



QlikView Custom Source Control Provider

This version was published in March, 2012

www.qlikview.com

Table of Contents

QLIKVIEW CUSTOM SOURCE CONTROL PROVIDER	3
PROVIDER FILE PROPERTIES	3
LAUNCHING PROVIDER	3
INTERPROCESS COMMUNICATION VIA NAMED PIPES	4
QLIKVIEW REQUEST.....	5
PROVIDER REPLY	15
USING PROVIDERS.....	17

QlikView Custom Source Control Provider

QlikView communicates with QlikView Custom Source Control Provider (Provider) in order to use the provided Source Control system. The Provider is responsible for executing the source control operations requested by QlikView, and when necessary, show dialogs with different settings and error messages to the user. This document includes details about how the Provider is launched and how the communication between the applications is performed.

Provider File Properties

For QlikView to recognize an .exe file as Provider, the file's version-information resource must include the property "QlikView Provider". Its value is used as a Provider display name and must be specified.

File version information can be set using different methods, for example by:

- Defining version information in a resource file for a C++ program. Search for "VERSIONINFO Resource" in the Internet to read more about this.
- Using third party tools, such as Version Resource Tool:
<http://www.codeproject.com/KB/install/VerPatch.aspx>.

Launching Provider

QlikView starts Provider process and passes two **command-line arguments**:

- named pipe name
- parent window handle

Each opened QlikView document that is attached to source control has its own Provider process. The Provider process is responsible for all source control operations for the document that owns it. The process is terminated when the QlikView document is closed.

A Provider process is launched:

- When **Add Project to Source Control** is selected from the **Source Control** menu.
- When **Get Project from Source Control** is selected from the **Source Control** menu.
- On opening a QlikView document that is attached to source control.
- In certain cases when the **Settings** button for a source control Provider is selected. If the active QlikView document is attached to source control, and the selected source control

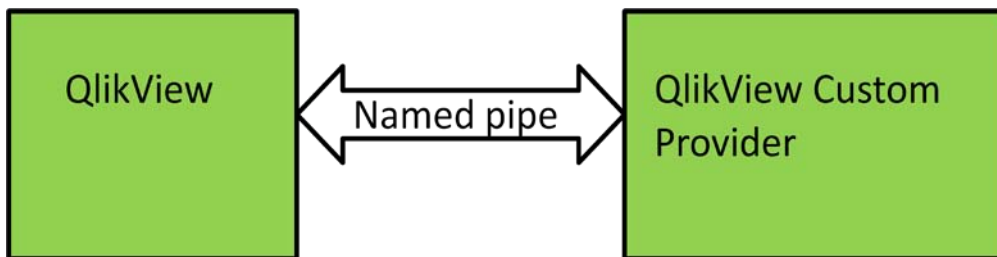
provider is identical, then no new Provider process will be launched. Instead, the Provider process that belongs to the active document will be used. A new Provider process is launched if no QlikView document is active or the active document is not attached to source control.

Interprocess Communication via Named Pipes

Named pipes are used for exchanging XML formatted messages:

- QlikView requests (QscRequest), which describe commands to be performed by the Provider
- Provider replies (QscReply), which describe status of the executed commands.

After sending a request, QlikView waits for the Provider reply before proceeding with other actions. In certain cases, replies may include result data.



Example scenarios of how QlikView and the Provider exchange messages to execute:

- Add project:
 1. QlikView sends an add project request (QSC_ADD_PROJECT) via the named pipe and waits for the Provider to reply. The request includes a list of file paths that should be added to source control and a path to the local project folder.
 2. The Provider shows a dialog to the user, for example asking the user for the repository URL and other relevant information. When the user confirms that he/she wants to add project to source control, the Provider adds the project to the specified repository.
 3. If the user cancels the operation, the Provider replies with QSC_CANCEL. Otherwise, the Provider replies with QSC_OK for success and QSC_UNKNOWN_ERROR for failure.

- Get project:
 1. QlikView sends a get project request (QSC_GET_PROJECT) via the named pipe and waits for the Provider to reply.
 2. The Provider shows a dialog to the user, for example asking the user for the repository URL and the path on the local computer to place the project on. When the user confirms that he/she wants to get project from source control, the Provider gets the project from the specified repository URL, validates the project path (must end with "-prj") and places it in the specified path on the local computer.
 3. If the user cancels the operation, the Provider replies with QSC_CANCEL. Otherwise, the Provider replies with QSC_OK for success and QSC_UNKNOWN_ERROR for failure.
- Check out:
 1. QlikView sends a check out request (QSC_CHECKOUT) via the named pipe and waits for the Provider to reply. The request includes a list of file paths that should be checked out from source control and a path to the local project folder.
 2. For some source control systems, such as Subversion, there is no need to check out a file before editing it. In this case, the Provider returns QSC_OK. For other source control systems, such as TFS, a file needs to be checked out before editing it. It is recommended that this should be done silently, i.e. without showing a dialog with the list of files to be checked out.
 3. The Provider returns either QSC_OK for success or QSC_UNKNOWN_ERROR for failure.

