Add-Ons and GUI Customizations

WAE Design supports tools for customizing the GUI, enabling you to tailor it to meet your specific needs.

Add-On Applications

WAE Design provides basic tools for incorporating add-ons (scripts or executables) into the WAE Design GUI.

- Perl library for editing table files, for example .txt format plan files (see the Plan Files and Tables chapter).
- CLI tools for editing plan files, such as `table_extract`, `table_edit`, and `table_replace`.
- A framework for creating dialog boxes for data entry to the add-on (see the Create Add-Ons section).

Once registered with the WAE Design GUI, add-ons are accessible from the Add-Ons menu. If you do not see an add-on that you expect, select the Add-Ons > Find Add-Ons menu.

Create Add-Ons

Step 1 Create an executable to perform the add-on task. This executable processes specific CLI parameters, which the GUI then uses to pass details of plan files, reports, and options specified through the add-on dialog box. See Construct Add-On Executable.

Step 2 Construct the add-on's configuration file (`addon.txt`), which contains two tables. See Construct addon.txt.

- `<AddOnConfigs>`—Defines the add-on name and description, as well as how the executable should be invoked.
- `<AddOnOptions>`—Defines the WAE Design GUI dialog box from which the executable is invoked from the GUI.

Step 3 Construct the `return-config-file` parameter of the executable to specify the WAE Design GUI behavior upon exiting the add-on. See Construct return-config-file.

Step 4 Register the add-on with the WAE Design GUI, which is putting the executable and `config.txt` file where they can be discovered by WAE Design. See Register Add-Ons.

Construct Add-On Executable

The add-on can invoke any executable file, such as a shell script, Perl script, or binary executable. The executable must recognize the parameters listed in Table 7-1, which are passed to the add-on executable when it is called from the GUI.

All parameters except `-plan-file` are used by the WAE Design GUI to pass a temporary file or directory name to the add-on script. The script then optionally adds the contents to create the desired file or report. For example, you can create an add-on that generates reports on the existing plan file, as well as one that returns a new plan for display in the GUI.

Table 7-1 Parameters Passed by the WAE Design GUI to the Add-On Executable

Parameter	Description
plan-file	Required: The plan file in .pln format that is currently active in the GUI.
out-file	Optional: A file name to which the add-on saves a modified version of the plan file. The WAE Design GUI opens this new plan file after the add-on has completed.
pdf-report-file	Optional: A file name to which the add-on writes a report in PDF format. When exiting the add-on, this report is opened using the default PDF viewer. For information on creating PDF reports, see the Reporting Tools chapter.
report-dir	Optional: A directory name in which the add-on creates a report that is imported into the currently active plan file and opens the report to its first section. The ShowReport value in the <AddOnReturnConfig> table can show this same report. The ShowReport value also overrides the report-dir value. See the Construct return-config-file section. For information on creating reports that can be stored in plan files, see the Reporting Tools chapter.
report-file	Optional: A file name to which the add-on writes a plain text report section. The WAE Design GUI places this report section in an Add-Ons report in the active plan file. This parameter is available for backward compatibility. We recommend you use the report-dir parameter instead. The ShowReport value in the <AddOnReturnConfig> table overrides this property value. See the Construct return-config-file section.
return-config-file	The file name to which the add-on writes the instructions on WAE Design GUI behavior after the add-on has completed. This file contains an <AddOnReturnConfig> table. For information, see theConstruct return-config-file section. This parameter is required if using the report-file parameter.

Error Messages and Warnings

The add-on can generate error messages and warnings by writing them to the standard output to the GUI log window (accessed via the Window > Log Window menu).

- Error messages must start with the word 'Error:' on a new line.
- Warning messages must start with the word 'Warning:' on a new line.

The add-on can inform the WAE Design GUI of its progress by writing the percentage completed to the standard output. This must be a single line containing a number followed by the `'%'` character. The GUI displays this progress in a progress bar while the add-on is running.

A return value of 0 from the add-on indicates that the add-on ran successfully. Any other value indicates an add-on failure.

Construct addon.txt

The `addon.txt` configuration file contains two tables.

- `<AddOnConfigs>`—Defines the add-on name, provides a description of the add-on, and identifies how to invoke the executable should be invoked (Table 7-2).
- `<AddOnOptions>`—Optional: Defines input fields in the add-on dialog box (Table 7-3). A CLI parameter is specified for each field, and these are passed to the executable, together with the input values. The rows are sequentially read.

Table 7-2 <AddOnConfigs>

Property	Value
Description	Description displayed in the dialog box.
Name	Name displayed in the dialog box. The name displayed in the Add-Ons menu is a subdirectory name.
PlanFormat	This value determines the type of column information are provided to the add-on. The advantage to using .txt or .db is that you do not have to extract the derived information using other means, such as `table_extract`. Default = pln; only plan file column information Options = txt, db; plan file and derived column information
Run	Required: The command used to execute the add-on.
Version	Version number displayed in the dialog box.
WindowsRun	The command used to execute the add-on under Windows, if different than the Run parameter.

Table 7-3 <AddOnOptions> Table

Column	Description
Tab	Create tabs and group options within them. • If left empty, the tab is unnamed. If only an unnamed tab is present, it is not displayed and instead the options are shown in a dialog box without tabs. • If the Tab column has the same value as a previously named Tab value, the option is added to the same tab, and is placed below the previous option. If the value is different, a new tab is created and the option is placed in that tab.
Group	Create groups and place options within them. • If there is no Group value, an option is not grouped, and if applicable, is placed below the last ungrouped option. • If the Group value is new, a new group is created within its associated tab. If that row does not have a Tab value, the group is created within an unnamed tab. • If the Group column has the same value as a previously named Group value, the option is added to that same group and is placed below the previous option provided they have the same Tab value.

Column	Description
OptionName	The name of the command-line option; this is prepended with a -(dash) when passed to the executable. The following names are reserved and cannot be used. • mate-version • out-file • pdf-report-file • plan-file • report-dir • report-file • return-config-file
Prompt	Text string that prompts for the desired value.
Type	Type of entry. • Bool—Drop-down list with True and False selections. Passes “true” or “false” to the script. • CheckBox—Toggle entry; if selected, it is true, and if not selected, it is false. • ComboBox—Selection of items to appear in a drop-down menu defined by a semicolon-delimited set of options. The user can either select from the menu or type an entry explicitly. • DropDown—Selection of items to appear in a drop-down menu defined by a semicolon-delimited set of options. The user can select from the menu. • Float—Floating point number. Range is expressed as x;y. • Integer—Integer. Range is expressed as x;y. • Password—Text string. Characters typed by the user are hidden. • OpenDir—Directory name. Same as OpenFile but for directory names. • OpenFile—Name of an existing file. Default specifies the file name (no path) in the current directory. Can browse with a browse button to a different directory. The complete path is passed to the script. • SaveFile—File name. Same as OpenFile, except it can create a file. • String—Text string. Range is ignored. • TableSelection—Drop-down list for the objects in a specified table. For example, if you specify TableSelection in the Type column and Interfaces in the Table column, the drop-down list gives you standard WAE Design choices of selected, all, none, or tags. The Range and Default columns are ignored if this type is specified. • TableEntry— Drop-down list containing the information for traffic levels, service classes, or interface queues. For example, if you have multiple traffic levels in the plan file, this drop-down list enables you to choose one of the available traffic levels. Specify the list in the Table column. The selected item is passed to the script as an argument for the option in the OptionName column.

Column	Description
Table	The table for the TableSelection type or the TableEntry type. • TableSelection options can be any table that is accessible from the WAE Design GUI using the View > Tables > Show/Hide Tables menu except for the Shortest Paths table. • TableEntry is either Queues, ServiceClasses, or TrafficLevels.
Range	Varies depending on the Type identified.
Default	Varies depending on the Type identified.

Construct return-config-file

The file named by the `return-config-file` parameter stores the return configuration information in the `<AddOnReturnConfig>` table (Table 7-4). This table contains a pair of columns, Property and Value, where the values define the WAE Design GUI behavior upon exiting the add-on. For information on the GUI behavior of views, traffic levels, service classes, and queues see the *WAE Design User Guide*. For information on plot layouts and traffic utilization colors, see the *WAE Network Visualization Guide*.

