```
--
-- A Practical Guide to Authority Collection
-- Steve Riedmueller & Carol Woodbury
-- Kisco Systems LLC
--

--Extract/Safeguard Authority Collection Data
--Authority collection data can be “dumped” into a table/PF
--Write all AUTHORITY_COLLECTION_LIBRARIES entries to an outfile
CREATE TABLE SRIEDMUE.AC_OUTFILE AS
            (SELECT *
                    FROM QSYS2.AUTHORITY_COLLECTION_LIBRARIES) WITH DATA;
--View the data in the outfile
SELECT * FROM SRIEDMUE.AC_OUTFILE;


--Start collection for user SRIED, but only for one library:
CL: STRAUTCOL USRPRF(SRIED) LIBINF((PRODLIB));
--Start collection for user SRIED, but only for *DTAARA in two libraries:
CL: STRAUTCOL USRPRF(SRIED) LIBINF((PRODLIB) (DATALIB)) OBJTYPE(*DTAARA);
--Start collection for user SRIED, but only for the IFS:
CL: STRAUTCOL USRPRF(SRIED) INCFSOBJ(*ALL);
--Start a full collection for user SRIED (all libraries and IFS):
CL: STRAUTCOL USRPRF(SRIED) LIBINF(*ALL) INCFSOBJ(*ALL);


--1. Set the flag "on" for the object:
CL: CHGAUTCOL OBJ('/QSYS.LIB/SRIED.LIB/QCLSRC.FILE') AUTCOLVAL(*OBJINF);
--2. Start the object-based collection function globally (no harm if already active)
CL: STRAUTCOL TYPE(*OBJAUTCOL);


--1. Set the flag "on" for all objects in a library:
CL: CHGAUTCOL OBJ('/QSYS.LIB/SRIED.LIB') AUTCOLVAL(*OBJINF) SUBTREE(*ALL);
--2. Start the object-based collection function globally (no harm if already active)
CL: STRAUTCOL TYPE(*OBJAUTCOL);


--1. Set the flag "on" for an IFS directory and all objects therein:
CL: CHGAUTCOL OBJ('/home/sried/') AUTCOLVAL(*OBJINF) SUBTREE(*ALL);
--2. Start the object-based collection function globally (no harm if already active)
CL: STRAUTCOL TYPE(*OBJAUTCOL);


--Which records to omit?
--Omit entries related to operating system programs that adopt authority:
SELECT *
    FROM QSYS2.AUTHORITY_COLLECTION
    WHERE AUTHORIZATION_NAME = 'SRIED'
          AND (ADOPTING_PROGRAM_SCHEMA IS NULL
```

```
              OR ADOPTING_PROGRAM_SCHEMA <> 'QSYS')
    ORDER BY CHECK_TIMESTAMP;


--Authority Reduction – find high-powered profiles
--Profiles with *ALLOBJ special authority
SELECT AUTHORIZATION_NAME,
       STATUS,
       SPECIAL_AUTHORITIES,
       GROUP_PROFILE_NAME,
       SUPPLEMENTAL_GROUP_LIST,
       TEXT_DESCRIPTION
    FROM QSYS2.USER_INFO_BASIC
    WHERE SPECIAL_AUTHORITIES LIKE '%*ALLOBJ%';
--Profiles that have *ALLOBJ or are members of a group having *ALLOBJ
SELECT AUTHORIZATION_NAME,
       STATUS,
       SPECIAL_AUTHORITIES,
       GROUP_PROFILE_NAME,
       SUPPLEMENTAL_GROUP_LIST,
       TEXT_DESCRIPTION
    FROM QSYS2.USER_INFO_BASIC
    WHERE SPECIAL_AUTHORITIES LIKE '%*ALLOBJ%'
          OR AUTHORIZATION_NAME IN (SELECT USER_PROFILE_NAME
                  FROM QSYS2.GROUP_PROFILE_ENTRIES
                  WHERE GROUP_PROFILE_NAME IN (SELECT AUTHORIZATION_NAME
                              FROM QSYS2.USER_INFO_BASIC
                              WHERE SPECIAL_AUTHORITIES LIKE '%*ALLOBJ%'
