

Graphics Reactor

Introduction

In the iteraplan Graphics Reactor, every iteraplan user can generate up-to-date customer-defined diagrams and reports. A user can create new diagrams and reports through a script file. He can upload it to the iteraplan server and make it available to all users. The script can process the full EAM landscape data, that is all building blocks and connections in the format that is also available as XML export (Mass Data -> Export/Import in the iteraplan classic client).

The user can define the structure of the result: it may be an enhanced diagram based on well-known visualisations (for example landscape diagram, masterplan), or a completely new type of diagram specific to the needs of the user. Also textual or tabular reports are possible. The technical format of the result is typically GraphML, SVG or HTML, but the Graphics Reactor is not technically limited to that.

The Graphics Reactor works by using **Extensible Stylesheet Language Transformations**. Executing a **XSLT**-file performs the transformation and creates one or more output files, which can be downloaded. Any other kind of files can also be uploaded, downloaded or deleted within the Graphics Reactor.

There is the possibility to automatically **execute a list of saved queries prior to the script execution**. The configured saved queries will be executed in a given order. The resulting building block ids can be referenced by the XSL/XSLT script.

Via the Graphics Reactor there is also the possibility to configure a URL to **publish a result output file**. A specific URL will be available to download an output file of a script file. You can configure the roles that have access to the published output file.

Using Graphics Reactor

The screenshot displays the Graphics Reactor web interface. At the top, there are four numbered icons: 1 (refresh), 2 (grid view), 3 (upload), and 4 (search). The main area shows a file list with columns for Name, Size, and Date. A context menu is open over the file 'publicDemo_fullModLinks.xsl', showing options: Execute File, Download, and Delete. To the right, there are two panels: 'Script Configuration' and 'Current Reactor Status'.

Script Configuration
/publicDemo_fullModLinks.xsl

Execute the following saved queries prior to the script
Saved Query Ids 1,5,2

Published output file
Output Filename publicDemo_strategicBuildingBlock.html
URL /published/strategicBuildingBlock
Access Roles admin,user
Cache Time 10 hours

EDIT

Current Reactor Status

08/08/2016, 11:21:10 AM Script "/publicDemo_fullModLinks.xsl" execution started.
08/08/2016, 11:21:13 AM Script "/publicDemo_fullModLinks.xsl" execution successfully completed.

CLEAR LOG EXPORT

XSLT parser: Saxon-HE-9.6.0-5

Change View Style

Click the icon button with number 1 to change the view of folder contents between icon and list view.

Create Folder

Click the icon button with number 2 to open a dialog to create a folder in the currently open folder.

Upload a File

Click the icon button with number 3 to open a dialog to upload a file to the currently open folder.

Search in the folder

Click the icon button with number 4 to open the search bar containing an input field. Enter a part of a folder or file name to search for it in the currently shown folder. Click the icon button again to close the search bar.

Delete a File or Folder

Right-click a file/folder and select delete.

Download a File

Right-click a file or folder and select download.

Execute a Script

Right-click the script you want to execute and select execute in the context menu. After the execution the file browser will be refreshed automatically and you can see the generated files in the folder where the executed script can be found. Additionally you get shown the status of the reactor in the box on the bottom right. The status box will be updated with reactor status messages when a script has been started and when it has been executed.

Edit the Configuration of a Script

If you select a script in the file browser the script configuration box on the top right will display the script configuration.

Click the edit button to open a dialog where you can configure idenpendently two configuration details:

Execute Saved Queries prior to the Script

Enter a comma separated list of ids of saved queries to execute them automatically prior to the script execution. The ids of the saved queries can be found in the classic client in the list of saved queries (Visualisations -> All saved queries).

Configured saved queries will automatically be executed in the given order prior to the script execution. Results (list of ids) will be placed in a number of XML files named after their position in the list. The resulting ids of the first saved query will be stored in a file named 1.xml, the results from the second query will be stored in a file called 2.xml and so on. You can then reference the resulting XML files in your XSL/XSLT script.

The XML files with the list of IDs will have the following format:

```
<?xml version="1.0" encoding="UTF-8"?>
<ids>
<id>6</id>
<id>17</id>
</ids>
```

Publish an Output File

Enter configuration details that are needed to get a result file published via URL:

- Output Filename: Path to the file that should be published
- Published URL: The URL name via that the result file gets published. Typically the output filename, but you can choose a different one. The URL name must not include a path, it is relative to the iteraplan instance path plus "/published/".
- Access Roles: A comma separated list of roles that have the right to access the URL. See [Users, Roles and Permissions](#).
- Cache Time: Is this cache time (in hours) exceeded - calculated from the last script run - the script will be run again and the new result file will be downloaded. Otherwise the existing result file will be immediately downloaded.

Permissions

A user must be logged in and have a role with both the permissions "Access iteraplan via REST API" and "Use of the Graphics Reactor" in order to access it.



- Make sure your transformation files are compatible with currently used XSL parser ([Saxon-HE-9.6.0-5](#)).
- All users work in the same Graphics Reactor file area. In case of simultaneous script executions (by more than one user), the executions will be handled in accordance with the First-In-First-Out principle.
- Make sure all uploaded files are UTF-8 encoded.