DB2 Web Query
Application Extension
Usage Instructions

Updated October 2012

Contents

Overview 3

Part 1: Setup 3

Part 2: Using the extension 4

Part 2.1 Browse mode 4

Part 2.2: Direct mode 4

Part 3: Properties file and options 6

Part 3.1: server 6

Part 3.2: port 6

Part 3.3: useSSL 6

Part 3.4: wqsoaBasicAuthEnabled 6

Part 3.5: useValidationList, validationListLibrary, and validationListName 6

Part 3.6: browse 7

Part 3.7: wqsoaShowAboutLink 7

Part 3.8: cacheTimeout 7

Part 3.9: wqsoaTemplateHead and wqsoaTemplateFoot 7

Part 3.10: wqsoaSelectListSize 7

Part 3.11: wqsoaSchema 8

Part 3.12: wqsoaEnableStats 8

Part 3.13: wqsoaEnableSavedReportParameters, wqsoaEnableReportSharing, and wqsaoEnableReportSharingWithLists 8

Part 3.14: wqsoaEnableBackgroundReports, wqsoaSMTPServer, and wqsaoSMTPSender 8

Part 3.15: debug 9

Overview

The DB2 Web Query Application Extension allows a Web Query executable to be run and the output via a URL. This Web Query executable can be a:

- Report
- Graph
- Compound document

There are limitations to the items that can be run. The types of items that cannot be run via this extension are reports that include OLAP, on demand pages, InfoMini, or HTML Composer dashboards. The Web Query executable can only be run with a std_reports structure. Here is an example. ***mydomain/sd_reports/myreportfldr/my_report.fex.*** This document contains the information for using the DB2 Web Query Application Extension.

Part 1: Setup

Be default, the Application Extension is already installed, configured, and available for use when you install Web Query Standard Edition. There is a properties file in the IFS that contains all of the possible options for the extension. We will discuss those in this document in Part 3.

Part 2: Using the extension

There are two modes of use for the Application Extension: browse and direct.

Part 2.1 Browse mode

To access the extension, simply go to http://your_server:12331/wqsoa, where your_server is your IBM i's IP address or DNS name. Notice that this is almost the same link you use to access the normal DB2 Web Query interface, just the last part changes.

Be default, it will prompt you to log in with your IBM i user profile and password. If you are in a corporate environment where this is on an intranet, you may wish to not require your users to login and simply grant them access. You can do this by providing a static user profile and password in the properties file, described in Part 3.

Once you are logged in, all the domains that you are authorized to will be shown. Click on a domain to access the folders inside. Likewise, click on a folder to see the reports and charts inside. Optionally, if the report requires any user input, the user will be prompted using the same method as the report in DB2 Web Query, via drop down, selection list, or text box.

Once the report is selected and any parameters provided, the output is shown however the report was created. This could be HTML back to the browser, a PDF file, or an Excel file.

Part 2.2: Direct mode

As you select the domain, folder, and report in browse mode, look at the URL in your browser's address bar. You will see that it is building along with us, just filling in the internal names of each item. If we know the internal name of the domain or folder or report (accessible from the properties window or APIs) that we want to use, we can just put it in ourselves! The pattern in the URL is in key value pairs.

Plus, you can provide as little or as much information as you want in the URL! Provide nothing, start at the domain level. Provide a domain, start at the folder level. Provide a domain and folder, start at the report level. Provide a domain, folder, and report, get back the report if no parameters are required or get the prompt for the missing parameters. And note that I said missing parameters. You can also provide one or more parameters for the report. If a report has two parameters, you could provide one and the other would be prompted to the user. Provide both, and the report will come back directly!

Now that you have used the browse mode to build the URL string or built it yourself or programmatically, you can now use that string directly anywhere you want. You can link to it from a web page or PC application. You can set it as the source of an HTML iframe (in-line frame) on an existing web page. You can email the link to a list of colleagues.

Keep in mind it is always just a URL. So if you send it via email or post it on a web site, the end users will need to be able to have access to the IBM i. In a corporate environment, this could mean establishing a

VPN connection or gaining clearance to a firewall that may be protecting your server.

Part 3: Properties file and options

By default, the properties file is set to provide functionality without any configuration needed. However, there are some additional options you can turn on with a few simple parameters. In this section, we will describe each setting in the properties file. Each section does have a brief comment included in line prefaced by a hash (#). We provide a web interface at http://server:12331/wqsoa/config to make changes to this file. You must have *SECADM special authority to access this page.

Part 3.1: server

This parameter defines the DB2 Web Query server to connect to. By default, it will be localhost, meaning the local system. However, it is possible to have more than one server with Web Query on it and you may want to point the Application Extension running on one server to DB2 Web Query running on the other. Keep in mind that it will still check the license on the system where the Application Extension is running, even if it points to a remote DB2 Web Query server.

Example value: server=localhost

Part 3.2: port

This parameter is the port that DB2 Web Query is running on. This will always be 12331.

