# Release summary - CA Agile Requirements Designer 2.3

The CA Agile Requirements Designer (ARD) development team work constantly to identify and develop new features, while enhancing the core functionality to meet end user needs. 2.3 builds on version 2.2, which launched a revamped HPE ALM integration the ACE Editor, automation mapping, subflow mapping configuration, and the XML and JSON importer. You can find out about the latest enhancements below.

## Contents

# The CA Service Virtualization Integration

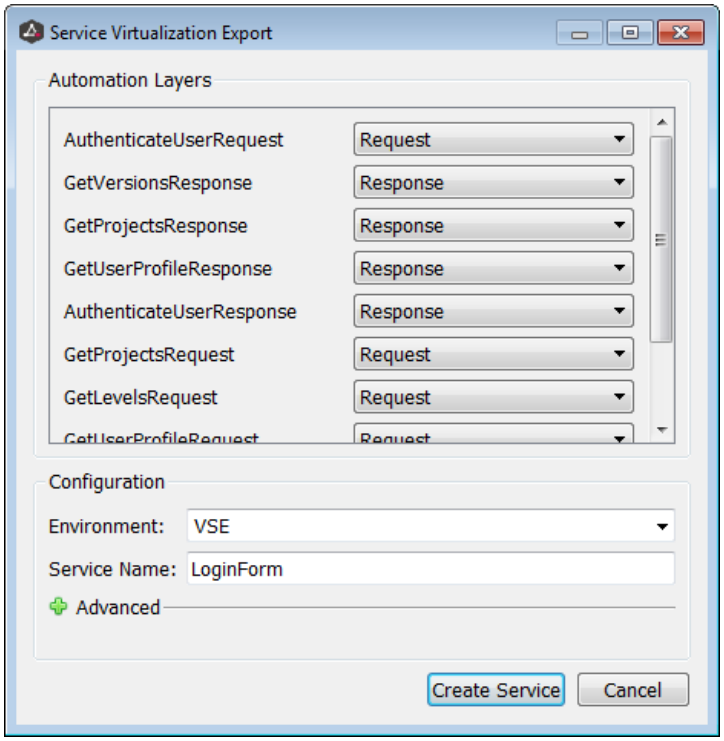
The ability to quickly generate rich data for service virtualization was introduced with version 2.0. A service or message can be modelled as a flowchart model, with a corresponding configuration file setting out code snippets. The snippets are then compiled by the path explorer, creating a set of data which covers every possible combination of Request and Response.

Version 2.2 accelerated this process, as JSON and XML can now be imported and decomposed into a rudimentary flowchart and configuration file automatically. This approach therefore helps eliminate the time spent manually defining complex data scenarios, allowing production-like systems in which to execute every test to be quickly spun up. To see how this approach works, watch this video.

Version 2.3 provides direct support for CA Service Virtualization, allowing Request Response Pairs generated in CA Agile Requirements Designer to be published directly to CA Service Virtualization. The creation of a new virtual service can also be triggered from within CA Agile Requirements Designer.

There is a new CA Service Virtualization option available by clicking the export button  in the automation tab of the view test cases section of the Path Explorer. Having entered the URL for the service and valid credentials to log in, you can then specify the virtual data to be exported. You can specify which layers within a multi-layer configuration file will be used to create the Requests, and which to create the Responses. There are also a range of “advanced” options, which reflect the options within CA Service Virtualization, including groupings, classes and SSLs. CA Agile Requirements Designer will provide the host URL needed to execute your tests against the virtual service once the export is complete.

A complete set of accurate virtual data can now be defined, created and maintained from a flowchart model, while triggering the virtual services needed for rigorous testing from within CA Agile Requirements Designer itself.




