

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 relations in the format that is also available as XML export (Report -> Export/Import).

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.

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 interface. At the top, a navigation bar contains seven numbered icons (1-7) for file operations. The main area is a file browser showing a folder named 'BuildingBlockTypes' and several files: 'BuildingBlo...', 'iteraplan.e...', 'iteraplan.x...', 'iteraplan.x...', 'publicDemo_...', 'publicDemo_...', and 'README.txt'. An 'EXECUTE' button is visible at the bottom right of the file browser. To the right, a 'Script Configuration' panel for '/publicDemo_htmlTable.xsl' shows settings for 'Published Output File', 'Output Filename', 'URL', 'Access Roles', and 'Cache Time'. Below the configuration is an 'EDIT' button. At the bottom, a 'Console' window shows the execution log: '04/11/2019 09:11:44 AM Reactor ready', '04/11/2019 09:12:05 AM Script /publicDemo_htmlTable.xsl execution started.', and '04/11/2019 09:12:07 AM Script "/publicDemo_htmlTable.xsl" execution successfully completed.'. Below the console are 'CLEAR LOG' and 'EXPORT' buttons. At the very bottom, it says 'XSLT parser: Saxon-HE-9.6.0-5'.

Upload a File

Click the upload-icon (1) to open a dialog to upload a file to the currently open folder.

Download a File

Choose a file and click the download-icon (2) to download item.

Delete a File or Folder

Choose a file/folder and click the trash can-icon (3) to delete item.

Create Folder

Click the folder-icon (4) to open a dialog to create a folder in the currently open folder.

Refresh a File Browser

Click on the arrows (5) to refresh a file browser.

Change View Style

Click the icon (6) to change the view of folder contents between icon and list view.

Search in the folder

Click the search-icon (7) 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.

Execute a Script

Choose the script you want to execute and click on the execute button (8). 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 the status of the reactor is displayed in the console at the bottom. The console 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 the following configuration details:

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 by which the result file gets published. Typically this is equal to 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.