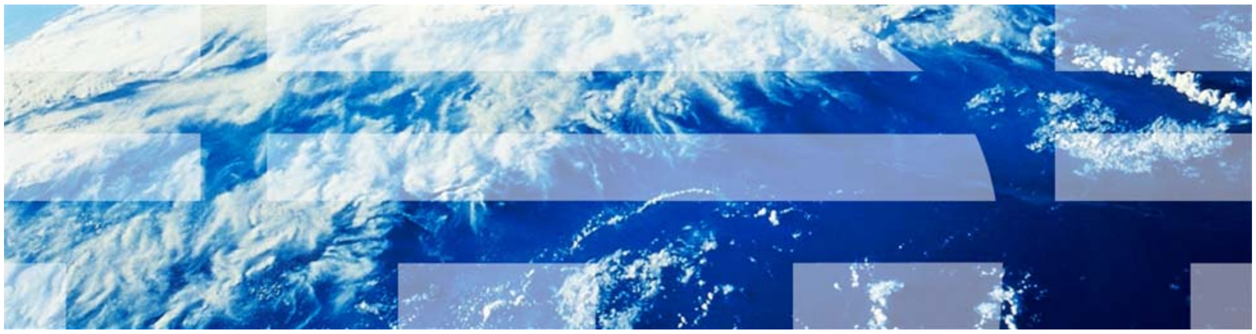
DB2 Web Query V2.1

Backup Guide



4/1/2014

Background

From an availability perspective, DB2 Web Query has several components that need to be considered.

- **Synonyms**
  Also referred to as metadata. All synonyms are stored under the Integrated File System (IFS) directory **/QIBM/UserData/qwebqry/apps.**

- **Reports**
  Also known as fexs or focexecs, these are the reports, documents and dashboards created via Web Query. These items reside in database tables in library **QWQREPOS**.

- **Schedules**
  This is the information created when a schedule is defined through the DB2 Web Query job scheduler. These also reside in library **QWQREPOS**. Note: schedules and scheduling is only applicable to DB2 Web Query StandardEdition.

- **Runtime Environments**
  This is the information created when a Run Time Environment (RTE) is created and enabled via the WRKWQRTE CL command. The information is stored in library **QWQREPOS**.

- **Configuration settings**
  This is the information associated with any Web Query installed configuration. This information is stored under different IFS directories:
  - **/QIBM/UserData/qwebqry/extensions** - for WQRAX Application Extension settings.
  - **/QIBM/UserData/qwebqry/base80** - for Report Broker, Change Management, Kerberos, and other client settings.
  - **/QIBM/UserData/qwebqry/WQLWI80** - for any web app container (LWI) settings.

For most customers the **Configuration settings** do not change from the installed defaults. If you do change any of these settings from their defaults, you may save the appropriate directory information, or simply remember the settings and restore them should you ever need a **full** recovery of the product.

Save options

There are three ways to consider backing up DB2 Web Query. Which one to use depends on which one is most convenient for your situation. You may decide to do all three, depending on your business backup requirements.

Method 1 - Full System Save

Many customers perform a full system save of their system on a frequent basis using the **SAVSYS** command. In this case Web Query will be saved along with everything else. However, make sure to include *SECDTA. The IFS directories and IFS files used by Web Query are secured using Authorization Lists. Those lists are saved as part of the SAVSYS SECDTA information. In addition, license information will be saved as well.

Note: if you are using an iASP to store your Web Query information, make sure to include the appropriate save operations to also account for the iASP data.

Method 2 - Object Save

Save:

This method provides a DB2 Web Query focused save that can be incorporated into your overall IBM i save operations.

To ensure a consistent save of the necessary Web Query environment is accomplished, three aspects must be considered.

a. The set of information associated with Reports, Schedules and Run Time Environments, which is stored in objects in library **QWQREPOS.**
b. Synonyms and configuration information, which are stored under the IFS directory **/QIBM/UserData/qwebqry.**
c. Authorization Lists (AUTLs), which are used to secure the IFS objects associated with the Synonyms.

Note: before saving, make sure Web Query is ended using the ENDWEBQRY or WRKWEBQRY commands.

Saving library QWQREPOS will save the reports, schedules and runtime environments. This library can be saved using traditional IBM i save methods, usually with the **SAVLIB** command. For example:

*SAVLIB LIB(QWQREPOS) DEV(*SAVF) SAVF(QGPL/WQREPOS)*

Saving the **/QIBM/UserData/qwebqry** directory and its contents can be done using traditional save methods, usually with the **SAV** command. For example:

SAV DEV('/QSYS.LIB/QGPL.LIB/WQSYNONYMS.FILE')
*OBJ(('\QIBM\UserData\qwebqry' *INCLUDE))*

Saving the Authorization Lists (that secure the IFS objects) is a bit trickier. Authorization Lists cannot be saved and restored like other objects. Instead, their definitions must be saved and then recreated during the restore. The information to be saved is:
1. The authorization list names,
2. The users associated with each list and
3. The authority each user has for each authorization list.