*Defining which automation layers will generate Requests, and which will generate Responses.*

# Native data pool editing and creation

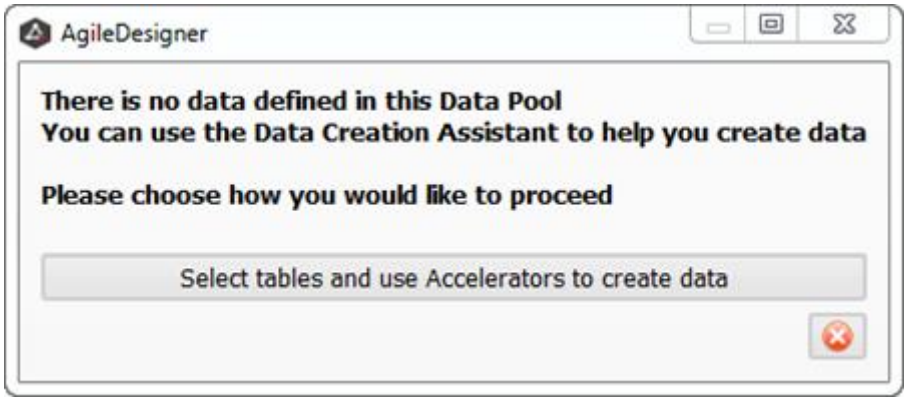
Data pool editing and creation are now built into the repository connector dock. It is now possible to define, create or publish data pools, as well as edit, add or remove tables from them. This enables a more seamless integration with CA Test Data Manager, with greater bi-connectivity, making it easier to find or make data at the same time as optimized tests are edited and created.

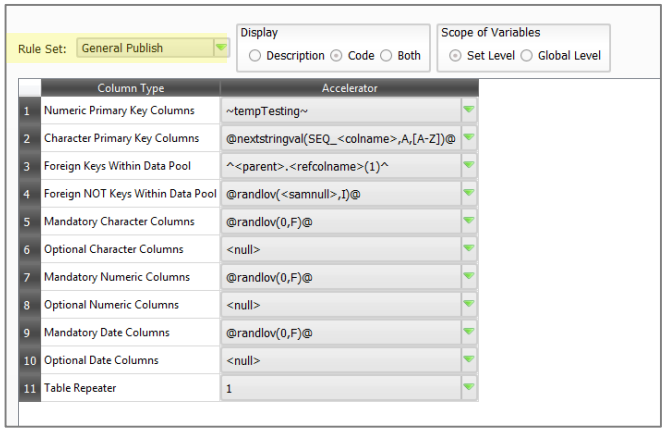
## Create new data pools

Child data pools can be created from within CA Agile Requirements Designer, in addition to data sets and data pools. This is as simple as right clicking on any item above the lowest item, and selecting “create child level”.

## Editing data in data pools

Data can be created in empty pools, while existing data in pools can also be edited. This is performed in the same way as within CA Test Data Manager, with both manual data editing and the use of accelerators. By right clicking on a data pool and clicking “edit data”, the two options will be displayed.



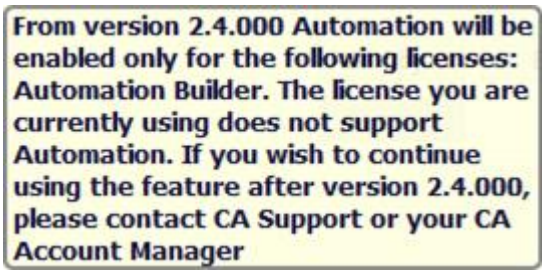




The rules in the accelerator are stored as XML files under *C:\Program Files\Grid-Tools\AgileDesigner\DmXmlAccelerators* . They are fully customizable. Five default accelerators exist, which can be used as templates for creating new rules. Applying an accelerator to create or edit data in a data pool is as simple as selecting a column and choosing a rule from the drop-down menu. The next time you click “edit data”, the rule will be displayed in the line edit view, where the column definitions can also be edited using the data painter.

# Test automation

The ability to generate automated test scripts using a re-usable configuration file was introduced with version 1.9. The creates a flexible approach to create automated tests directly from a flowchart, while maximizing test coverage and avoiding repetitious, linear manual scripting and maintenance. For an overview of this approach, please watch this video.

