

QPR Suite ETL Task Workbench

Version 1.4.0 (26.9.2016)

Contents

1 Overview	1
2 Installation and running.....	1
3 Configuration file.....	2
4 Integration tasks.....	3
4.1 Export embedded files	3
4.2 Export to Excel (not implemented)	4
4.3 Import embedded files (not implemented)	4
4.4 Import measure values	5
4.5 Modelling client create or update.....	6
4.6 Modelling client delete	7
4.7 Perform object operation	7
4.8 Synchronize metrics model	8
4.9 Web services create copy (not implemented)	8
4.10 Web services create or update	9
4.11 Web services delete.....	10

1 Overview

QPR Suite ETL Task Workbench is a tool for setting up integrations where data is extracted from or stored to a QPR Suite system. The tool has variety of use cases, one of which is transferring data between QPR EA and QPR Metrics.

QPR Suite ETL Task Workbench can be configured to use both QPR Web Services and QPR API interfaces for extracting and loading data. An expression language is included in the platform to enable more flexible data transformations. Still, writing full procedural programming logic is not possible. For more information about the expression language, see QPR Reporting Add-on documentation:

<https://community.qpr.com/node/2198>.

The tool implements ETL (extract, transform, load) process steps as follows:

- **extract** is made using Web Service QueryObjects
- **transform** is made using expressions, and
- **load** is made using Web Service CreateObject, SetAttribute and DeleteObject.

The tool is compatible with QPR Suite 2016.1, QPR Suite 2015.1, QPR Suite 2014.1, QPR Suite 2012.2, QPR Suite 2012.1.

2 Installation and running

QPR Suite ETL Task Workbench is run using the command line.

To install:

1. Copy the files in **Binaries** folder to any location in the computer where the integration is run, usually the server computer (e.g. **C:\QPRSuiteETLTasksWorkbench**).
2. Configure the xml configuration file and save to any location in the computer (e.g. **C:\QPRSuiteETLTasksWorkbench\configuration**).
3. To run the integration manually, create a shortcut where there is the following command:

[exe path] [configuration file path]

Example:

```
"C:\QPRSuiteETLTasksWorkbench\QPRSuiteETLTasksWorkbench.exe"  
"C:\QPRSuiteETLTasksWorkbench\configuration\configuration.xml"
```

4. For automatic run, create Windows Task Scheduler task for the integration. Use the same run command as in step 3.
5. When running the integration for the first time and .Net 4.5 is not installed, the installation is prompted. Windows restart is required after .Net 4.5 installation. Manual installation of .Net 4.5 is also possible:
 - Online installer: <https://www.microsoft.com/en-us/download/details.aspx?id=48130>
 - Offline installer: <http://www.microsoft.com/fi-fi/download/details.aspx?id=48137>

3 Configuration file

The configuration file uses XML format having the following structure:

```
integrationtasks (1)
  commonparameters (0...1)
    parameter (1...n)
  [integrationtask] (1...n)
    headerproperty (0...n)
    rowproperty (0...n)
    attribute (0...n)
    binaryattribute (0...n)
```

The root tag is **integrationtasks** and it has integration tasks as sub tags. Tasks are executed in the order they are defined in the configuration. Available integration task types are listed in the **Integration Tasks** chapter.

QPR Web Services and QPR API interface features and restrictions determine which tasks are suitable for the purpose. E.g. updating modelling client content requires QPR API and thus modellingclientcreateorupdate task need to be used (webservicecreateorupdate doesn't work). Integration task tags are described in more detail in the next chapter.

Integration task tags have **headerproperties** and **rowproperties** child tags which define properties of the task. Property values can be defined two ways:

- **value** attribute: a static string value.
- **expression** attribute: value is calculated using defined expression.

Headerproperties are processed once for the tag when tag processing starts. Row properties are processed for each source object, so those expressions may reference to source object attributes. Also row properties may reference to header properties but not vice versa.