QlikView Request

The data stream of the QlikView request is formatted as follows:

- 4 bytes that specify the length of the following string
- A zero terminated string that is formatted according to the QscRequest XML schema

QscRequest XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:simpleType name="QscCommand">
    <xs:restriction base="xs:string">
      <xs:enumeration value="QSC_SHOW_SETTINGS"/>
      <xs:enumeration value="QSC_INITIALIZE_PROJECT"/>
      <xs:enumeration value="QSC_ADD_PROJECT"/>
      <xs:enumeration value="QSC_GET_PROJECT"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

```

<xs:enumeration value="QSC_GET_LATEST"/>
<xs:enumeration value="QSC_CHECKIN"/>
<xs:enumeration value="QSC_UNDO_CHECKOUT"/>
<xs:enumeration value="QSC_IS_OUTOFDATE"/>
<xs:enumeration value="QSC_GET_SOURCECONTROLLED_FILES"/>
<xs:enumeration value="QSC_HAS_PENDING_CHANGES"/>
<xs:enumeration value="QSC_UNDO_CHECKOUT_MODIFIED_FILES"/>
<xs:enumeration value="QSC_TERMINATE"/>
</xs:restriction>
</xs:simpleType>

<xs:element name="QscRequest">
  <xs:complexType>
    <xs:all>
      <xs:element name="Command" type="QscCommand" />
      <xs:element name="Parameters">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="String" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="FileNames">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="String" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:all>
  </xs:complexType>
</xs:element>
</xs:schema>

```

QscRequest Element

Child Element	Description
Command	Command to be executed by the Provider. See QscCommand .
Parameters	Array of strings. Holds command specific parameters.
FileNames	Array of strings. Holds file paths.

QscCommand Type

Type Value	Description
QSC_SHOW_SETTINGS	<p>Show the Provider's own settings dialog. The Provider may require the user to specify some information such as which dll to use etc.</p> <p>The command is sent when the user presses the Settings button in the source control Settings dialog in QlikView.</p> <p>Request parameters:</p> <p>[0] - Window handle of the source control Settings dialog.</p> <p>Request filenames:</p> <p>None</p> <p>Reply:</p> <ul style="list-style-type: none"> • QSC_OK, if the Provider's settings dialog was successfully displayed. • QSC_CANCEL, if the user canceled the operation.
QSC_INITIALIZE_PROJECT	<p>Initialize the Provider, e.g. loading dlls.</p> <p>The command is sent when a QlikView document that is attached to source control opens or reopens. A QlikView document reopens after Add Project, Get Project, Get Latest, Check In and Undo Checkout operations.</p>

	<p>Request parameters:</p> <p>[0] - The local path to the project (the PRJ folder)</p> <p>Request filenames:</p> <p>None</p> <p>Reply:</p> <p>QSC_OK, if the Provider was successfully initialized.</p>
QSC_ADD_PROJECT	<p>Add the project to source control.</p> <p>Typically, the Provider shows a dialog where the user can choose a repository to add the project to.</p> <p>The command is sent when Add Project to Source Control is selected.</p> <p>Request parameters:</p> <p>[0] - The local path to the project (the PRJ folder)</p> <p>Request filenames:</p> <p>A list of local file paths of the files to add to source control.</p> <p>Reply:</p> <ul style="list-style-type: none"> • QSC_OK, if the project was successfully added to source control. • QSC_CANCEL, if the user canceled the operation.
QSC_GET_PROJECT	<p>Get a project from source control.</p> <p>Typically, the Provider shows a dialog where the user can choose a repository URL to get the project from. It should also be possible to specify the local destination on the user's local computer.</p> <p>The command is sent when Get Project from</p>

	<p>Source Control is selected.</p> <p>Request parameters:</p> <p>None</p> <p>Request filenames:</p> <p>None</p> <p>Reply:</p> <ul style="list-style-type: none"> • QSC_OK, if the project was successfully retrieved from source control. • QSC_CANCEL, if the user canceled the operation.
<p>QSC_GET_LATEST</p>	<p>Get the latest version of the project from source control. It is the Provider's responsibility to provide merge functionality in case of conflicts.</p> <p>The command is sent when Get Latest Version is selected.</p> <p>Request parameters:</p> <p>[0] - The local path to the project (the PRJ folder)</p> <p>Request filenames:</p> <p>A list of all files (local file paths) that QlikView holds as current project files. This list may not be the same as the files in the local project folder.</p> <p>By comparing these files and the project files in the repository, the Provider can calculate which files have been added/removed or modified.</p> <p>Reply:</p> <ul style="list-style-type: none"> • QSC_OK, if the project was successfully retrieved from source control. • QSC_CANCEL, if the user canceled the operation.