## License changes



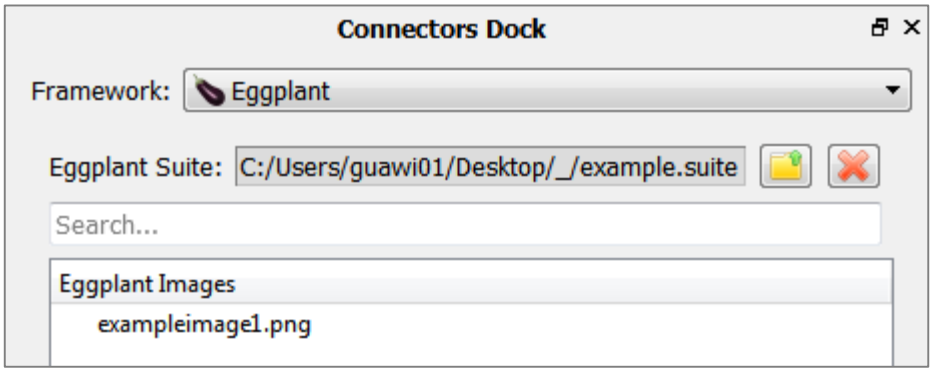

*You will see this warning displayed in version 2.3 if you are likely to need the new license from version 2.4 on.*

The ability to generate automated test scripts in CA Agile Requirements Designer is being migrated to a separate licensing scheme. Starting with version 2.4, you will need the Automation Builder license. Please contact your account manager to ensure that you have the licencing you will need in future. If you do not have the Automation Builder license, a warning will be displayed in the set-up wizard from version 2.3 on, and when utilising automation functionality.

## Eggplant

An integration has been developed specifically with automation framework Eggplant, to reflect the character and image recognition it uses to identify objects. In addition to Ranorex, Eggplant objects can also now be browsed from the connectors dock.

Images and text stored in Eggplant project files will appear in the connector dock in CA Agile Requirements Designer when a project file is opened. From there, they can be dragged, dropped and modelled. This creates a layer of abstraction above the automated code snippets, meaning that a model of Eggplant objects can be used to generate automated code snippets.




*An image, taken from a .suite file opened in the Connector Dock.*

As the objects are dragged and dropped, actions are assigned to them. Each action has a corresponding code snippet in the automation configuration file. CA Agile Requirements Designer will then compile the snippets into a set of automated test scripts, complete with the object names needed by Eggplant to execute the tests. Automated code can also be assigned within the automation tab of the block properties window, where the Eggplant objects will appear under the auto-complete prompts.

# Integration enhancements

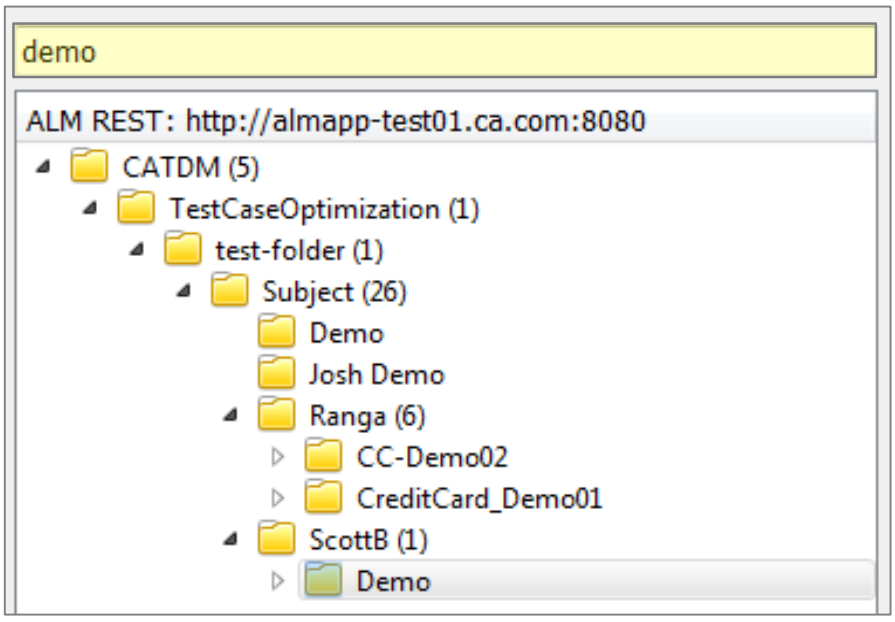
Version 2.3 builds on the already extensive range of integrations offered by CA Agile Requirements Designer, from project management tools, to automation frameworks and requirements gathering platforms. The usability and ease of connectivity have been enhanced in this release, with a particular focus on the connector dock, importing and exporting.