Example value: port=12331

Part 3.3: useSSL

This parameter determines if the connection type is HTTP or HTTPS using SSL/TLS. If you have installed a certificate into the DB2 Web Query web server, set this parameter to "true."

Example value: useSSL=false

Part 3.4: wqsoaBasicAuthEnabled

This parameter determines how the Application Extension gets its login credentials. By leaving the default of true, the end user will be prompted to provide these values. Used in conjunction with the user and password parameters mentioned above, this can also be set to false to use the hard coded values for every user session.

Example value: wqsoaBasicAuthEnabled=true

Part 3.5: useValidationList, validationListLibrary, and validationListName

By default, the Application Extension will prompt the user for their user profile and password. However, if you want all users to run under a common profile and password, provide that user profile and password as a Web Query Password object. Use the CRTWQPWD CL command to create the validation list. Then set useValidationList to true, leave validationListLibrary as qusrsys, and set validationListName to the name supplied during the CRTWQPWD command. Finally, change the wqsoaBasicAuthEnabled parameter to false.

Example value: useValidationList=true

Example value: validationListLibrary=qusrsys

Example value: validationListName=wqpassword

Part 3.6: browse

This parameter determines if the browse mode described in Part 2.1 is available or not. For security reasons, you may choose to disable this allowing only direct access.

Example value: browse=true

Part 3.7: wqsoaShowAboutLink

This parameter determines if the "About" link is shown when browse mode is used.

Example value: wqsoaShowAboutLink=true

Part 3.8: cacheTimeout

This parameter determines the number of seconds before any cached report graphs are discarded.

Example value: cacheTimeout=14400

Part 3.9: wqsoaTemplateHead and wqsoaTemplateFoot

The Application Extension automatically puts the wqsoaTemplateHead before each screen in browse mode and wqsoaTemplateFoot after each screen in browse mode. You can change these to match your own corporate look and feel for web sites. Load the HTML and all supporting assets (images, stylesheets, etc) into the /QIBM/UserData/qwebqry/WQLWI80/lwi/apps/eclipse/plugins/wqsoa/WEB-INF/classes/ folder. This only affects browse mode, not the reports themselves. Those are styled based on the report author.

Example value: wqsoaTemplateHead=ibm_header.html

Example value: wqsoaTemplateFoot=ibm_footer.html

Part 3.10: wqsoaSelectListSize

When there is a multiple choice parameter box, this determines how many possible entries are shown at once, affecting the overall height of that parameter.

Example value: wqsoaSelectListSize=5

Part 3.11: wqsoaSchema

Some of the options below require a schema or library to store data. This is the schema that will be used. Please create it on the system.

Example value: wqsoaSchema=QZRDWQSOA

Part 3.12: wqsoaEnableStats

There is an option to log statistics on who ran what report, with what options, when, and how long it took to run from request to data return. This can be useful to see if and when reports are being used or to test performance times. Stats will be stored in the wqsoaSchema.

Example value: wqsoaEnableStats=false

Part 3.13: wqsoaEnableSavedReportParameters, wqsoaEnableReportSharing, and wqsaoEnableReportSharingWithLists

The Application Extension allows users to save reports with their parameters. If they have lots of parameters and are using the same ones, this can be helpful. Saved parameters do not save the report output, just a link to the report and the parameter selection. This can be deleted or edited. Plus, optionally, once save is enabled, you can enable the ability for these saved reports to be shared and furthermore shared with lists of users. These build on each other – saved reports must be enabled to share, and share with lists requires share which requires save. Again, data for this is stored in the wqsoaSchema.

Example value: wqsoaEnableSavedReportParameters=false

Example value: wqsoaEnableReportSharing=false

Example value: wqsoaEnableReportSharingWithLists=false

Part 3.14: wqsoaEnableBackgroundReports, wqsoaSMTPServer, and wqsaoSMTPSender

The Application Extension has the ability to support a “run in the background” mode. This will allow users to submit an email address to have the results sent to once the report is complete, rather than waiting for the browser to return it. You must provide a valid SMTP server below. This could be the local IBM i or a remote system. You can use an IP address or DNS name. If you use a DNS name, make sure the IBM i can resolve that name. You can use the PING command to test that.

If you are sending emails to addresses not controlled by that SMTP server, you may run into an issue with relaying being blocked. Talk to your email server administrator about this issue if you are sending emails outside of your company. For example, your SMTP server is for mycompany.com. Sending to an address anything@mycompany.com would be fine. However, if sending to anything@gmail.com, it could be blocked if relaying is prevented.

The third parameter is the sender or from address. This could be a fictional address, a help desk, or perhaps someone in the area that could answer questions about the reporting environment.

Example value: wqsoaEnableBackgroundReports=false

Example value: wqsoaSMTPServer=localhost

Example value: wqsoaSMTPSender="WQSOA Report Service" <no-reply@localhost>

Part 3.15: debug

This parameter determines if debug messages are sent to the servlet log. You may need to add "com.ibm.ejs.ras.level=INFO" into the servlet's tracing specification if it is not already there. Use this option under guidance from a support representative.

Example value: debug=false