iProDeveloper

print | close

Programming for the V5R4 Check Password Exit Point Program

System iNetwork Systems Management Newsletter

Carsten Flensburg

Carsten Flensburg

Wed, 04/15/2009 (All day)

This newsletter has on previous occasions covered the Password Validation Exit Point (QIBM_QSY_VLD_PASSWRD) which enables you to block password changes to do not conform to the password rules implemented by the exit program registered to this exit point. I have included links to a couple of articles explaining this exit point in more details, at the end of this article. An important shortcoming of the QIBM_QSY_VLD_PASSWRD exit point is however the fact that it is only evoked when a user changes his or her own password using the Change Password (CHGPWD) command, the Change Password (QSYCHGPW) API or the System i Access dialogue box being displayed in case of an expired password. If instead a password is changed running the Change User Profile (CHGUSRPRF) or Create User Profile (CRTUSRPRF) command, this will circumvent the exit point and consequently deprive you of the control provided by the exit point in question.

To close this hole IBM introduced with release 5.4 IBM a new Check Password Exit Point (QIBM_QSY_CHK_PASSWRD). Contrary to the QIBM_QSY_VLD_PASSWRD exit point the new exit point does not allow you to actually block the password change, but instead registers the result of the exit program's password check process to the system audit journal (QAUDJRN). You then have the option to interrogate the system audit journal to ensure that any password not conforming to your password composition rules goes undetected and deal with the culprit according to your company security policy. Anyway, the check password exit program receives two parameters; one input parameter data structure containing information about the user profile and password being changed. This data structure's format is identical to the one employed by the QIBM_QSY_VLD_PASSWRD exit point. The second parameter provides for the exit program's return value; the value '0' indicates that the new password passed the test successfully, the value '1' that the test was not passed. Given that you have configured your system's audit system values appropriately (see article link below) you can look up audit journal entries having journal code 'T' and entry type 'CP' to verify the outcome.

In addition to the registration of the password check result stored in the CP audit journal entry, the message CPI22AE can be found in the job log of the job performing a password change that does not conform to the password composition rules implemented by the password check exit program:

```
Message ID . . . . . . . :   CPI22AE



Message . . . . :   Password does not meet all password rules.
```

```
  Cause . . . . . :   The new password does not meet the password
rules
    specified by the QPWDxxxxxx system values, or one of the programs
registered
    for the QIBM_QSY_CHK_PASSWRD exit point has indicated that the
password does
    not meet the installation specific rules.



  Technical description . . . . . . . . :   An exception was signaled
by
    QSYPWDCR that indicates the new password value does not conform to
the
    password composition rules.
```

The following is a documentation of the password composition conformance registration in the system audit journal and can be viewed in its full context in the System i Security Reference as of release 5.4:

```
  Journal code 'T'; Entry type 'CP'; Output format 'J5':

  Offset  Type        Field
    642   Char(1)     Password Changed; Y=Yes
    967   Char(10)    Password Composition Conformance

    Indicates whether the new password conforms to the password
composition rules:

    *PASSED   Checked and conforms.

    *SYSVAL   Checked but does not conform because of a system value
based rule.

    *EXITPGM  Checked but does not conform because of an exit program
response.

    *NONE     Not checked; *NONE was specified for the new password.
    This field has meaning only when the Password Changed field
contains a Y.
```

After proper registration of the password check exit program (see below), the exit program will be called whenever a user profile's password is changed, irrespective of the command or API performing the password change. More than one program can be registered to the password check exit point and the check programs will be called in turn until all check programs have been called - or a until a negative return code is received. The old password value will always be set to *NOPWD and the old password length set to 12 (since the password value is sent in a unicode character set).