## The connector dock

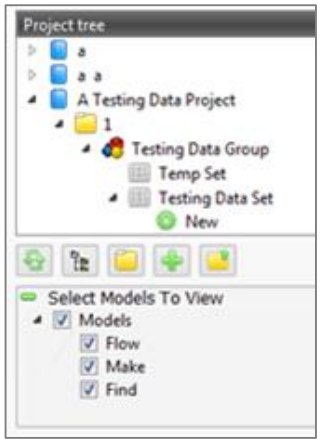
The connector dock has been enhanced, making it easier to find any relevant assets within existing projects, quickly dragging and dropping them to your flowchart model.

### The search box

A new search box means that artefacts can quickly be found and dragged and dropped to the flowchart. The search box will reduce down the number of items shown in the tree structure in real-time, quickly querying projects from within CA Agile Requirements Designer. JIRA issues and HPE ALM test cases, for instance, can be searched for quickly by name. The search functionality extends to the repository as well, to quickly find shared assets created in CA Agile Requirements Designer or CA Test Data Manager.




### The repository connection tree

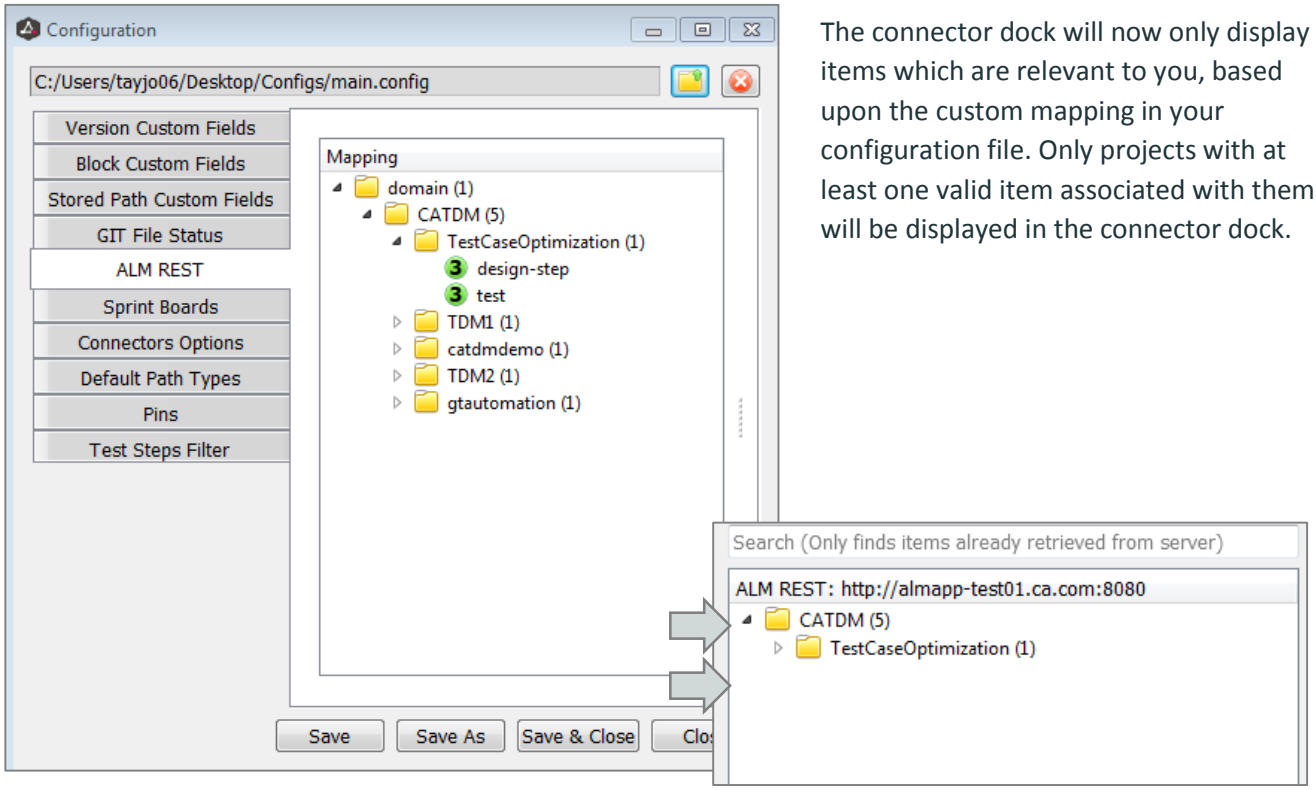



The repository connection tree now displays all the level and object types found in CA Test Data Manager, under the same tree structure.

Filters now also exist at the bottom of the connector box, enabling you to quickly browse items when connected into the CA Test Data Manager repository. The items displayed can quickly be refined by selecting from three modes to view: flows, make and find. The flow option will display .VTF files, the make option will show data pools, and the find option will show test match data pools.

