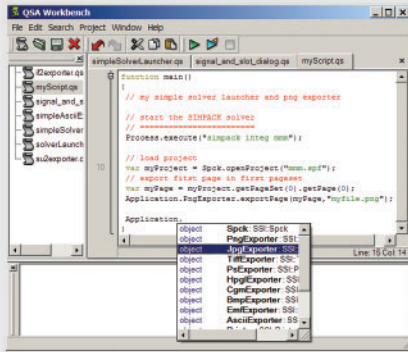


Dr. Trautenberg,  
INTEC GmbH

# Customizing and Scripting SIMPACK



*Integrated Development Environment for Generating Scripts within SIMPACK*

Beginning with the new SIMPACK Plot module, INTEC is introducing an all new scripting engine to SIMPACK. The scripting engine is based on QSA, an object oriented scripting language similar to JavaScript. Not only is QSA tightly integrated with the new architecture, it is also an important foundation of it bringing application customizing to a new level. It allows the user to define and perform batch tasks, to write macros for repeated tasks as well as to define complete custom graphical user interface (GUI) windows.

In addition QSA is also used for defining intelligent „live texts“ as well as when entering element parameters. To facilitate easy generation of those scripts, SIMPACK comes with an integrated scripting development environment (IDE).

ess handling and a set of classes and objects for generating graphical user interfaces (GUI). Furthermore, INTEC has expanded the language to enable the customers to access SIMPACK specific objects and functionality, such as accessing the value pairs of a curve or exporting a plot page to a bitmap file.

## DEVELOPMENT ENVIRONMENT

Generating and testing scripts in a standard ASCII file editor can be a tedious job. To simplify and speed up this task an Integrated Development Environment (IDE) for writing and testing scripts has been included into SIMPACK. This IDE, which is called QSA Workbench, has state of the art productivity features for generating scripts such as syntax highlighting, automatic indentation, bracket check, code completion and automatic parameter list display. The latter two features being extremely useful since they free the scriptor from having to look up each single statement in the documentation and reduce the probability for errors in the script code.

## WHY SCRIPTING?

The new SIMPACK GUI offers a wide range of features for increasing the productivity in day to day work. In addition to that, scripting has been added directly to the new SIMPACK GUI. It enables the user to customize the SIMPACK GUI to specific needs, to add user specific functionality and to automate processes by implementing and executing batch tasks. Typical applications for scripting are user specific ASCII plot curve exporters or a custom GUI dialog for modifying a set of parameters, launching the SIMPACK solver and viewing the results. The functionality made available to the user by integrating the scripting language is explained in this article.

## AUTOMATING YOUR PROCESS

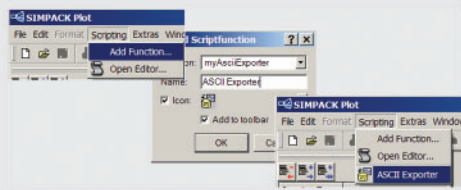
Once simulation models are created and verified quite often complex simulation scenarios and parameter studies are performed. The scripting language allows to automate these processes by enabling the user to execute the SIMPACK solver, start external processes, read and write ASCII files as well as to export the generated results either as ASCII data, as bitmap or as vector graphics files. Of course SIMPACK's QSA workbench can be used for generating and testing these batch scripts. Naturally with a batch script it is possible to run these scripts without the SIMPACK GUI ever appearing on the screen, in the so called silent mode.

## THE LANGUAGE

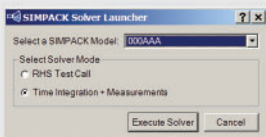
The scripting language used throughout the new SIMPACK GUI architecture is QSA which is an Acronym for QT Script for Applications. QSA is object oriented and has a C/C++ style syntax. It adheres to the international ECMAScript 4.0 standard, the same standard used by Netscape's JavaScript and Microsoft's JScript. In addition to the standard QSA has some application customizing specific functionality such as file and directory handling, proc-

## SCRIPTING CONSOLE

The scripting console is a command line for entering and executing script commands from within the SIMPACK GUI. It enables the user to perform GUI tasks such as sending a plot page to the printer by entering a script com-



*Adding a Macro to the SIMPACK GUI*



*Custom GUI Defined by Script Code for Launching the SIMPACK Solver*

mand rather than using the respective pull down menus or icons. The scripting console comes with the same set of script generation features as the QSA workbench, described above.

### MACROS IN THE SIMPACK GUI

Once the user has generated a specific script, e.g. a custom ASCII exporter which exports curve values in a user specific ASCII format, and has saved this script in the SIMPACK script database it can be reused. One way of reusing the script would be to load it into the QSA Workbench and to execute the script from there. Although this is a good approach while the script is under development, it would be cumbersome to have to do it this way each time a script should be used. Therefore INTEC has introduced the possibility of registering a script to the SIMPACK GUI. Doing so provides access to this script in the Script pull-down menu as well as in the Scripts toolbar where the script can be represented by a user provided icon. Thus a new macro can be added to the SIMPACK GUI.

### GENERATING CUSTOM GUIs

One of the distinct features of the new SIMPACK scripting engine is the possibility to expand the SIMPACK GUI with sophisticated custom GUI windows. The custom GUIs can range from simple dialogs, such as querying the user for selecting a file, to more elaborate dialogs, e.g. a dialog for specifying the export settings of a custom ASCII file exporter. Even complex dialogs, such as a custom GUI, that lets the user select between a set of pre-built simulation models, offers the ability to change parameters, to choose between pre-defined solver options, start the solver and view the results.

### BRINGING LIFE TO YOUR TEXT INPUTS

Texts which the user has entered in the GUI, e.g. an annotation text showing the maximum value of a curve, are typically static texts, which have to be updated manually once the element they are describing changes. The SIMPACK scripting engine can bring life to these

texts making them update according to the property they are describing. To achieve this script code can be embedded into the description text thus making it a „Live Text“.

### SCRIPTING „IN PLACE FORMULAS“

The scripting can even be used in less obvious places, such as when defining the parameters of a SIMPACK modeling element, e.g. a filter. Typically only numbers and Substitution Variables (SubVars) can be assigned to element parameters.

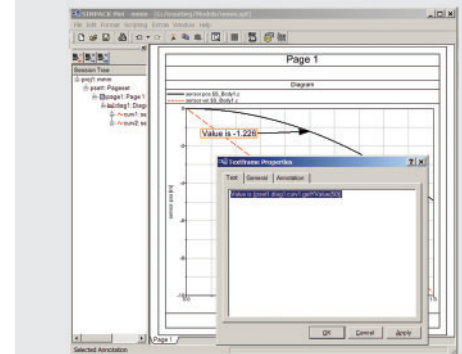
By utilizing the math and data access capabilities of the scripting language SIMPACK now offers the possibility to define and use „in place formulas“ when specifying parameters. Instead of having to manually compute the value of mathematical PI divided by two and then entering the resulting value for the parameter, the user can now simply type „PI/2“. Of course access to SubVars is also available when using these „in place formulas“. Please note that this functionality is currently only available for elements accessible in the new SIMPACK Plot GUI.

### SCRIPTABLE FUTURE

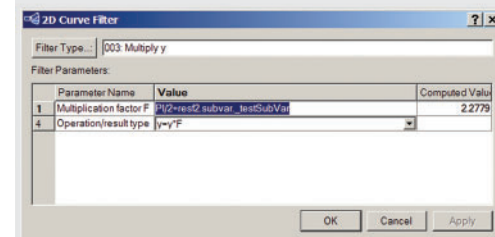
In addition to the discussed features future SIMPACK versions will have an even more enhanced scripting environment. A fully blown script code debugger and a utility for macro recording, i.e. recording user actions and automatically generating script code reflecting those actions, are on the roadmap for scripting. So is extended read and write access to properties of SIMPACK objects, such as the possibility to generate diagrams and curves using the scripting language.

### AVAILABILITY

Scripting and performing batch jobs with the new scripting language have been available since SIMPACK version 8.709. The Scripting Console, Macros, GUI customisation and the Scripting IDE will be available with the next SIMPACK version due in November 2005.



*Annotation Definition by Using „Live Text“*



*Specifying a Filter Parameter by Using „In Place Formulas“*