The password check exit program allows you to perform password checks in addition to those supported by the password system values including the recent release 6.1 enhancements introduced with the QPWDRULES system value. Given these circumstances the password check exit program typically will implement local and individual password rules not supported by the aforementioned system values. For the sake of demonstrating one such individual type of check, I've employed the System i spelling aid dictionary object type (*SPADCT) and the Check Spelling (QTWCHKSP) API in the sample exit program that I include with this article. Using a user created spelling aid dictionary allows me to create a collection of words that I would like to disallow being used as passwords, and then use the QTWCHKSP API in the password check exit program to perform the check against my special non-password dictionary. To avoid the most common methods of masking such passwords I remove appended digits and underscores before performing the dictionary check and only if the check fails, i.e. the word was not found in the dictionary, will I let the check pass. Another more generic approach would be using for example regular expressions to match for specific or complex patterns that should either be followed or avoided.

Anyway, since the primary intention of the sample exit program is to offer a starting point for you to adapt and implement password rules making sense given your specific circumstances and requirements, let's stick with the former and simpler method, so here's how to create a user spelling aid dictionary:

1. Create a source physical file in your preferred library using the following command:

```
CRTSRCPF FILE(QGPL/QDCTSRC)
             RCDLEN(92)
             MBR(CBX708S)
             CCSID(*HEX)
```

2. Add a member to the source file using the command below:

```
 ADDPFM FILE(QGPL/QDCTSRC)
         MBR(CBX708D)
         SRCTYPE(SPADCT)
```

3. Enter the words that should be disallowed as passwords into the member using an editor of your choice, you can specify one word per line or multiple words per line, as long as the words are separated by blanks, as in the following example:

```
  JANUARY FEBRUARY MARCH APRIL MAY
  JUNE JULY AUGUST SEPTEMBER
  OCTOBER NOVEMBER DECEMBER
```

Please keep all words in upper case to ensure case insensitive matching; the exit program upper cases all passwords before calling the spell check API.

4. Save the member and compile the spelling aid dictionary using PDM option 14 or the following command:

```
 CRTSPADCT SPADCT(QGPL/CBX708D)
            SRCFILE(QGPL/QRPGLESRC)
            SRCMBR(CBX708D)
            LNGATR(*NONE)
            REPLACE(*NO)
```

Note that the spelling aid dictionary must be located in the same library as your password check exit program in order for the exit program to access it so you'll need to replace library QGPL in the above commands with the name of the library you intend to use for the password check exit program. As for the regular expression approach mentioned earlier, I've included links to articles explaining this function below in case you're interested in pursuing that option. You'll also find a link to the CBX708 source member providing the code for the check password exit program. Once you've compiled the exit program, you can register it using the following command:

```
ADDEXITPGM  EXITPNT( QIBM_QSY_CHK_PASSWRD )
                FORMAT( CHKP0100 )
                PGMNBR( *LOW )
                PGM( /CBX708 )
                TEXT( 'Password check exit program' )
```

In case you're tempted to try and run the exit program in the debugger when fired by the exit point, you should be aware that the system will not allow the program to be interrupted while executing. In order to see the program in action you'll therefore need to call it with dummy parameters from either the command line or setting up another program to perform the call. If you've followed the instructions so far the password check exit program will cause the CP journal entry to contain the composition conformance value *EXITPGM if a new password for example specifies AUGUST_22 or MARCH011. In an upcoming article I will present a new Print Password Audit Report (PRTPWDAUD) command to help you monitor and report password change activity on your system. Until then you can use the release 5.4 Copy Audit Journal Entries (CPYAUDJRNE) command - I've included a link to an article written by Dan Riehl explaining the CPYAUDJRNE command below.

You can download the code for the sample exit program here.

IBM documentation:

Exit point QIBM_QSY_CHK_PASSWRD documentation

Audit journal entry code/type 'T'/'CP' documentation

Check Spelling (QTWCHKSP) API

Previously published articles:

Common Sense Security Auditing (QAUDJRN):

Creating and Managing the QAUDJRN Journal

Regular Expressions and RPG

Using Regular Expressions in ILE RPG

Validate E-Mail Address with a Regular Expression

Password Change Blocking for V5R4 and Earlier

i/OS 6.1 Password Rules System Value (QPWDRULES)

Additional Validation of New Passwords

Extracting Information from QAUDJRN using CPYAUDJRNE

Source URL: http://iprodeveloper.com/systems-management/programming-v5r4-check-password-exit-point-program