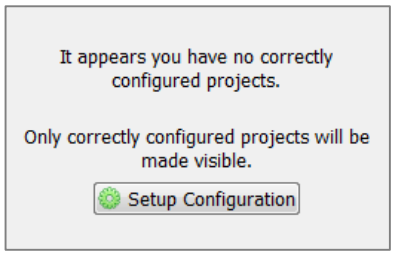
Any filters applied will be displayed wherever in CA Agile Requirements Designer the repository connector tree is shown.

## Only relevant items are displayed in the connector dock




The connector dock will now only display items which are relevant to you, based upon the custom mapping in your configuration file. Only projects with at least one valid item associated with them will be displayed in the connector dock.

## Configuration file prompt



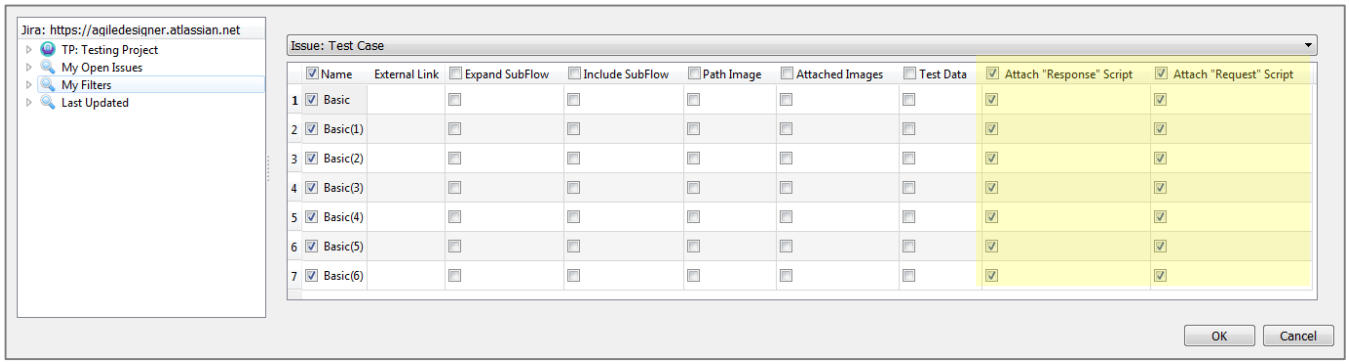

CA Agile Requirements Designer now identifies when there is a valid configuration file for a given connector or not. If there is no valid configuration, and you cannot therefore import or export assets, you will now be prompted to create a configuration file for the connector you are using.

## Exporting

Version 2.3 extends the ability to export test assets to connected tools. The flexibility of exports had already been substantially increased with version 2.2 and custom mapping now allows users to configure where attributes and assets stored in CA Agile Requirements Designer will be stored in a connected tool.