<p>QSC_CHECKIN</p>	<p>Check in pending changes.</p> <p>Typically, the Provider shows a dialog with a list of changes and asks the user to confirm the changes before checking in the files.</p> <p>The command is sent when Check In Pending Changes is selected.</p> <p>Request parameters:</p> <p>[0] - The local path to the project (the PRJ folder)</p> <p>Request filenames:</p> <p>A list of all files (local file paths) that QlikView holds as current project files. This list may not be the same as the files in the local project folder.</p> <p>By comparing these files and the project files in the repository, the Provider can calculate which files have been added/removed or modified.</p> <p>Reply:</p> <ul style="list-style-type: none"> • QSC_OK, if the changes were successfully checked in to source control. • QSC_CANCEL, if the user canceled the operation.
<p>QSC_CHECKOUT</p>	<p>Check out files.</p> <p>For source control systems that do not require files to be checked out prior to editing, e.g. Subversion, return QSC_OK. For source control systems that require files to be checked out prior to editing, e.g. TFS, check out the files silently.</p> <p>The command is sent when QlikView wants to make changes in a file that need to be checked out, typically, when the user saves a source controlled project.</p>

	<p>Request parameters:</p> <p>[0] - The local path to the project (the PRJ folder)</p> <p>Request filenames:</p> <p>A list of files (local file paths) that needs to be checked out.</p> <p>Reply:</p> <ul style="list-style-type: none"> • QSC_OK, if the source control system does not require check out prior to editing. Also, if the changes were successfully checked out from source control.
QSC_UNDO_CHECKOUT	<p>Undo pending changes.</p> <p>Typically, the Provider shows a dialog with a list of changes and asks the user to confirm before undoing the pending changes.</p> <p>The command is sent when Undo Pending Changes is selected.</p> <p>Request parameters:</p> <p>[0] - The local path to the project (the PRJ folder)</p> <p>Request filenames:</p> <p>A list of all files (local file paths) that QlikView holds as current project files. This list may not be the same as the files in the local project folder.</p> <p>By comparing these files and the project files in the repository, the Provider can calculate which files have been added/removed or modified.</p> <p>Reply:</p> <ul style="list-style-type: none"> • QSC_OK, if the operation was successful. • QSC_CANCEL, if the user canceled the operation.

<p>QSC_IS_OUTOFDATE</p>	<p>Checks if the project is out of date.</p> <p>The Provider should not show any dialogs for this request.</p> <p>The command is sent during Check In and Get Latest operations.</p> <p>Request parameters:</p> <p>[0] - The local path to the project (the PRJ folder)</p> <p>Request filenames:</p> <p>A list of all files (local file paths) that QlikView holds as current project files. This list may not be the same as the files in the local project folder.</p> <p>Reply:</p> <ul style="list-style-type: none"> • QSC_OK, if the operation was successful. • OutputValues[0] – “true” if the project is out of date, otherwise “false”.
<p>QSC_GET_SOURCECONTROLLED_FILES</p>	<p>Get all project source controlled files.</p> <p>The Provider should not show any dialogs for this request.</p> <p>The command is sent when the user saves a source controlled project.</p> <p>Request parameters:</p> <p>[0] - The local path to the project (the PRJ folder)</p> <p>Request filenames:</p> <p>None</p> <p>Reply:</p> <ul style="list-style-type: none"> • QSC_OK, if the operation was successful. • FileNames – a list of all source controlled files (local file paths) that belong to the project.

<p>QSC_HAS_PENDING_CHANGES</p>	<p>Checks if the project has changes that are not checked in to source control.</p> <p>The Provider should not show any dialogs for this request.</p> <p>The command is sent when something happens that could cause the source control menu items to be disabled/enabled. For example, after a successful check in, the Check In Pending Changes menu item should be disabled. The command is also sent when QlikView needs to know if the status text “In SC” needs to be updated, e.g. when opening a document that is attached to source control.</p> <p>Request parameters:</p> <p>[0] - The local path to the project (the PRJ folder)</p> <p>Request filenames:</p> <p>A list of all files (local file paths) that QlikView holds as current project files. This list may not be the same as the files in the local project folder.</p> <p>By comparing these files and the project files in the repository, the Provider can calculate if there are pending changes to source control.</p> <p>Reply:</p> <ul style="list-style-type: none"> • QSC_OK, if the operation was successful. • OutputValues[0] – “true” if there are pending changes to source control.
<p>QSC_UNDO_CHECKOUT_UNMODIFIED_FILES</p>	<p>Undo check out files that are checked out but not modified.</p> <p>For source control systems that do not require files to be checked out prior to editing, e.g. Subversion, just return QSC_OK. For source control systems that require files to be</p>

