

Portal Actions Import Tool

Version 1.0.0 (9.7.2014)

Contents

1 Overview	2
2 Compatibility	2
3 Running	2
4 Configuration	2
5 XML data file description	3
6 Embedded data mode	3

1 Overview

Portal Actions Import Tools is for importing QPR Portal actions through QPR Web Service. Both creating new and updating existing actions are possible. Also embedded attachment files can be saved. Running of the tool may be automated. The tool accepts **XML** (XML Spreadsheet 2003) and **CSV data files**. The tool is implemented with .Net 4.0. The tool must be installed in a workstation.

The import tool either creates a new action or updates an existing one, based on an **identifier attribute**. If there is a matching action already in the QPR system, that action is updated; otherwise a new action is created. The tool does not delete any actions (deleting actions is best done using QPR Web Service tester).

Configuration file is as an XML file, which is provided as a parameter for the tool run command. (Please don't confuse XML data file and XML configuration file.)

The tool writes log file **ImportPortalActionsApplication.txt** stored QPR log files folder. Starting of the run, ending, creating actions, and updating actions are logged.

2 Compatibility

Compatible with QPR 2014.1, QPR 2012.2.

3 Running

The import is run using following command (in the command line, cmd.exe) in the tool installation folder:

ImportPortalAction [xml configuration file path]

4 Configuration

Configuration is defined in an xml file (see related example configurations). The root tag is "importportalactions". Following child tags are used:

Parameter	Description
username	QPR username.
password	Password the QPR user.
webservicelocation	QPR Web Service location, such as http://HOSTNAME:9002/QPR2014-1/Portal/QPR.Isapi.dll/wsforward/MainService.svc/wsHttp
wcfsecuritymode	Either None or Message corresponding to WCF binding's security mode. Usually security mode is Message where LDAP is used by QPR.
qprauthenticationmode	Options: <ul style="list-style-type: none"> - QPR: defined username and password are used - Windows: Windows user account running the import tool is used to log in to QPR <p>If authentication mode is Windows, no username or password need to be defined. Note that if authentication mode is Windows, wcfsecuritymode must</p>

	be Message.
actiontype	Imported action type name (e.g. Comment, Lesson).
datalocation	Path for the XML or CSV data file.
qprconfigurationfilelocation	Path for QPR_Servers.ini.
setattributeoptions	Options parameter for QPR Web Services SetAttribute operation. SetAttribute is used to update attributes of existing actions.
createobjectoptions	Options parameter for QPR Web Services CreateObject operation. CreateObject is used to create new actions.
identifierattribute	Action's web service attribute which is working as an identifier. Note that the identifier must be unique.
attachmentpath	Folder where attachment files are stored.
dataformat	Either xml or csv depending on the data file format.
embeddeddatamode	Either true or false depending on whether embedded data mode is used.
columnseparator	Characters separating columns in csv file.
skiperrors	Either true or false . When true, continues import after encountering web service errors. When false, stops on all errors.

5 XML data file description

XML format read by the tool is **XML Spreadsheet 2003** (file extension is .xml). Files of that type can be saved in MS Excel. This xml format contains e.g. following tags: Workbook, Worksheet, Table, Row, Cell, Data. Xml format is used when configuration parameter is set **dataformat=xml**.

6 Embedded data mode

Purpose of the **embedded data mode** is to import embedded attachment files of actions. Embedded data mode is activated using the *embeddeddatamode* configuration parameter. Actions and their embedded attachments are imported in separate runs (and different configuration files are needed).

File data contents can be in one column in the data file, or as separate files in the file system.

In case of in a column, following columns are needed:

- **filecontents**: data itself in hexadecimals e.g. 0x446F63756D65...
- **filelogicalname**: file name in QPR
- **mimetype**: file mimetype

Also *identifier* needs to be defined because it determines the action where the attachment is stored

In case file data contents is read from separate files in the file system, following columns are needed:

- **filephysicalname**: file name in the operating system
- **filelogicalname**: file name in QPR
- **mimetype**: file mimetype

Also *attachmentpath* needs to be defined.