To accomplish this, the best way is to create a file that contains the information, then save that file. During the restore, the file can be restored and its contents used to recreate the authorization lists.

The following is the source for a CL program that will create and populate the file described above. Copy and paste the CL source below into a source physical file, create the program using the CRTCLPGM command, and then call the CL program. The resulting file, `QGPL/WQAUTLIST`, will contain the necessary authorization list information.

Note: To get the CL program to compile, you will first need to do the following command before doing the CRTCLPGM.

```
QSYS/DSPOBJD OBJ(QSYS/QWQ*) OBJTYPE(*AUTL) +
          OUTPUT(*OUTFILE) OUTFILE(QTEMP/AUTLLIST)
```

Note2: if you prefer a different file than `QGPL/WQAUTLIST`, modify the CL source to the alternate name before compiling it.

CL source for Authorization List 'save':

```
PGM
DCL           VAR(&SQLSTRING) TYPE(*CHAR) LEN(100)
DCLF          FILE(QTEMP/AUTLLIST) OPNID(AUTLLIST)
MONMSG        MSGID(CPF0864) EXEC(GOTO CMDLBL(EOFAUTL))
MONMSG        MSGID(SQL9010)
QSYS/DSPOBJD OBJ(QSYS/QWQ*) OBJTYPE(*AUTL) +
               OUTPUT(*OUTFILE) OUTFILE(QTEMP/AUTLLIST)
QSYS/RUNSQL SQL('DROP TABLE QGPL/WQAUTLIST') COMMIT(*NONE)
QSYS/RUNSQL SQL('CREATE TABLE QGPL/WQAUTLIST (AUTLNAME +
               CHAR(10) NOT NULL, AUTLUSER CHAR(10) NOT +
               NULL, AUTH CHAR(8) NOT NULL)') COMMIT(*NONE)
READAUTL:
RCVF          OPNID(AUTLLIST)
QSYS/DSPAUTL AUTL(&AUTLLIST_ODOBNM) OUTPUT(*OUTFILE) +
               OUTFILE(QTEMP/AUTLUSERS)
CHGVAR        VAR(&SQLSTRING) VALUE('INSERT INTO +
               QGPL/WQAUTLIST SELECT ''' *CAT +
               &AUTLLIST_ODOBNM *CAT ''', OAUSR, OAOBJA +
               FROM QTEMP/AUTLUSERS')
QSYS/RUNSQL SQL(&SQLSTRING) COMMIT(*NONE)
GOTO          CMDLBL(READAUTL)
EOFAUTL:
ENDPGM
```

Once the CL program is created and run, save the resulting file `QGPL/WQAUTLIST` using the SAVOBJ command.

So in summary, to save the necessary objects:
1. Save library QWQREPOS
2. Save the IFS directory /**QIBM/UserData/qwebqry**
3. Create and populate the file `QGPL/WQAUTLIST` using the CL program defined above, then save the file `QGPL/WQAUTLIST`

Restore:

If a restore is performed, it is important to restore/recreate the Authorization Lists before restoring the IFS directory. As with the save, the 'restore' of the Authorization Lists involves recreating them from the saved information. Below is the CL source for a program to create and populate the Authorization Lists. Note that the code assumes the `QGPL/WQAUTLIST` file has been restored first. Also note that the code assumes the users (user profiles) also exist.

CL source for Authorization List 'restore':

```
PGM
DCLF          FILE(QGPL/WQAUTLIST) OPNID(AUTLLIST)
MONMSG        MSGID(CPF0864) EXEC(GOTO CMDLBL(EOFAUTL))
MONMSG        MSGID(CPF2278 CPF2282)
```

```
READAUTL:
RCVF        OPNID(AUTLLIST)
CRTAUTL     AUTL(&AUTLLIST_AUTLNAME) AUT(*EXCLUDE)
CHGOBJOWN   OBJ(&AUTLLIST_AUTLNAME) OBJTYPE(*AUTL) +
              NEWOWN(QWQADMIN)
IF          COND(&AUTLLIST_AUTLUSER = '*PUBLIC') +
              THEN(GOTO CMDLBL(READAUTL))
ADDAUTLE    AUTL(&AUTLLIST_AUTLNAME) +
              USER(&AUTLLIST_AUTLUSER) AUT(&AUTLLIST_AUTH)
GOTO        CMDLBL(READAUTL)
EOFAUTL:
ENDPGM
```

Restoring the **/QIBM/UserData/qwebqry** directory and its contents can be done using traditional restore methods, usually with the **RST** command. For example:

RST DEV('/QSYS.LIB/QGPL.LIB/WQSYNONYMS.FILE')
OBJ((‘QIBM\UserData\qwebqry' *INCLUDE
'QIBM\UserData\qwebqry')) ALWOBJDIF(*ALL)

