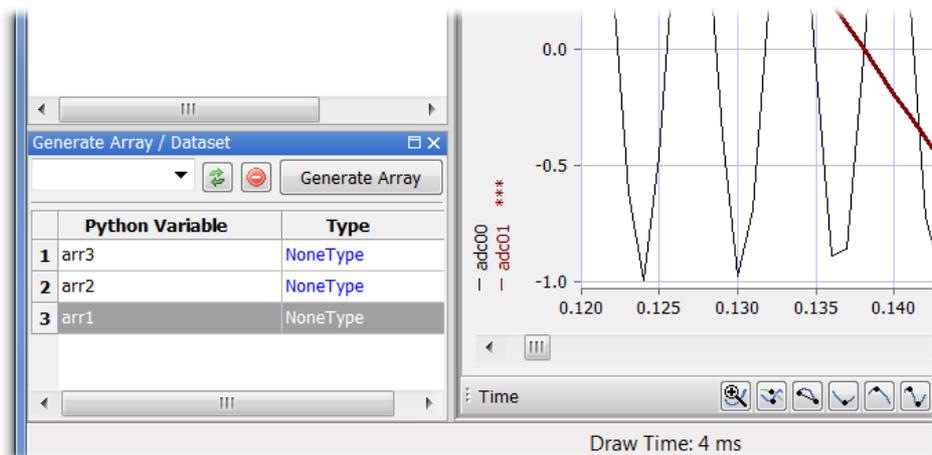
- Variable entry combo box – this allows the user to enter the Python variable names to create. Note that when a variable is first created, it will be of type None.
- Variable table – this shows all the defined Global Python variables and their types / dimensions. Some operations require a table row(s) to be selected – this is done in the usual manner by clicking on the row label to the left.
- Refresh button – will update the variable table.
- Delete button – this will delete the Python variables for the selected table rows.
- Generate Array / Variable button – executes the specified data translation (see below).

This window can perform the following translation functions:

Populate Python Variable from Selected Trace

To perform this function, navigate to an open chart view and select a single chart trace. Note that if more than one trace is selected in all the charts / views, this feature is disabled as the source of the data is ambiguous.

Next, select a table row to select the Python variable to assign the trace data to. This will then enable the Generate Array button.:



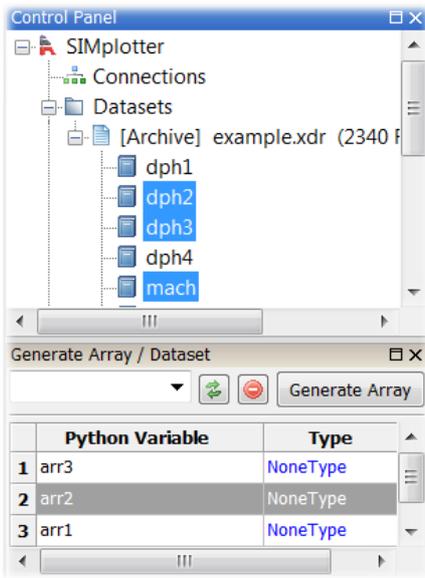
Pressing the Generate Array button will make the selected Python variable an array containing the data from the selected trace. Note that only the data from the viewable time range area of the chart is copied to the array, not the entire range from the dataset.

If the selected trace is from an XY chart, the Python variable will contain two arrays, one for the X values and one for the Y values. Again, only data from the source chart's time range will be copied to the arrays.

Populate Python Variable from Selected Tree Variable(s)

To perform this function, navigate to the Control Panel tree, expand a dataset item to view its variables, then select one or more variables from the dataset.

Next, select a table row to select the Python variable to assign the trace data to. This will then enable the Generate Array button:

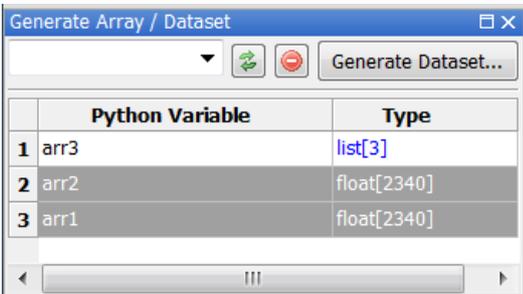


Pressing the Generate Array button will make the selected Python variable an array containing the data from the selected variable. Or if multiple dataset variables selected, the Python variable will become an array of arrays, one for each selected variable (order as shown in the tree).