Certain tasks contain **attribute** tags defining attribute values that are saved for the created or updated objects. **binaryattribute** is corresponding with the exception that it defines binary data attributes instead of string attributes. Storing of binary data uses QPR Web Service's SetBinaryData operation. *Attribute* and *binaryattribute* values can be defined like *headerproperties* and *rowproperties* using *value* and *expression* XML attributes.

First tag below integrationtasks is **commonparameters** which has **parameter** sub tags. They define common arguments for expressions that are available in all tasks. Parameter values can be defined using value or expression attributes as explained above for header and row properties.

The integrationtasks root tag contains following attributes:

Attribute	Description
logfile	Folder path and file name of the log file. If omitted, logging is disabled. Logged data is appended to the end of an existing file. The folder for the log file must pre-exists but the file doesn't.
servername	QPR server name for Modelling and Metrics Clients. Needed only if Modelling Client API is used.

modellingserverport	QPR server port for Modelling Client. Needed only if Modelling Client API is used.
metricsserverport	QPR server port for Metrics Client. Needed only if Metrics Client API is used.
username	QPR username for logging in to QPR Web Service, Modelling Client and Metrics Client.
password	QPR password.
domain	QPR user domain (for QPR API).
loglevel	Determines the amount of information that is logged during the integration process. One of the following: <ul style="list-style-type: none"> - error: Only errors are logged. - information: In addition to error level, main phases of the integration run are logged. - verbose: In addition to information level, very detailed logging is performed.
wcfsecuritymode	See QPR Reporting Add-on documentation: https://community.qpr.com/node/2198 .
qprauthenticationmode	See QPR Reporting Add-on documentation: https://community.qpr.com/node/2198 . In addition, the option <i>passedsession</i> is not available for QPR Suite ETL Workbench.
qprwebserviceaddress	See QPR Reporting Add-on documentation: https://community.qpr.com/node/2198 .

4 Integration tasks

In tables of this chapter mandatory properties for the integration tasks are defined as **(M)**. In addition, **(h)** means header properties and **(r)** row properties. If either of these is not mentioned, property can be used either as a header or a row property.

4.1 Export embedded files

Exports embedded files from QPR system to files in disk (to defined location and name).

Note that the user account running the tool needs to have rights to write to the target folder(s).

When configuring the export logic, take into account that the tool won't try to write files with the same name to the same folder (existing files will be overwritten).

This task works with the following procedure:

1. If sourcedataset is provided, that is used. Alternatively if sourcequery is provided, a web service query is made (also following properties are used: sourcecriteria, sourcesortby, sourceattributes, sourceoptions).
2. Source data rows are processed one by one by calling GetBinaryData is for every row. A file is saved to the location defined location.

Property	Description
sourcedataset (h)	Source data as a dataset. This is an alternative to sourcequery (and following properties: sourcecriteria, sourcesortby, sourceattributes, sourceoptions). Dataset data is available as expression arguments (like sourceattributes) (column names as argument names).
sourcequery (h)	QPR Web Service QueryObjects query to get the source data. Alternative to sourceobjects. More information: http://kb.qpr.com/qpr2016-1/index.html?queryobjects.htm .
sourcecriteria (h)	"criteria" for the source query.
sourcesortby (h)	"sort by" for the source query.
sourceattributes (h)	Queried "attributes" for the source query. Attribute names are available as expression arguments for properties getbinarydataobjectid, getbinarydataattribute, getbinarydataoptions, filename and filepath.
sourceoptions (h)	"options" for the source query.
getbinarydataobjectid (M)(r)	Object id for the GetBinaryData operation to get the file binary data. More information: http://kb.qpr.com/qpr2016-1/index.html?getbinarydata.htm .
getbinarydataattribute (M)(r)	Attribute for the GetBinaryData operation to get the file binary data.
getbinarydataoptions (r)	Options for the GetBinaryData operation to get the file binary data. Empty value is provided for GetBinaryData, if this parameter is not provided.
filename (M)(r)	Name of the stored file including the file suffix, e.g. file1.docx
filepath (M)(r)	Full directory path for the stored file, e.g. C:/folder 1/folder 2/

4.2 Export to Excel (not implemented)

Tag **exporttoexcel** exports data in a dataset to an Excel file. The first row in the Excel will contain dataset header names.