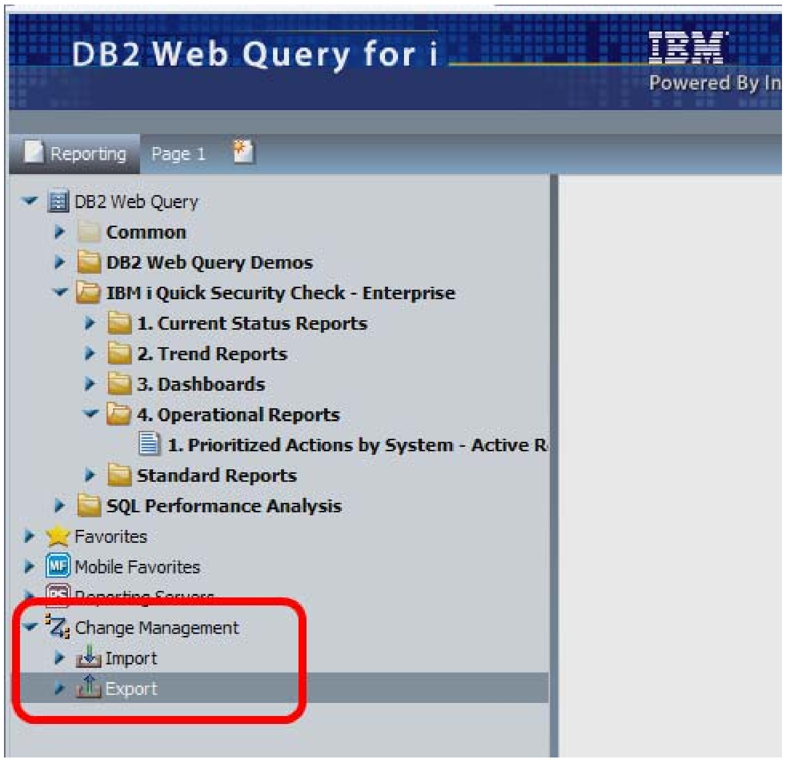
Restoring library QWQREPOS can be done using traditional IBM i restore methods, usually with the **RSTLIB** command. For example:
*RSTLIB RSTLIB(QWQREPOS) DEV(*SAVF) SAVF(QGPL/WQREPOS)*

Method 3 – Export (and Import)

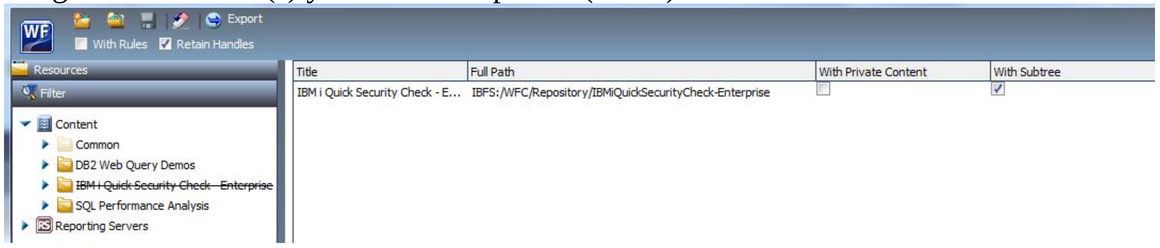
The final approach is to **export** from DB2 Web Query's Change Management facility and save the resulting export file. An import can be done later if a 'restore' is required. The advantage of an export/import method is that the import will automatically populate the underlying objects as necessary: QWQREPOS and the IFS objects. Another advantage is that a subset of information can be chosen, for example exporting only reports that have changed since the last major save.

To Export:

A. Within the DB2 Web Query Portal, find the **Change Management** folder, expand it, right click on **Export** and select **New Scenario. Note that you must have ADMIN rights to use DB2 Web Query Change Management functions.**



Drag over the folder(s) you wanted exported (saved).



Title	Full Path	With Private Content	With Subtree
IBM i Quick Security Check - E…	IBFS:/WFC/Repository/IBMiQuickSecurityCheck-Enterprise	☐	✓

When finished, select the WF in the upper left box and click **Save**.

Once this is complete, the export 'package' will be in a subdirectory under the Web Query directory structure. Save this export (directory) and everything under it. The export can be found under directory **/QIBM/UserData/qwebqry/base80/cm/export.**

Saving the export directory and its contents can be done using traditional save methods, usually with the **SAV** command. For example:

```
        SAV DEV('/QSYS.LIB/QGPL.LIB/WQEXPORT.FILE')
        OBJ(('\QIBM\UserData\qwebqry\base80\cm\export' *INCLUDE))
```

Alternatively, you can map a network drive and copy the export directory and contents as well.

Restore:

Should you ever need to import from this exported package, load the exported directory back into the **/QIBM/UserData/qwebqry/base80/cm/import** directory, then go into the same Web Query interface used for the export but instead select **Import**.

Restoring the exported directory and its contents into the import directory can be done using traditional restore methods, usually with the **RST** command. For example:

```
        RST DEV('/QSYS.LIB/QGPL.LIB/WQEXPORT.FILE')
        OBJ((‘QIBM\UserData\qwebqry\base80\cm\export' *INCLUDE
        'QIBM\UserData\qwebqry\base80\cm\import ')) ALWOBJDIF(*ALL)
```

Alternatively, if you had mapped a network drive for the export, you can use that network drive approach to copy the exported directory contents into the import directory as well.