Unlike using a selected trace, using this method will copy the entire dataset range of values into the array(s). If multiple variables are selected, the Python variable will contain N arrays, one for each of the variables.

Create User Dataset from Selected Python Variables

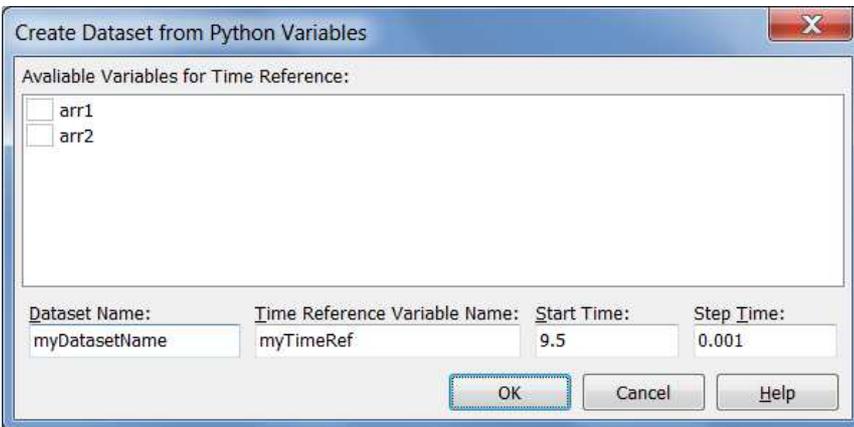
To perform this function, select one or more variable rows from the table. This will enable the Generate Dataset button.:



Note that if rows are selected, and Control Panel tree variables or a trace is selected, the button will still be displaying Generate Array. To use this feature, make sure now tree traces are selected, and not chart traces are selected.

The Generate Dataset button will only be enabled if the selected table row(s) contain python array variables, and if multiple rows are selected, the array dimensions of the selected variables must be the same.

When the Generate Dataset button is pressed, the Create Dataset dialog will be displayed:



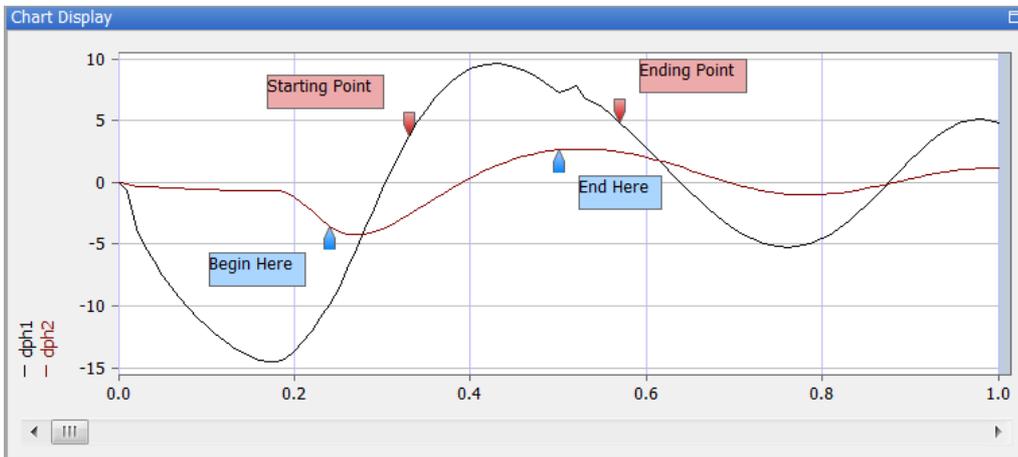
The list will show the names of all the variables selected from the table. To create a dataset, a Time Reference must be specified. From this dialog, the user may choose one of the variables listed to be the Time Reference, in which case the Time Reference Variable Name, Start Time, and Step Time fields will be disabled.

Alternately, the user may time in a name, start time, and step time, and a variable will automatically be generated along with the listed variables. In this case, frames will be generated with time stamps beginning from the specified start time, and incrementing by the specified step time.

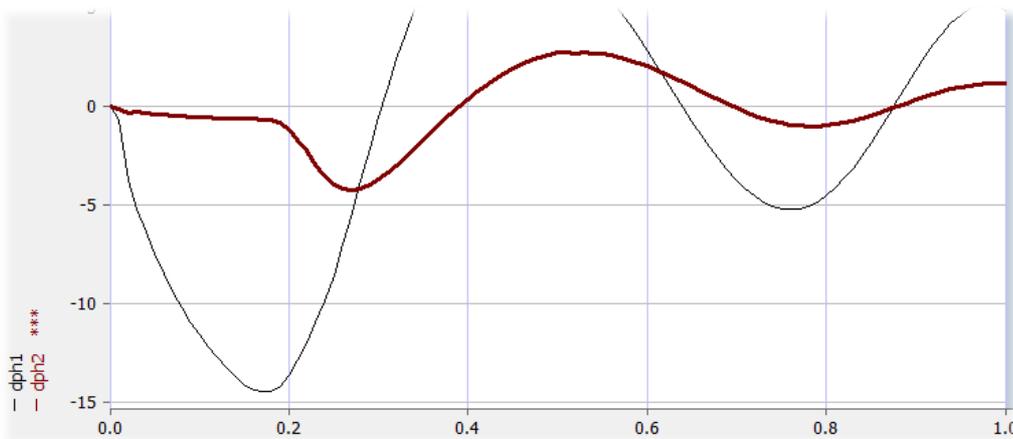
A Dataset Name must also be specified. Pressing Ok will generate the new dataset, using values from the supplied Python variables and optionally generating the time reference variable.

Data Markers

A Data Marker is a small symbol with optional text that can be assigned to a specified variable at a specified time. Data Markers are stored internally as specially formatted comments within the dataset, but appear in the chart view when a trace is shown corresponding to the variable:



To add markers to a chart (dataset variable), first “select the trace” by right-clicking on the desired trace legend text:

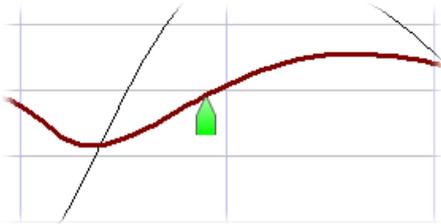


The trace will be drawn in a wider pen, and the selected trace legend text will show ***. With the trace selected, click on the Chart Data Markers tool from the chart toolbar:



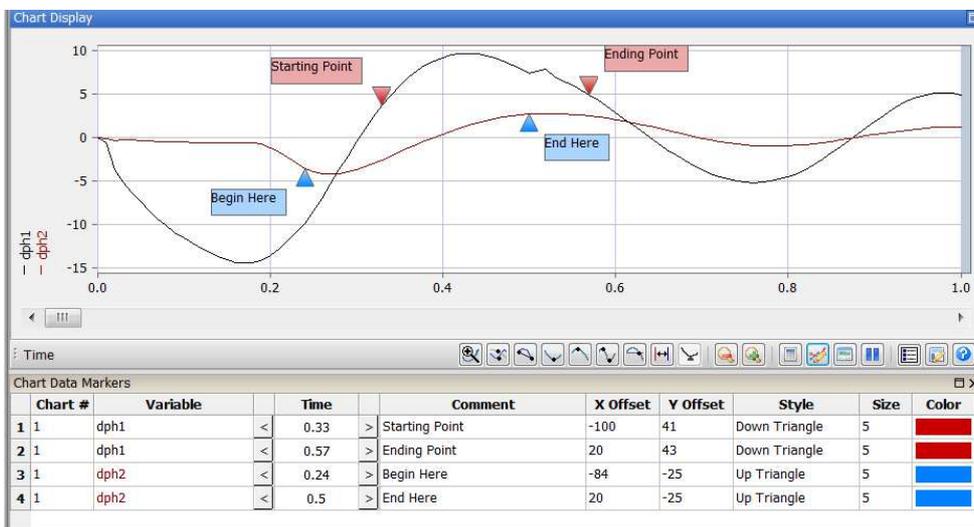
The addition of data markers to the chart is done in the same manner as adding Chart Readout marks. Note that the Data Marker toolbar button is disabled unless one or more traces in the view are selected.