Property	Description
dataset (h)	Input dataset for the export. The dataset may contain type of data that can be stored in an Excel cell, such as string, integer, date, and boolean.
excellocation (h)	Excel file location in the disk. If an already existing file is found, it's deleted before the export.
Worksheet name (h)	Worksheet name for the exported data.

4.3 Import embedded files (not implemented)

Imports files from disk to embedded attachments to QPR system. All imported files must be in a single folder.

Property	Description
sourcefolder (M)(h)	Source folder in disk where files are read.
identifierquery (M)	QPR Web Service QueryObjects query used to identify whether the object already exists. Following arguments are available: <ul style="list-style-type: none"> - filename - filesuffix - filemimetype - filecontentlength - filepath
identifiercriteria	"criteria" for the identifier query. Same arguments as identifierquery are available.
identifieroptions	"options" for the identifier query. Same arguments as identifierquery are available.
createobjectnamespace (M)	Namespace for QPR Web Services CreateObject operation. Same arguments as identifierquery are available.
createobjectoptions (r)	Options for the QPR Web Services CreateObject operation used to set attribute values. Same arguments as identifierquery are available.
setbinarydataobjectids (r)	Object ids for SetBinaryData. More information: http://kb.qpr.com/qpr2016-1/index.html?setbinarydata.htm . Same arguments as identifierquery are available.
setbinarydataattribute (r)	Attribute for for SetBinaryData. Same arguments as identifierquery are available.
setbinarydataoptions (r)	Options for for SetBinaryData. Same arguments as identifierquery are available.

4.4 Import measure values

Tag **importmeasurevalues** imports Metrics measure values using Web Services SetSCMeasureValuesForModel operation. The task first tries to import all values in a single operation (this is called "mass mode"). Depending on the errormode property, the task tries the import value-by-value, if an error occur in the mass mode.

Property	Description
sourcedataset (M)(h)	Source data as a dataset, which must contain columns scorecard, measure, series, period, value . The columns must be strings, except the value column may also be of a numeric datatype. The dataset can also contain other columns, and the columns may be in any order. See how the data is formatted in http://kb.qpr.com/qpr2016-1/index.html?setscmeasurevaluesformodel.htm .
modelid (M)(h)	Model id for the model where the values are imported (i.e. modelid for the SetSCMeasureValuesForModel operation).
options (h)	Options for the SetSCMeasureValuesForModel operation.

errormode (h)	<p>Error modes determine how error situations are handled. Available error modes (1-4):</p> <ul style="list-style-type: none"> - 1: If mass importing fails, throws an error. - 2: If mass importing fails, ends import without throwing an error. Integration log mentions if an error has occurred (in log level Information). - 3: If mass import fails, tries value-by-value import. If value-by-value import fails, throws an error. - 4: If mass import fails, tries value-by-value import. If value-by-value import fails, imports rest of the rows without throwing an error. Integration log mentions if an error has occurred, and also the lines where errors have occurred (in log level Information). <p>Error modes 1 and 3 are for serious errors that need to be investigated. Error modes 2 and 4 are for error situations that are not serious: the errors are mentioned in the log, but the integration is considered to end with success.</p>
---------------	---

4.5 Modelling client create or update

Tag **modellingclientcreateorupdate** creates new objects or updates attributes of existing using QPR Modelling Client (QPR API). Data is fetched using QPR Web Service (called **source query**). For all the source objects, an **identifier query** is executed determining whether a new object is created or an existing object attributes updated.

The operation may be used to only update properties of existing objects, if the identifier query is built so that it always returns an object.

Note that the EA/PD models must be available in QPR Web Services, i.e. they must be published.

Property	Description
sourcedataset (h)	Source data as a dataset. This is an alternative to sourcequery (and following properties: sourcecriteria, sourcesortby, sourceattributes, sourceoptions). Dataset data is available as expression arguments (like sourceattributes) (column names as argument names).
sourcequery (h)	QPR Web Service QueryObjects query to get the source data. Alternative to sourcedataset.
sourcecriteria (h)	"criteria" for the source query.
sourcesortby (h)	"sort by" for the source query.
sourceattributes (h)	Queried "attributes" for the source query. Attribute names are available as expression arguments for properties identifierquery, identifiercriteria, identifieroptions, createmodelid, createparameters and all attributes.
sourceoptions (h)	"options" for the source query.
identifierquery (M)	QPR Web Service QueryObjects query used to identify whether the object already exists.
identifiercriteria	"criteria" for the identifier query.

identifioptions	"options" for the identifier query.
modelid	Id of the model where the objects are created updated. In terms of existing objects, the object id contains the model id, so that model id is used when updating existing objects. modelid is mandatory if the task may create objects.
createparameters	Parameters for the QPR API CreateElement operation, which is used when new objects need to be created.

Set attributes are defined using **attribute** tags. Attributes **name** and **typename** are mandatory, if the operation is supposed to create objects.

4.6 Modelling client delete

Tag **modellingclientdelete** deletes EnterpriseArchitect/ProcessDesigner objects using QPR Modelling Client (QPR API) and DeleteElement command (documentation: <http://kb.qpr.com/qpr2016-1/index.html?deleteelement2.htm>).

DeleteElement command enables to delete objects and object instances. Object instances are deleted by adding InstanceId parameter. If the object to delete (sourceobjects or sourcequery) are instances, this parameter is automatically added.

If there are objects to delete from several models, it recommended to sort the queried based on the model, because the Modelling client opens one model at a time.

Property	Description
sourceobjects (h)	Array of source objects. The array is formulated using an expression. Alternative to sourcequery.
sourcequery (h)	QPR Web Service QueryObjects query to get the source data. Alternative to sourceobjects.
sourcecriteria (h)	"criteria" for the source query.
sourcesortby (h)	"sort by" for the source query.
sourceoptions (h)	"options" for the source query.
deleteelementparameters	parameters (the second parameter) for the QPR API's DeleteElement operation

4.7 Perform object operation

Tag **performobjectoperation** uses QPR Web Services' PerformObjectOperation (<http://kb.qpr.com/qpr2016-1/index.html?performobjectoperation.htm>)

Property	Description
objectId (h)	Single object as string, or an array of objects as string array.

operation (h)	Operation as string. See http://kb.qpr.com/qpr2016-1/index.html?performobjectoperation.htm
options (h)	Options as string. See http://kb.qpr.com/qpr2016-1/index.html?performobjectoperation.htm
dataset (h)	Several objectids, operations, options can be defined using a dataset. The dataset must contains headers "objectid", "operation" and "options".

4.8 Synchronize metrics model

Tag **synchronizemetricsmodel** refreshes QPR Metrics model drilldown, inheritance, and optionally recalculation for all scorecards in the defined model(s) using QPR Metrics Client (QPR API).

Property	Description
modelsquery (M) (h)	Query to get the Metrics model for which the operation is executed. The query can return multiple models, and the operation is done all of them.
refreshdrilldown (h)	true or false . If true, model drilldown is refreshed. Default is true.
applyinteritance (h)	true or false . If true, model inheritance is done for all base scorecards. Default is true.
recalculate (h)	true or false . If true, model recalculation is done. Default is false.