Table 7-4 <AddOnReturnConfig> Properties and Values

Property	Value
OutputFileName	Output (resulting) plan file name. The string can contain a $1 variable, which is the name of the input plan file (without file extension). The default is $1-out. **Example:** If the input plan file name is euro_geo.pln, by default the output file name is euro_geo-out.pln.
PlotLayout	The GUI displays the specified layout view. If not specified, the GUI displays the Default layout, which is customary for a newly opened plan file.
Queue	The GUI displays traffic for the specified queue. If both ServiceClass and Queue are specified, the ServiceClass value has priority.
ServiceClass	The GUI displays traffic for the specified service class. If not specified, the GUI displays undifferentiated traffic.
ShowReport	The GUI opens the report, including the report-dir report, which should be inserted into the plan before this rule is applied.
ShowReportSection	The report opened by ShowReport opens to this section. If not specified, the report opens to the first section.
TrafficLevel	The GUI displays this traffic level. If not specified, the GUI displays the Default traffic level, which is customary for a newly opened plan file.
UtilColorList	The GUI fills interfaces with these colors to show levels of traffic utilization. If not specified, the default threshold settings and colors are used. Following is the format, where threshold is a real number and colors are of the form #RRGGBB (standard HTML color names). Format: threshold%<=color<=threshold% **Example:** 0% <= red <= 25%, 25% <=green<=50%, 50%<=purple<= 75%, 75%<=blue<=100%

Property	Value
UserView	The GUI displays this user-defined plot view. If both View and UserView are specified, the View value has priority. For information on creating plot views, see the Customized Network Plot Views section.
View	The GUI displays one of these views. If not specified, the GUI displays whichever view is appropriate for the output plan file. • FailureImpact—Failure Impact view • MeasUtil—Measured Traffic view • SimLSPRsrv—LSP Reservations view • SimUtil—Simulated Traffic view • SimWCUtil—Worst-Case Traffic view

Register Add-Ons

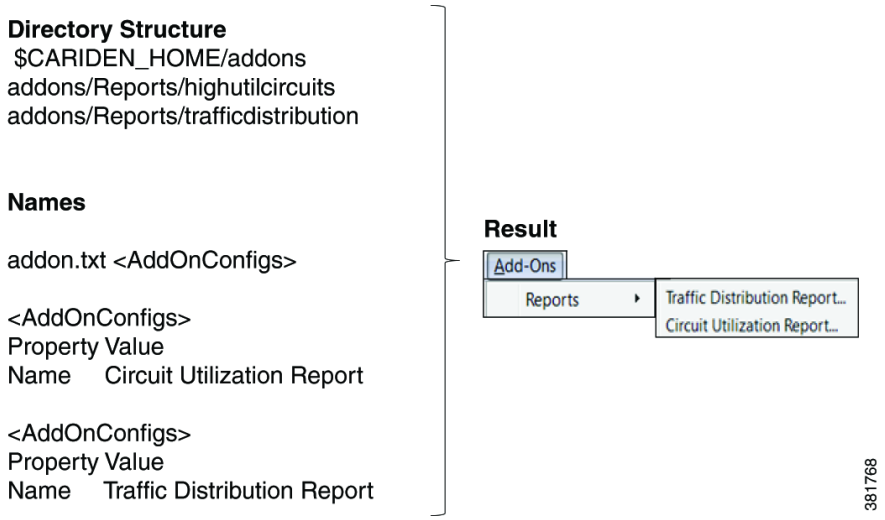
To register an add-on with WAE Design, create a subdirectory for the add-on in the `$CARIDEN_HOME/addons` directory. At a minimum, this directory must contain the `addon.txt` configuration file that defines the dialog box and information about the executable.

WAE Design adds a menu item in the Add-Ons menu using the Name parameter in the <AddOnConfigs> table. If this parameter is omitted, WAE Design uses the name of the subdirectory.

To create multiple levels of sub-menus in the WAE Design GUI, simply create nested subdirectories. See Figure 7-1 for an example.

Add-ons and their sub-menus appear in the Add-Ons menu the next time you launch the GUI, or after you select the Add-Ons > Find Add-Ons menu.

Figure 7-1 Example Add-On Submenus



Example Add-Ons

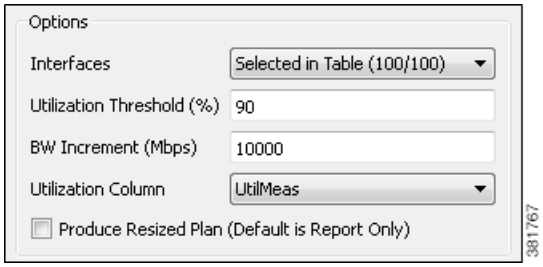
WAE Design includes three example add-ons: Circuit Upgrade, Circuit Utilization Report, and Traffic Distribution Report.