Move the mouse to the desired X (time) location over the chart and left-click. SIMplotter will create a Data Marker on the selected trace for that time value:



Note that there are new System Preference settings for the default size, style, and color of the Data Markers.

When the Chart Data Markers tool is selected, the chart view will automatically display the Data Marker Table pane under the chart:



As data markers are added to the chart, rows will be appended to this table. The user may adjust the marker properties, or delete markers using this table. The following fields are available:

- Chart # and Variable – shown for reference.
- Time – displays the time value associated with the data marker. The user may click the < or > buttons to move the time value left or right, or with the time cell selected, press the < or > keys on the keyboard (or command or period of the shift key is not pressed).
- Comment – the user may optionally enter text to be displayed along with the data marker symbol. Note the text entry is single line only, but the display will auto-wrap the text when displayed.
- X, Y Offsets – these values control the pixel offset of the text box relative to the marker point.
- Style, Size, Color – these control the display characteristics of the data marker symbol.

To delete a data marker(s), select the desired row(s) in the table, and right-click in the table to bring up the popup menu:

Time	Comment	X Offset	Y Offset	Style	Size	Color
33	> Starting Point	-100	41	Down Triangle	5	Red
57	> Ending Point	20	43	Down Triangle	5	Red
24	> Begin Here				5	Blue
5	> End Here				5	Blue

Scroll to Data Marker

Delete All Chart Data Markers

Delete Selected Data Marker(s)

Using this popup menu, selected or all data markers can be deleted.

Also available from this popup menu is the Scroll to Data Marker feature. If the chart scale is such that a particular marker is not visible, right click on the row containing the marker and select this popup menu choice. The affected chart will be scrolled in time, such that the specified marker becomes visible.

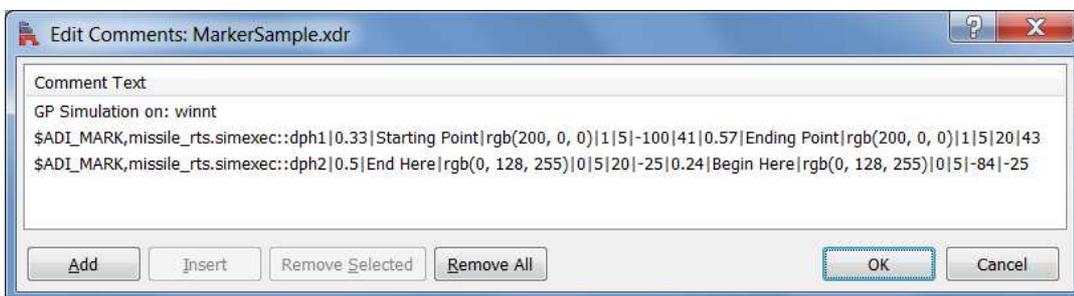
The display of Data Markers on the chart view is controlled by adjusting the Data Marker Mode using the chart toolbar button:



Pressing this button will cycle through the modes:

1. No data markers displayed
2. Data markers only displayed
3. Data markers with text displayed
4. Data markers with text displayed and Marker Table shown (automatically selected when the Data Marker tool is selected to add new markers). Note that there is a new System Preference for the default marker mode to use when displaying a new chart view.

As mentioned, data markers are stored as specially formatted comments within the dataset. As such, the user may add markers via Python script commands by adding comments to a dataset:



The format of a data marker comment line is:

\$ADI_MARK,variableName|time|text|color\style\size\xOffset|yOffset|time|text|...

Note a comma delimits the first two fields, then the | symbol delimits each marker entry after that. The text used for defining a color can be a simple RGB value:

|rgb(200,0,0)|...

There can be only one comment line for data markers for each variable.
Please refer to the release notes for detailed information.

<http://www.adi.com/ftp/downloads/ADvantageSAS/ReleaseNotes.htm>