	<p>checked out prior to editing, e.g. TFS, undo check out all files that are checked out but not modified. This should be done silently, i.e. without display of dialogs.</p> <p>The command is sent when the user saves a source controlled project.</p> <p>Request parameters:</p> <p>[0] - The local path to the project (the PRJ folder)</p> <p>Request filenames:</p> <p>A list of all files (local file paths) that have changes (some of them may already be checked out).</p> <p>By comparing the content of these files and the corresponding files in the repository, the Provider can calculate files that are checked out and identical to the repository version.</p> <p>Reply:</p> <ul style="list-style-type: none"> • QSC_OK, if the operation was successful.
QSC_TERMINATE	<p>Clean up as the process will be terminated. If QlikView receives a reply, then it closes the named pipe and terminates the Provider process. Otherwise make sure to terminate.</p> <p>Request parameters:</p> <p>None</p> <p>Request filenames:</p> <p>None</p> <p>Reply:</p> <p>QSC_OK,if clean-up is finished and QlikView can terminate the Provider process.</p>

Command Request Example

```
<QvxRequest>
  <Command> QSC_INITIALIZE_PROJECT</Command>
  <Parameters>
    <String> C:\MyQlikViewApps\ABC-prj</String>
  </Parameters>
</QvxRequest>
```

Provider Reply

As a response to QlikView's request, the Provider sends a message formatted according to QscReply XML schema. The data stream is formatted as follow:

- 4 bytes that specify the length of the following string
- A zero terminated string that is formatted according to the QscReply XML schema

QscReply XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:simpleType name="QscResult">
    <xs:restriction base="xs:string">
      <xs:enumeration value="QSC_OK"/>
      <xs:enumeration value="QSC_UNKNOWN_COMMAND"/>
      <xs:enumeration value="QSC_UNSUPPORTED_COMMAND"/>
      <xs:enumeration value="QSC_UNEXPECTED_COMMAND"/>
      <xs:enumeration value="QSC_CANCEL"/>
      <xs:enumeration value="QSC_CHECKIN"/>
      <xs:enumeration value="QSC_UNKNOWN_ERROR"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:element name="QscReply">
    <xs:complexType>
      <xs:all>
```

```

<xs:element name="Result" type="QscResult" />
<xs:element name="OutputValues">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="String" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="FileNames">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="String" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:all>
</xs:complexType>
</xs:element>
</xs:schema>

```

QscReply Element

Child Element	Description
Result	The result of the operation. See QscResult .
OutputValues	String array of result values. See QscCommandType table for how it is used.
FileNames	String array of filenames. See QscCommandType table for how it is used.

QscReply Example

```
<?xml version="1.0"?>  
<QscReply>  
  <Result>QSC_OK</Result>  
</QscReply>
```

Using Providers

QlikView looks for the following Providers in the specified order:

1. At the same location as the current QV.exe
2. In the "Program Files\Common Files\QlikTech\Custom Data" folder and subfolders
3. For 64-bit operating systems only: in the "Program Files (x86)\Common Files\QlikTech\Custom Data" folder and subfolders
4. For 64-bit operating systems only: "Program Files (x86)\QlikView" folder and subfolders

On 64-bit operating systems, the 64-bit QlikView loads both 64- and 32-bit Providers.

For all 32-bits Providers, QlikView adds "**32" after the display name. If several Providers have the same display name, the first one will keep its display name and the others will get a unique suffix, e.g. "QlikView Svn Provider *32", "QlikView Svn Provider *32 (1)" and "QlikView Svn Provider *32 (2)".

QlikView



© 2012 QlikTech International AB. All rights reserved. QlikTech, QlikView, Qlik, Q, Simplifying Analysis for Everyone, Power of Simplicity, New Rules, The Uncontrollable Smile and other QlikTech products and services as well as their respective logos are trademarks or registered trademarks of QlikTech International AB. All other company names, products and services used herein are trademarks or registered trademarks of their respective owners.

The information published herein is subject to change without notice. This publication is for informational purposes only, without representation or warranty of any kind, and QlikTech shall not be liable for errors or omissions with respect to this publication. The only warranties for QlikTech products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting any additional warranty.