Circuit Upgrade Add-On

The `$CARIDEN_HOME/addons/CircuitUpgrade` directory includes a sample add-on named CircuitUpgrade that analyzes the Interfaces table Utilization columns. Upon exiting the add-on dialog box (Figure 7-2), the add-on executable does the following.

- For every interface whose utilization (in the specified utilization column) exceeds a threshold, the circuit capacity is increased by the specified increment until the condition is met.
- A report named *Circuit Upgrade* is generated. The report contains only one section, which is named *List of Upgrades*.
- The WAE Design GUI opens a new plan file using the same name with -upgrade added as a suffix.

Figure 7-2 Example Circuit Upgrade Dialog Box



Tab and Group Customizations

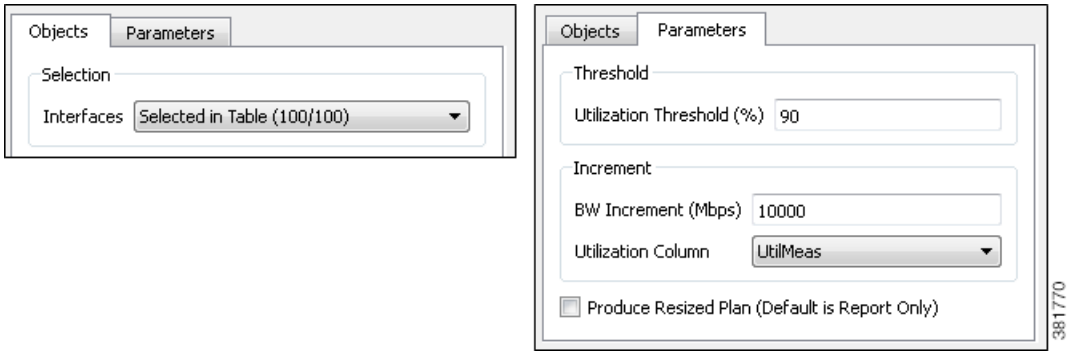
This example <AddOnOptions> table customizes the default sample addon.txt file as follows (Figure 7-3).

- Creates two tabs: Objects and Parameters
- In the Objects tab, creates a group named Selection.
- In the Parameters tab, creates two groups: Threshold and Increment. Another available selection is created to give the option to produce the resized plan file is left ungrouped. A resized plan is an output plan that identifies which circuits need to be upgraded; they are resized based on the criteria set. If not used, you only receive a report detailing the actions WAE Design suggests making.

<AddOnOptions>

Tab	Group	OptionName	Prompt	Type	Default	Range	Table
Objects	Selection	interfaces	interfaces	TableSelection			Interfaces
Parameters	Threshold	thresh	Utilization Threshold (%)	Float	90	0;100	
Parameters	Increment	increment	BW Increment (Mbps)	Float	10000		
Parameters	Increment	col	Utilization Column	DropDown	UtilMeas	UtilMeas;Util-Sim;WCUtil	
Parameters		resize	Produce Resized Plan (Default is Report Only)	CheckBox	FALSE		

Figure 7-3 Example Customizable Tabs and Groups



Reports Add-Ons

The `$CARIDEN_HOME/addons/Reports` directory contains two sub-directories, each containing an add-on, thus demonstrating how to create sub-menus. Each of these add-ons, Circuit Utilization Report and Traffic Distribution Report, demonstrate two key add-on features.