                                    AND GROUP_ID_NUMBER <> '0'));
--Authority Reduction – start authority collection
--Start authority collection for the target user profile:
CL: STRAUTCOL TYPE(*USRPRF) USRPRF(SUPERUSER) LIBINF(*ALL) INCFSOBJ(*ALL);
--Authority Reduction – analyze the data
--List the required authority for all objects that the user touched
SELECT DISTINCT SYSTEM_OBJECT_SCHEMA,
                SYSTEM_OBJECT_NAME,
                SYSTEM_OBJECT_TYPE,
                CHECK_ANY_AUTHORITY,
                AUTHORITY_CHECK_SUCCESSFUL,
                DETAILED_REQUIRED_AUTHORITY,
                DETAILED_CURRENT_AUTHORITY,
                AUTHORITY_SOURCE
    FROM QSYS2.AUTHORITY_COLLECTION
    WHERE AUTHORIZATION_NAME = 'SUPERUSER'
          AND (ADOPTING_PROGRAM_SCHEMA IS NULL
              OR ADOPTING_PROGRAM_SCHEMA <> 'QSYS')
    ORDER BY SYSTEM_OBJECT_SCHEMA,
             SYSTEM_OBJECT_NAME;
--Authority Reduction – grant authority
--Grant the required object authority via authorization list, group, or private
```

```
authority
CL: GRTOBJAUT OBJ(SRSAMPLE)   OBJTYPE(*LIB)  USER(SUPERUSER) AUT(*EXECUTE);
CL: GRTOBJAUT OBJ(CREATEEMP)  OBJTYPE(*PGM)  USER(SUPERUSER) AUT(*EXECUTE);
CL: GRTOBJAUT OBJ(DELETEEMP)  OBJTYPE(*PGM)  USER(SUPERUSER) AUT(*EXECUTE);
CL: GRTOBJAUT OBJ(DEPARTMENT) OBJTYPE(*FILE) USER(SUPERUSER) AUT(*OBJOPR *READ);
CL: GRTOBJAUT OBJ(EMPLOYEE)   OBJTYPE(*FILE) USER(SUPERUSER) AUT(*OBJOPR *READ *ADD
*DLT);
CL: GRTOBJAUT OBJ(VIEWDEPT)   OBJTYPE(*PGM)  USER(SUPERUSER) AUT(*EXECUTE);
CL: GRTOBJAUT OBJ(VIEWEMP)    OBJTYPE(*PGM)  USER(SUPERUSER) AUT(*EXECUTE);
--Revoke *ALLOBJ special authority or remove the group membership.
CL: CHGUSRPRF USRPRF(SUPERUSER) SPCAUT(*NONE);
CL: CHGUSRPRF USRPRF(SUPERUSER) GRPPRF(*NONE);
--Authority Reduction – review the results
--List the required authority for all objects that the user touched
SELECT DISTINCT SYSTEM_OBJECT_SCHEMA,
                SYSTEM_OBJECT_NAME,
                SYSTEM_OBJECT_TYPE,
                CHECK_ANY_AUTHORITY,
                AUTHORITY_CHECK_SUCCESSFUL,
                DETAILED_REQUIRED_AUTHORITY,
                DETAILED_CURRENT_AUTHORITY,
                AUTHORITY_SOURCE
    FROM QSYS2.AUTHORITY_COLLECTION
    WHERE AUTHORIZATION_NAME = 'SUPERUSER'
          AND (ADOPTING_PROGRAM_SCHEMA IS NULL
              OR ADOPTING_PROGRAM_SCHEMA <> 'QSYS')
          AND CHECK_TIMESTAMP > '2024-04-01 14:00:00'
    ORDER BY SYSTEM_OBJECT_SCHEMA,
             SYSTEM_OBJECT_NAME;
--Lock down a Library or File – start AC
--Start authority collection for the file:
CL: CHGAUTCOL OBJ('/QSYS.LIB/SRSAMPLE.LIB/SALARY.FILE') AUTCOLVAL(*OBJINF);
CL: STRAUTCOL TYPE(*OBJAUTCOL);
--Lock down a Library or File – analyze the data
SELECT DISTINCT AUTHORIZATION_NAME,
                SYSTEM_OBJECT_SCHEMA,
                SYSTEM_OBJECT_NAME,
                SYSTEM_OBJECT_TYPE,
                CHECK_ANY_AUTHORITY,
                AUTHORITY_CHECK_SUCCESSFUL,
                DETAILED_REQUIRED_AUTHORITY,
                DETAILED_CURRENT_AUTHORITY,
                AUTHORITY_SOURCE
    FROM QSYS2.AUTHORITY_COLLECTION_LIBRARIES
    WHERE SYSTEM_OBJECT_SCHEMA = 'SRSAMPLE'
          AND SYSTEM_OBJECT_NAME = 'SALARY'
          AND (ADOPTING_PROGRAM_SCHEMA IS NULL
              OR ADOPTING_PROGRAM_SCHEMA <> 'QSYS');
```

```
--Lock down an IFS path – start authority collection
CL: CHGAUTCOL OBJ('/coredata/') AUTCOLVAL(*OBJINF) SUBTREE(*ALL);
CL: STRAUTCOL TYPE(*OBJAUTCOL);
--Lock down an IFS path – analyze the data
SELECT DISTINCT AUTHORIZATION_NAME,
                PATH_NAME,
                CHECK_ANY_AUTHORITY,
                AUTHORITY_CHECK_SUCCESSFUL,
                DETAILED_REQUIRED_AUTHORITY,
                DETAILED_CURRENT_AUTHORITY,
                AUTHORITY_SOURCE
    FROM QSYS2.AUTHORITY_COLLECTION_FSOBJ
    WHERE PATH_NAME LIKE '/coredata%'
    ORDER BY AUTHORIZATION_NAME,
             PATH_NAME;
--Lock down an IFS path – secure the path
--Grant the required private authority for SUPERUSER:
CL: CHGAUT OBJ('/coredata') USER(SUPERUSER) DTAAUT(*RX);
CL: CHGAUT OBJ('/coredata/process.csv') USER(SUPERUSER) DTAAUT(*RW);
--Lock out *PUBLIC for the whole subtree:
CL: CHGAUT OBJ('/coredata') USER(*PUBLIC) DTAAUT(*EXCLUDE) OBJAUT(*NONE)
SUBTREE(*ALL);
--Lock down an IFS path – review the results
SELECT DISTINCT AUTHORIZATION_NAME,
                PATH_NAME,
                CHECK_ANY_AUTHORITY,
                AUTHORITY_CHECK_SUCCESSFUL,
                DETAILED_REQUIRED_AUTHORITY,
                DETAILED_CURRENT_AUTHORITY,
                AUTHORITY_SOURCE
    FROM QSYS2.AUTHORITY_COLLECTION_FSOBJ
    WHERE PATH_NAME LIKE '/coredata%'
          AND CHECK_TIMESTAMP > '2024-02-05 19:00:00'
    ORDER BY AUTHORIZATION_NAME,
             PATH_NAME;


--Who is using a Netserver fileshare?
--Enable authority collection on the shared directory:
CL: CHGAUTCOL OBJ('/home') AUTCOLVAL(*OBJINF) SUBTREE(*NONE);
--Review AC data for that path (Netserver access handled by QZLSFILE% jobs)
SELECT CHECK_TIMESTAMP,
       AUTHORIZATION_NAME,
       JOB_NAME,
       PATH_NAME
    FROM QSYS2.AUTHORITY_COLLECTION_FSOBJ
    WHERE JOB_NAME LIKE 'QZLSFILE%'
          AND UPPER(PATH_NAME) LIKE '/HOME/%';
```

```
--Who is using an object in a particular library?
--Turn on authority collection for all 3 copies of the program:
CL: CHGAUTCOL OBJ('/qsys.lib/applib.lib/samplepgm.pgm') AUTCOLVAL(*OBJINF);
CL: CHGAUTCOL OBJ('/qsys.lib/applib2.lib/samplepgm.pgm') AUTCOLVAL(*OBJINF);
CL: CHGAUTCOL OBJ('/qsys.lib/oldpgms.lib/samplepgm.pgm') AUTCOLVAL(*OBJINF);
--Check the collected data after some time:
SELECT DISTINCT AUTHORIZATION_NAME,
                SYSTEM_OBJECT_SCHEMA,
                JOB_NAME,
                JOB_USER,
                JOB_NUMBER,
                "CURRENT_USER"
    FROM QSYS2.AUTHORITY_COLLECTION_LIBRARIES
    WHERE SYSTEM_OBJECT_NAME = 'SAMPLEPGM';


--Who is using a system command like ENDSBS?