4.9 Web services create copy (not implemented)

Tag **webservicescreatecopy** executes QPR Web Services' CreateCopy operation (<http://kb.qpr.com/qpr2016-1/index.html?createobject2.htm>).

Property	Description
sourcedataset (h)	Source data as a dataset. This is an alternative to sourcequery (and following properties: sourcecriteria, sourcesortby, sourceattributes, sourceoptions). Dataset data is available as expression arguments (like sourceattributes) (column names as argument names).
sourcequery (h)	QPR Web Service QueryObjects query to get the source data. Alternative to sourcedataset.
sourcecriteria (h)	"criteria" for the source query.
sourcesortby (h)	"sort by" for the source query.
sourceattributes (h)	Queried "attributes" for the source query. Attribute names are available as expression arguments for properties identifierquery, identifiercriteria, identifieroptions, createmodelid, createparameters and all attributes.
sourceoptions (h)	"options" for the source query.

identifierquery (M)	QPR Web Service QueryObjects query used to identify whether the object already exists.
identifiercriteria	"criteria" for the identifier query.
identifioptions	"options" for the identifier query.
setattributeoptions	Options for the QPR Web Services SetAttributes operation used to set attribute values.
createcopyparentids	
createcopyoptions	Options for the QPR Web Services CreateCopy operation used to set attribute values.

4.10 Web services create or update

Tag **webservicescreateorupdate** creates new objects or updates attributes of existing ones using QPR Web Service. When this task is executed, the defined **source query** is made. For all result objects an **identifier query** is made. If a match is found, the matching element is updated. If no match is found, a new element is created.

Property	Description
sourcedataset (h)	Source data as a dataset. This is an alternative to sourcequery (and following properties: sourcecriteria, sourcesortby, sourceattributes, sourceoptions). Dataset data is available as expression arguments (like sourceattributes) (column names as argument names).
sourcequery (h)	QPR Web Service QueryObjects query to get the source data. Alternative to sourcedataset.
sourcecriteria (h)	"criteria" for the source query.
sourcesortby (h)	"sort by" for the source query.
sourceattributes (h)	Queried "attributes" for the source query. Attribute names are available as expression arguments for properties identifierquery, identifiercriteria, identifioptions, createmodelid, createparameters and all attributes.
sourceoptions (h)	"options" for the source query.
identifierquery (M)	QPR Web Service QueryObjects query used to identify whether the object already exists.
identifiercriteria	"criteria" for the identifier query.
identifioptions	"options" for the identifier query.
setattributeoptions	Options for the QPR Web Services SetAttributes operation used to set attribute values.
createobjectnamespace (M)	Namespace for QPR Web Services CreateObject operation.
createobjectoptions	Options for the QPR Web Services CreateObject operation used to set attribute values.

Set attributes are defined using **attribute** tags. Attributes **name**, **typename** and attribute defining parents ("parentid", "parentelementid", "parentscorecardid") are mandatory, because they are needed when creating objects.

Attribute tag have following XML attributes:

- **setseparately** (boolean): Set the attribute separately using QPR Web Services SetAttribute operation. Other attributes are set at the same time using SetAttributes. Attributes are only saved separately when updating an existing object – when a new object is created, all attributes are set in QPR Web Services CreateObject operation. Default value is false.
- **options** (string): Options to use in QPR Web Services SetAttribute (only when setseparately is true). Default value is empty.

4.11 Web services delete

Tag webservicesdelete executes QPR Web Services' DeleteObject operation (<http://kb.qpr.com/qpr2016-1/index.html?deleteobject2.htm>). All the queried objects are deleted.

Property	Description
sourceobjects (h)	Array of source objects. The array is formulated using an expression. Alternative to sourcequery.
sourcequery (h)	QPR Web Service QueryObjects query to get the source data. Alternative to sourceobjects.
sourcecriteria (h)	"criteria" for the source query.
sourceoptions (h)	"options" for the source query.
deleteobjectoptions	"options" for the DeleteObject operation.