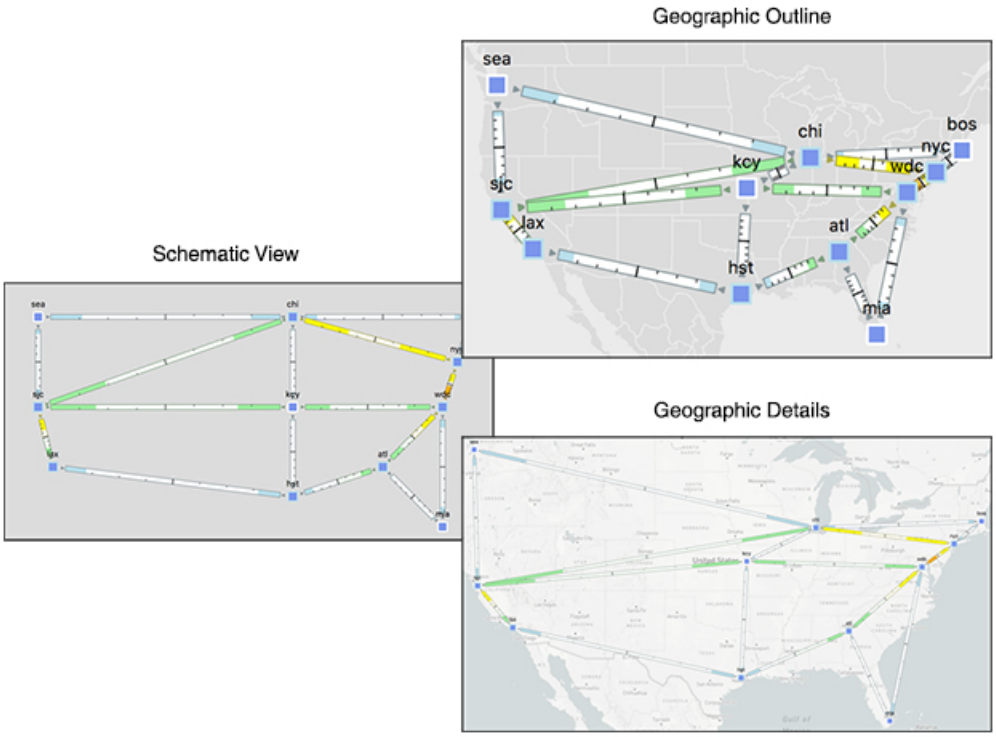
- Passing information you input to the `mate_jasper` tool to generate a report based on the current plan file, and opening the resulting PDF report. Each report is based on the .jrxml JasperReports template file input to `mate_jasper`. For information on using `mate_jasper`, see the Reporting Tools chapter and `mate_jasper` Help output.
- Using derived column information for the resulting report.

Note To run these example report add-ons, you must have ActivePerl installed if you are running a Windows operating system. (ActivePerl is an open-source Perl scripting language provided by ActiveState.)

Customized Network Plot Views

Map Server

WAE Design uses an online detailed geographic map database to draw the detailed map background, which is one of the canvas plot options available in the WAE Design GUI. For information on viewing a detailed background map and changing plot options, see the *WAE Network Visualization Guide*.



Static Backgrounds

By editing the plan file directly (for example, through an add-on), you can add a user-defined static view to display interfaces with customized color fills and fill percentages. The view then appears as an option in the Network Plot drop-down list located in the far left corner of the visualization toolbar.

To create this custom network plot view, create or edit a `<PlotViewInterfaces>` table (Table 7-5). You must have an entry (row) for each interface you want you want color filled, and each of these interfaces must have the same PlotView name. For example, Figure 7-4 shows four interfaces in the user-defined plot view named Affinities.

Table 7-5 <PlotViewInterfaces> Columns

Column	Description
PlotView	Name of the plot view; this name appears in the Network Plot drop-down list.
Node	Node containing the interfaces.

Column	Description
Interface	Interface name. **Example:** POS1/1/1
FillPercent	The percent of the interface to be filled with color.
FillBound	Used only if the FillPercent color represents a utilization percentage. FillBound is the maximum utilization percentage possible, or the QoS bound. The remainder of the interface is gray. The default is 100%. • FillBound less FillPercent is colored white. For example, on cr1.chi to cr2.nyc in Figure 7-4, the FillBound is 80 and the FillPercent is 50, thus filling 30 percent of the interface with white. • If the FillBound is greater than the FillPercent, then the difference is filled with red and white stripe, which represents a QoS violation. For example, on the cr2.hst to cr1.lax interface in Figure 7-4, the FillBound is 25 and the Fill-Percent is 75, thus filling 50% of the interface as a QoS violation.

Figure 7-4 Example <PlotViewInterfaces> Table Defining an Affinities View and Coloring Five Interfaces

<PlotViewInterfaces>					
PlotView	Node	Interface	Color	FillPercent	FillBound
Affinities	cr1.lax	{to_cr2.hst}	Green	25	100
Affinities	cr2.hst	{to_cr1.lax}	Glue	75	25
Affinities	cr1.chi	{to_cr2.nyc}	Red	50	80
Affinities	cr1.chi	{to_cr1.sea}	Yellow	25	80
Affinities	cr1.kcy	{to_cr1.sjc}	Orange	75	100