--Turn on authority collection for the command:
CL: CHGAUTCOL OBJ('/qsys.lib/endsbs.cmd') AUTCOLVAL(*OBJINF);
CL: STRAUTCOL TYPE(*OBJAUTCOL);
--Check the data after some time:
SELECT DISTINCT AUTHORIZATION_NAME,
                SYSTEM_OBJECT_SCHEMA,
                JOB_NAME,
                JOB_USER,
                JOB_NUMBER,
                "CURRENT_USER"
    FROM QSYS2.AUTHORITY_COLLECTION_LIBRARIES
    WHERE SYSTEM_OBJECT_NAME = 'SAMPLEPGM';


--Research AF entries in the audit journal - QSYS
SELECT USER_NAME,
       QUALIFIED_JOB_NAME,
       VIOLATION_TYPE_DETAIL,
       OBJECT_LIBRARY,
       OBJECT_NAME,
       OBJECT_TYPE
    FROM TABLE (
            SYSTOOLS.AUDIT_JOURNAL_AF()
         );
--Turn on authority collection for this file:
CL: CHGAUTCOL OBJ('/qsys.lib/extsavf.lib/saveqtemp.file') AUTCOLVAL(*OBJINF);
CL: STRAUTCOL TYPE(*OBJAUTCOL);
--Review the authority collection entries:
SELECT AUTHORIZATION_NAME,
       SYSTEM_OBJECT_NAME,
       SYSTEM_OBJECT_SCHEMA,
       SYSTEM_OBJECT_TYPE,
       AUTHORITY_CHECK_SUCCESSFUL,
```

```
       DETAILED_REQUIRED_AUTHORITY,
       DETAILED_CURRENT_AUTHORITY
    FROM QSYS2.AUTHORITY_COLLECTION_LIBRARIES
    WHERE SYSTEM_OBJECT_NAME = 'SAVEQTEMP'
    ORDER BY CHECK_TIMESTAMP;
--Grant the required authority to the user profile:
CL: GRTOBJAUT OBJ(EXTSAVF/SAVEQTEMP) OBJTYPE(*FILE) USER(PROFILEX) AUT(*OBJOPR *READ
*ADD *EXECUTE);


--Research AF entries in the audit journal - IFS
SELECT USER_NAME,
       QUALIFIED_JOB_NAME,
       VIOLATION_TYPE_DETAIL,
       OBJECT_TYPE,
       PATH_NAME
    FROM TABLE (
            SYSTOOLS.AUDIT_JOURNAL_AF()
         );
 --Turn on authority collection for all files in this directory:
CL: CHGAUTCOL OBJ('/web/development/olrdev/logs/access/*') AUTCOLVAL(*OBJINF);
--Research AF entries in the audit journal - IFS
SELECT DISTINCT AUTHORIZATION_NAME,
                PATH_NAME,
                AUTHORITY_CHECK_SUCCESSFUL,
                DETAILED_REQUIRED_AUTHORITY,
                DETAILED_CURRENT_AUTHORITY,
                AUTHORITY_SOURCE
    FROM QSYS2.AUTHORITY_COLLECTION_FSOBJ
    WHERE PATH_NAME LIKE '/web/development/olrdev/logs/access/%'
          AND AUTHORITY_CHECK_SUCCESSFUL = 0
    ORDER BY AUTHORIZATION_NAME,
             PATH_NAME;
--Grant the required authority to the user profile:
CL: CHGAUT OBJ('/web/development/olrdev/logs/access/*') USER(QTMHHTTP)
OBJAUT(*OBJEXIST);


--Investigate FTP Processes and Activity
--Turn on authority collection for the user profile:
CL: STRAUTCOL USRPRF(FTPUSER) INCFSOBJ(*ALL);
--Check the authority collection entries for this user:
SELECT DISTINCT JOB_NAME,
                SYSTEM_OBJECT_SCHEMA,
                SYSTEM_OBJECT_NAME,
                SYSTEM_OBJECT_TYPE,
                CHECK_ANY_AUTHORITY,
                AUTHORITY_CHECK_SUCCESSFUL,
                REQUIRED_AUTHORITY,
                DETAILED_REQUIRED_AUTHORITY
```

```
    FROM QSYS2.AUTHORITY_COLLECTION
    WHERE AUTHORIZATION_NAME = 'FTPUSER'
          AND (ADOPTING_PROGRAM_SCHEMA IS NULL
              OR ADOPTING_PROGRAM_SCHEMA <> 'QSYS')
          AND JOB_NAME LIKE 'QTFTP%'
    ORDER BY SYSTEM_OBJECT_SCHEMA,
             SYSTEM_OBJECT_NAME;


--What process/user is accessing a file in the root?
--Find recently-used CSV files in the system root directory:
SELECT PATH_NAME,
       OBJECT_TYPE,
       CREATE_TIMESTAMP,
       ACCESS_TIMESTAMP,
       DATA_CHANGE_TIMESTAMP
    FROM TABLE (
            QSYS2.IFS_OBJECT_STATISTICS(START_PATH_NAME => '/', SUBTREE_DIRECTORIES
=> 'NO')
         )
    WHERE UPPER(PATH_NAME) LIKE '%.CSV'
    ORDER BY ACCESS_TIMESTAMP DESC;
--Turn on authority collection for all of the files in this directory:
CL: CHGAUTCOL OBJ('/downloadfile.csv') AUTCOLVAL(*OBJINF);
CL: STRAUTCOL TYPE(*OBJAUTCOL);
--View the authority collection data:
SELECT DISTINCT JOB_NAME,
                AUTHORIZATION_NAME,
                PATH_NAME,
                CHECK_ANY_AUTHORITY,
                AUTHORITY_CHECK_SUCCESSFUL,
                DETAILED_REQUIRED_AUTHORITY,
                DETAILED_CURRENT_AUTHORITY,
                AUTHORITY_SOURCE
    FROM QSYS2.AUTHORITY_COLLECTION_FSOBJ
    WHERE PATH_NAME = '/downloadfile.csv'
    ORDER BY AUTHORIZATION_NAME,
             PATH_NAME;


--Which user profiles have collections running?
--user profiles that have authority collection turned on
SELECT *
    FROM QSYS2.USER_INFO
    WHERE AUTHORITY_COLLECTION_ACTIVE = 'YES';
--user profiles with authority collection turned off, but a repository exists
SELECT *
    FROM QSYS2.USER_INFO
    WHERE AUTHORITY_COLLECTION_ACTIVE = 'NO'
          AND AUTHORITY_COLLECTION_REPOSITORY_EXISTS = 'YES';
```

```
--Which "Native" objects have collection enabled?
SELECT OBJLIB,
       OBJNAME,
       OBJTYPE,
       OBJATTRIBUTE,
       AUTHORITY_COLLECTION_VALUE
    FROM TABLE (
            QSYS2.OBJECT_STATISTICS(OBJECT_SCHEMA => 'STEVER', OBJTYPELIST =>
'*FILE', OBJECT_NAME => 'TESTFILE')
        );


SELECT DISTINCT SYSTEM_OBJECT_SCHEMA,
                SYSTEM_OBJECT_NAME,
                SYSTEM_OBJECT_TYPE
    FROM QSYS2.AUTHORITY_COLLECTION_LIBRARIES;
--An Alternative Method for Checking
--objects having AC repository entries, and the current OBJAUTCOL setting
WITH OBJECTS (LIBNAME, OBJNAME, OBJTYPE) AS (
        SELECT DISTINCT SYSTEM_OBJECT_SCHEMA,
                        SYSTEM_OBJECT_NAME,
                        SYSTEM_OBJECT_TYPE
            FROM QSYS2.AUTHORITY_COLLECTION_LIBRARIES
    )
    SELECT OBJECTS.LIBNAME,
           OBJECTS.OBJNAME,
           OBJECTS.OBJTYPE,
           AUTHORITY_COLLECTION_VALUE
        FROM OBJECTS,
             LATERAL (
                 SELECT AUTHORITY_COLLECTION_VALUE
                     FROM TABLE (
                             QSYS2.OBJECT_STATISTICS(
                                 OBJECT_SCHEMA => OBJECTS.LIBNAME, OBJTYPELIST =>
OBJECTS.OBJTYPE, OBJECT_NAME => OBJECTS.OBJNAME)
                         )
             );


--Which IFS objects have collection enabled?
SELECT PATH_NAME,
       OBJECT_TYPE,
       AUTHORITY_COLLECTION_VALUE
    FROM TABLE (
            QSYS2.IFS_OBJECT_STATISTICS(START_PATH_NAME => '/coredata/process.csv')
        );
--An Alternative Method for Checking
--IFS files having AC repository entries, and the current OBJAUTCOL setting
```

```
WITH IFSFILES (PATH_NAME) AS (
        SELECT DISTINCT PATH_NAME
            FROM QSYS2.AUTHORITY_COLLECTION_FSOBJ
    )
    SELECT IFSFILES.PATH_NAME,
           AUTHORITY_COLLECTION_VALUE
        FROM IFSFILES,
             LATERAL (
                 SELECT AUTHORITY_COLLECTION_VALUE
                     FROM TABLE (
                             QSYS2.IFS_OBJECT_STATISTICS(START_PATH_NAME =>
IFSFILES.PATH_NAME, SUBTREE_DIRECTORIES => 'NO')
                         )
             );


--Deleting authority collection data
--Deleting entries for a user profile:
CL: ENDAUTCOL USRPRF(SRIEDMUE);
CL: DLTAUTCOL TYPE(*USRPRF) USRPRF(SRIED);
--Deleting entries for an entire IFS subtree:
CL: ENDAUTCOL TYPE(*OBJAUTCOL);
CL: DLTAUTCOL TYPE(*OBJ) OBJ('/home/sried/') SUBTREE(*ALL);
--Deleting entries for a specific object:
CL: ENDAUTCOL TYPE(*OBJAUTCOL);
CL: DLTAUTCOL TYPE(*OBJ) OBJ('/QSYS.LIB/SRIED.LIB/QCLSRC.FILE');
```