### Export automation scripts

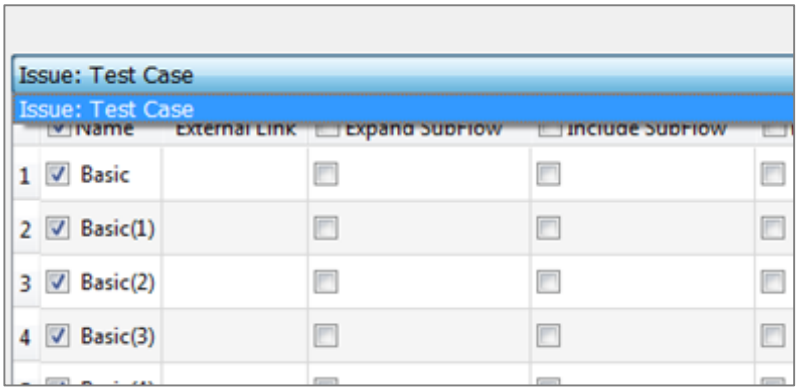
Automated test scripts can now be attached when exporting test cases to project management tools and development platforms, including JIRA, HPE ALM and GIT.




*Exporting XML scripts to issues stored in JIRA.*

Whichever layer is selected from a multi-layer configuration can be exported, and this includes virtual data such as XML. The test data and virtual data needed to automatically execute an optimized set of test cases can therefore be exported along with the automated test itself. Traceability is thereby retained at both the flowchart and test case level, while bi-connectivity means that all assets can be maintained at the same time from the flowchart, providing testers with an up-to-date test pack in one place.

### Only relevant export types are displayed

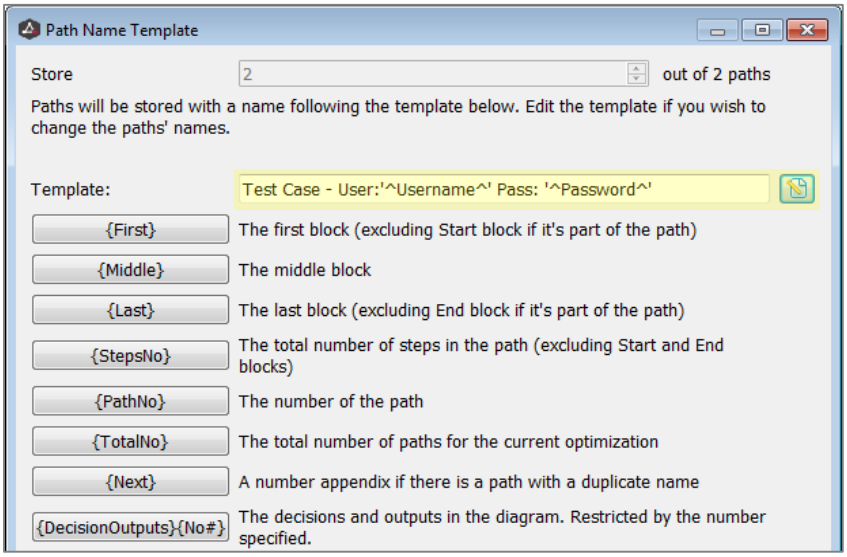



CA Agile Requirements Designer will now analyze your custom mapping and configuration, only allowing users to choose from export types which are both relevant and valid. If, for instance, test cases have been mapped to issues in JIRA in then configuration, then the option to export test cases will be given when exporting when there are also stored test cases in CA Agile Requirements Designer.

## Resolvable path names

Path naming has now been updated, allowing you to quickly and easily bulk name tests based on the data contained in them.

When storing paths, the new  button will open the path explorer. There, you can define the test case names as resolvable data functions, based on the outputs of the blocks. By checking the new “Resolve path name using path data” option, the stored tests will be stored based on the data variables set out in the flowchart.




*A resolvable naming expression, created in the data painter.*

## Global editing of subflow optimization techniques

The optimization technique used to generate paths through subflows can now be set globally. Navigating to “Open / Change Subflow Optimization” under the tools menu will open the following window, where the optimization technique can be set for each subflow contained in the master flow: