

ADvantage 8.4.0.5 Update Summary

1 Minor Fixes / Enhancements

This update includes ~30 minor enhancements and fixes, including:

- New Python script commands to launch ADATT user interface
- Additional display options for ADEPT panels
- New script commands for opening and closing ADEPT panels
- Identification of missing device files at build time rather than run time
- Removal of port type matching restriction for device connections

2 Major Enhancements

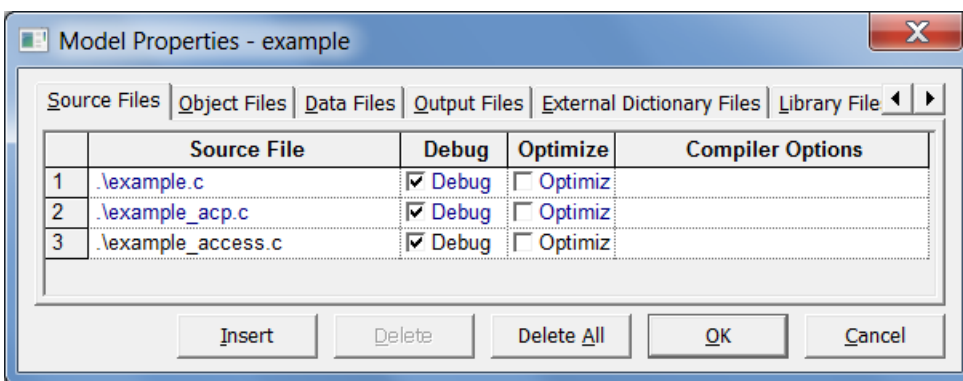
This update includes the following major enhancements:

Auto-Connect Ports

Customers have identified a need to be able to control I/O directly from ADvantageVI without the necessity of adding variables (ports) to their model. To facilitate this, I have added a feature to ADvantageDE called “Auto-Create Ports”, or ACP.

For this to work within our current constructs, the following are required:

- An actual model variable must exist in the source somewhere. DE will generate a “special” C source file with these variables declared. This implies only model types that support additional C files can support this feature (C, C++, Fortran, Simulink, Adsim). This additional source will be added automatically to the Model Properties additional source file list.



- A model port must then be created for this variable – this requires the variable appear in the model dictionary. We already support “merging” both the _src and _debug dictionaries that come from our bridge code. DE will now generate an _acp dictionary that contains these variables and will be merged in a similar fashion.

- A connection must be made between the model port and the device port.

With all this in place, the variable can appear in the data browser, and the user can view device inputs and change device outputs, all outside the normal model computation code.

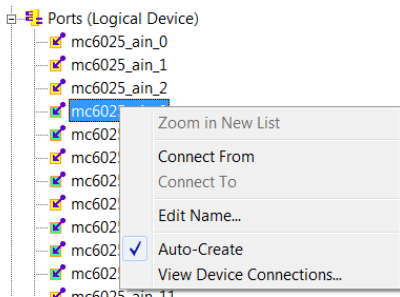
A user can now designate a logical device as “Auto-Create” via the Device Port Connections dialog (I chose this since it provides a nice view of all a device(s) ports):

	Device Name	Port Name	Port Addr	Type	Direction	Elements	Description	Connected	Auto-Create
1	rtx9_mc6025_ain	mc6025_ain_0	0	float	Input	1	MC6025 Analog Input Channe	<input checked="" type="checkbox"/> Connected	<input checked="" type="checkbox"/> Auto-Create
2	rtx9_mc6025_ain	mc6025_ain_1	1	float	Input	1	MC6025 Analog Input Channe	<input type="checkbox"/> Connected	<input type="checkbox"/> Auto-Create
3	rtx9_mc6025_ain	mc6025_ain_2	2	float	Input	1	MC6025 Analog Input Channe	<input type="checkbox"/> Connected	<input type="checkbox"/> Auto-Create
4	rtx9_mc6025_ain	mc6025_ain_3	3	float	Input	1	MC6025 Analog Input Channe	<input checked="" type="checkbox"/> Connected	<input checked="" type="checkbox"/> Auto-Create
5	rtx9_mc6025_ain	mc6025_ain_4	4	float	Input	1	MC6025 Analog Input Channe	<input checked="" type="checkbox"/> Connected	<input checked="" type="checkbox"/> Auto-Create
6	rtx9_mc6025_ain	mc6025_ain_5	5	float	Input	1	MC6025 Analog Input Channe	<input type="checkbox"/> Connected	<input type="checkbox"/> Auto-Create
7	rtx9_mc6025_ain	mc6025_ain_6	6	float	Input	1	MC6025 Analog Input Channe	<input checked="" type="checkbox"/> Connected	<input checked="" type="checkbox"/> Auto-Create
8	rtx9_mc6025_ain	mc6025_ain_7	7	float	Input	1	MC6025 Analog Input Channe	<input type="checkbox"/> Connected	<input type="checkbox"/> Auto-Create
9	rtx9_mc6025_ain	mc6025_ain_8	8	float	Input	1	MC6025 Analog Input Channe	<input checked="" type="checkbox"/> Connected	<input checked="" type="checkbox"/> Auto-Create
10	rtx9_mc6025_ain	mc6025_ain_9	9	float	Input	1	MC6025 Analog Input Channe	<input checked="" type="checkbox"/> Connected	<input checked="" type="checkbox"/> Auto-Create
11	rtx9_mc6025_ain	mc6025_ain_10	10	float	Input	1	MC6025 Analog Input Channe	<input checked="" type="checkbox"/> Connected	<input checked="" type="checkbox"/> Auto-Create
12	rtx9_mc6025_ain	mc6025_ain_11	11	float	Input	1	MC6025 Analog Input Channe	<input type="checkbox"/> Connected	<input type="checkbox"/> Auto-Create
13	rtx9_mc6025_ain	mc6025_ain_12	12	float	Input	1	MC6025 Analog Input Channe	<input checked="" type="checkbox"/> Connected	<input checked="" type="checkbox"/> Auto-Create
14	rtx9_mc6025_ain	mc6025_ain_13	13	float	Input	1	MC6025 Analog Input Channe	<input type="checkbox"/> Connected	<input type="checkbox"/> Auto-Create
15	rtx9_mc6025_ain	mc6025_ain_14	14	float	Input	1	MC6025 Analog Input Channe	<input checked="" type="checkbox"/> Connected	<input checked="" type="checkbox"/> Auto-Create
16	rtx9_mc6025_ain	mc6025_ain_15	15	float	Input	1	MC6025 Analog Input Channe	<input checked="" type="checkbox"/> Connected	<input checked="" type="checkbox"/> Auto-Create
17	rtx9_mc6025_aout	mc6025_aout_0	0	float	Output	1	MC6025 Analog Output Chan	<input checked="" type="checkbox"/> Connected	<input checked="" type="checkbox"/> Auto-Create
18	rtx9_mc6025_aout	mc6025_aout_1	1	float	Output	1	MC6025 Analog Output Chan	<input checked="" type="checkbox"/> Connected	<input checked="" type="checkbox"/> Auto-Create

12 of 18 Ports Connected

The last column is new and is the only editable field.

Alternately, the Auto-Create designation is also available from the port popup tree menu:



You will see “auto-create” ports appear with a differently colored icon in the tree.

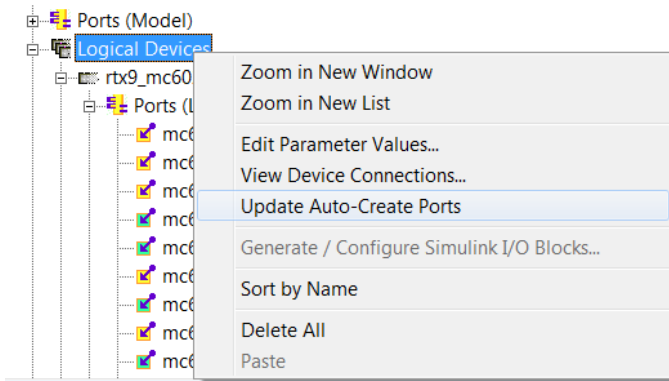
When a change to a ports ACP flag is made, DE will automatically go through the steps outlined above to re-generate an additional source file, re-generate the ACP dictionary, and make connections between new model ports and ACP device ports:

```

10:22:51: Removing all existing auto-created ports and connections.
10:22:51: Processing 11 Auto-Connect Ports (ACP) ...
10:22:51: ACP dictionary and source file generated.
10:22:52: Adding ACP connection from Device rtx9_mc6025_ain Port mc6025_ain_3
10:22:52: Adding ACP connection from Device rtx9_mc6025_ain Port mc6025_ain_4
10:22:52: Adding ACP connection from Device rtx9_mc6025_ain Port mc6025_ain_6
10:22:52: Adding ACP connection from Device rtx9_mc6025_ain Port mc6025_ain_8
10:22:52: Adding ACP connection from Device rtx9_mc6025_ain Port mc6025_ain_9
10:22:52: Adding ACP connection from Device rtx9_mc6025_ain Port mc6025_ain_10
10:22:53: Adding ACP connection from Device rtx9_mc6025_ain Port mc6025_ain_12
10:22:53: Adding ACP connection from Device rtx9_mc6025_ain Port mc6025_ain_14
10:22:53: Adding ACP connection from Device rtx9_mc6025_ain Port mc6025_ain_15
10:22:53: Adding ACP connection to Device rtx9_mc6025_aout Port mc6025_aout_0
10:22:53: Adding ACP connection to Device rtx9_mc6025_aout Port mc6025_aout_1

```

There is also a popup menu option to manually do this processing:



Note that the first step in the ACP update is to remove all ACP information from the project in the event the driving change is such that no ACP ports now exist.

Once made, ACP connections are no different than any other connections. A device port connected in this way is un-available for a standard connection. Since device ports cannot be multiply connected, if a standard model connection to an ACP port is desired, the user must first turn off the ACP flag for that port. Then it may be connected in the standard manner. Likewise, if an existing standard connection is in place between a device port and a model port, and the device port is changed to ACP, this new connection will force removal of the previous normal connection.

Also note that any new model ports created for connection to ACP ports will appear in their own group in the VI data browser window to prevent confusion with other model ports:

	Location	Name	Value	Units	Type	Description
✖ mc6025_ain_10	example	mc6025_ain_10	0.0000		F	MC6025 Analog Input Channel 10 (ADI Auto-Created)
✖ mc6025_ain_12	example	mc6025_ain_12	0.0000		F	MC6025 Analog Input Channel 12 (ADI Auto-Created)
✖ mc6025_ain_14	example	mc6025_ain_14	0.0000		F	MC6025 Analog Input Channel 14 (ADI Auto-Created)
✖ mc6025_ain_15	example	mc6025_ain_15	0.0000		F	MC6025 Analog Input Channel 15 (ADI Auto-Created)
✖ mc6025_ain_3	example	mc6025_ain_3	0.0000		F	MC6025 Analog Input Channel 3 (ADI Auto-Created)
✖ mc6025_ain_4	example	mc6025_ain_4	0.0000		F	MC6025 Analog Input Channel 4 (ADI Auto-Created)
✖ mc6025_ain_6	example	mc6025_ain_6	0.0000		F	MC6025 Analog Input Channel 6 (ADI Auto-Created)
✖ mc6025_ain_8	example	mc6025_ain_8	0.0000		F	MC6025 Analog Input Channel 8 (ADI Auto-Created)
✖ mc6025_ain_9	example	mc6025_ain_9	0.0000		F	MC6025 Analog Input Channel 9 (ADI Auto-Created)
✖ mc6025_aout_0	example	mc6025_aout_0	0.0000		F	MC6025 Analog Output Channel 0 (ADI Auto-Created)
✖ mc6025_aout_1	example	mc6025_aout_1	0.0000		F	MC6025 Analog Output Channel 1 (ADI Auto-Created)
ADL_ACP_PORTS						
ADL_APPLICATION						
ADL_EXTERNAL						
ADL_INTERACT						
ADL_NET						
ADL_RUNSPECS						
ADL_STATISTICS						
DEBUG						

Scripted ADATT Report Generation

All ADATT functions are now provided via Python script commands. Customers have expressed a desire to use scripted test sequences rather than the tree-based user interface. Since Python already provides all other capabilities of ADATT (prompting users, execution control within a file and to multiple files, delays, condition checks, etc.), the addition of the report capability within a script completes the set of functionality needed to replace the UI.

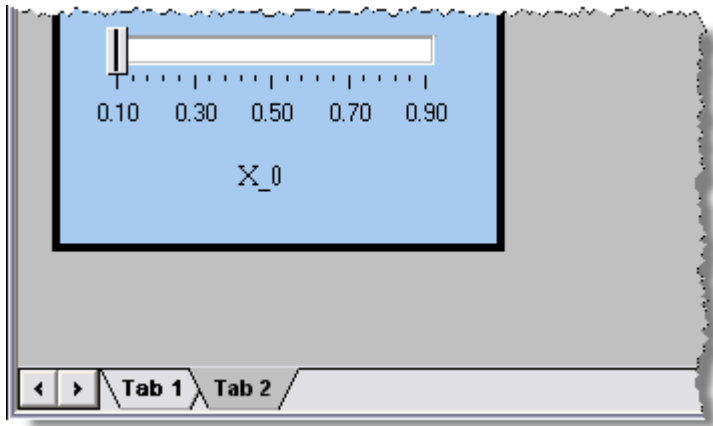
The underlying report generation library has been completely re-written using SAS development tools. This will allow for better maintainability and support of future enhancements. With this re-design, the following enhancements have been made:

- Simpler table generation – the command to generate a table now takes #rows and #cols parameters.
- Table items – are now specified by row/col parameters, and can have individual pass/fail criteria.
- Merged cells – generated report tables may now include merged cells.
- Support for multiple reports – all report interactions are id based, so multiple reports may be populated and generated in parallel during a run.

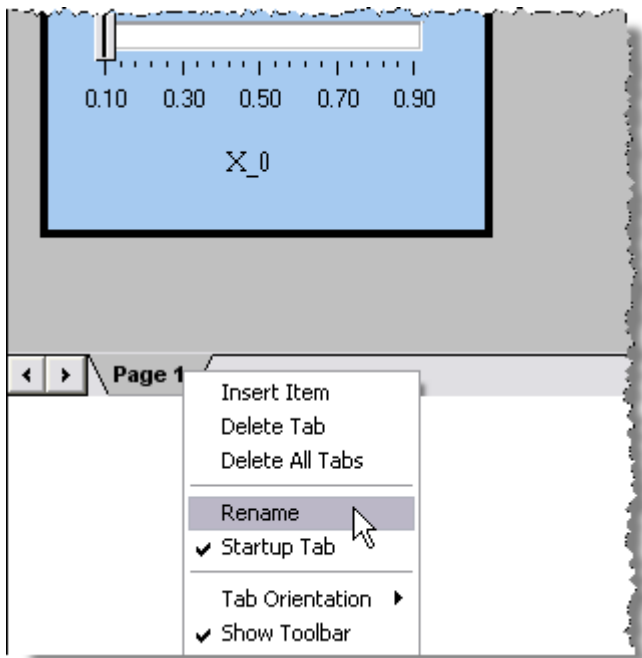
It is worth noting that these changes are supported in both the script command and user interface control of ADATT.

Tabbed ADEPT Panel Pages

The ADEPT panel editor window now supports multiple tabbed pages within a single panel:



Panel widget objects may be cut/pasted between tabs, and several options for the tabs are available:



Please refer to the documentation for additional details.

Interact2Python Conversion Tool

This version includes the new **Interact2Python** conversion tool. This is a command line tool for converting existing ADvantageVI and SIMplotter Interact script files to Python. Please refer to the documentation for additional details.

Devices / Drivers

This update adds support for the following new logical devices:

- Applied Data Sciences HSD2 (completed from update 8.4.0.4)
- Persistent version of Ballard 429-423
- New variant of Serial IO Power Supply – addition of current output port
- PCI-MDAC
- TEWS TIP-606 (RTS)
- Pickering 50-110 Reed Relay
- Measurement Computing PDISO16 Isolated Input and Relay
- NI-6528, PCI-ARB, and PCI-AO32 modified to optionally hold output state between runs.

Please refer to the release notes for detailed information.

<https://www.adi.com/downloads/ADvantage8.4/ADvantageReleaseNotes.htm>