


[print](#) | [close](#)

APIs by Example: Authorization List APIs and Secured Objects, Part 2

[System iNetwork Programming Tips Newsletter](#)

[Carsten Flensburg](#)

Carsten Flensburg

Thu, 08/17/2006 (All day)

As I discussed last time in APIs by Example, currently no command exists to list the IFS directory objects secured by a specific authorization list. To fill that gap, I provide the Display Authorization List IFS Objects (DSPAUTLIFS) command in this article. To do that, I use the sample program for the List Objects Secured by Authorization List (QSYLATLO) API that I presented in part one of this article (a link to part one is at the end of this article).

The QSYLATLO API returns the secured object information to a user space, and last time, I described in detail how that information is retrieved. So to create the DSPAUTLIFS command, I simply add a UIM list panel group and, for each IFS object returned, write the object information to the list.

Here's what the DSPAUTLIFS command prompt looks like:

```

                Display Auth List IFS Objects (DSPAUTLIFS)
Type choices, press Enter.
Authorization list . . . . . Name
Output . . . . . *          *, *PRINT

```

Running the command DSPAUTLIFS AUTL(INEWS) presents the following list on my system:

```

                Display Authorization List IFS Objects
                                WYNDHAMW
                                13-08-06 10:57:49
Authorization List . : INEWS      Owner . . . . . : CFLEN
Library . . . . . : QSYS      Primary group . . : *NONE
                                Primary
Object                Type      Owner      group

```

In the event that an IFS object name is too long to fit in the available space in the list, only the right-most part of the name is displayed, and a < (less than) sign to the left of the column indicates the insufficient space. Placing the cursor on the list entry and pressing function key F22 then brings up a window with the entire object name:

```

.....
:                Display Entire Name                :
: /QOpenSys/QUSER/CF101/iSeriesNews/CBX0012.TXT      :

```



Full help text for both command and list panel group is included to explain all details. Please note that if a very large number of objects are secured by an authorization list, the situation could arise in which the object list occupies more space than a user space can hold (approximately 16 MB). In that case, the DSPAUTLIFS command returns the escape message CPF3CAA "List is too large for user space &1."

Because the UIM list panel group has a similar space constraint, I also need to keep the space usage for each list entry to a minimum. So instead of storing the full IFS object name in the list, I store only the object's file ID, which I retrieve using the Get Attributes (QpolGetAttr) API. When the user presses function key F22 to display the entire object name, the Get Path from File ID (QpolGetPathFromFileID) API is called to get the IFS object name from the file ID. The file ID is, however, stored only for objects that have a name longer than the 42 bytes available in the list column. For shorter names the (full) object name stored in the list column is used by the Display Entire Name function.

Notice that I have adapted the list panel and appearance to the native IBM Display Authorization List Objects (DSPAUTLOBJ) command. This also applies to including the qualification of the authorization list object. Authorization lists are always located in the QSYS system library, and you cannot actually specify the library on any of the authorization list commands. Yet IBM still includes the library as a constant in the authorization list command definitions and displays it on some of the authorization list panels, so I decided to follow that behavior.

To manipulate the authorization list of an IFS directory object, a number of methods and commands are available:

1. The Work with Authority (WRKAUT) command lets you change the authority of one specific IFS object. To change the authorization list of the object, press F15 from the initially displayed panel.
2. The Work with Object Links (WRKLNK) command displays a list of IFS directory objects. From the list panel, you specify option 9 to invoke the WRKAUT command for the selected list entry.
3. The Change Authority (CHGAUT) command lets you change the authorization list for one or more objects that match the specified pattern.
 - The following example changes the authorization list of all objects in the iSeriesNews directory to INEWS: `CHGAUT OBJ('/qopensys/QUSER/iSeriesNews/*') AUTL (INEWS)`
 - Likewise, you can remove all authorization lists for the same objects with the following command: `CHGAUT OBJ('/qopensys/QUSER/iSeriesNews/*') AUTL(*NONE)`

The DSPAUTLIFS command is only one of many home-grown utilities for the IFS that have been presented in the iSeries Network newsletters over recent years. Here are some examples of previously published IFS commands:

Display IFS Object Locks (DSPIFSLCK) command:

<http://www2.systeminetwork.com/article.cfm?id=18313>

A CLRPFM Utility for the IFS:

<http://www2.systeminetwork.com/article.cfm?id=18408>

Change IFS Attributes (CHGIFSATR) command:

<http://www2.systeminetwork.com/article.cfm?id=19583>

Display Attributes (DSPATR) command:

<http://www2.systeminetwork.com/article.cfm?id=19719>

Display Job IFS Object Locks (DSPJOBIFSL) command:

<http://www2.systeminetwork.com/article.cfm?id=52214>

Delete IFS Tree (DELTREE) command:

<http://www2.systeminetwork.com/article.cfm?id=52993>

This APIs by Example includes the following sources:

```
CBX160    -- Display Authorization List IFS Objects - CPP
CBX160E   -- Display Authorization List IFS Objects - UIM Exit
CBX160H   -- Display Authorization List IFS Objects - Help
CBX160P   -- Display Authorization List IFS Objects - Panel Group
CBX160V   -- Display Authorization List IFS Objects - VCP
CBX160X   -- Display Authorization List IFS Objects

CBX160M   -- Display Authorization List IFS Objects - Build Command
```

To create these objects, compile and run CBX160M. Compilation instructions are in the source headers, as usual.

You can find part one of this article here:

<http://www2.systeminetwork.com/article.cfm?id=52913>

This article demonstrates the following APIs:

List Objects Secured by Authorization List (QSYLATLO) API:

<http://publib.boulder.ibm.com/infocenter/iseries/v5r3/topic/apis/qsylatlo.htm>

Open Display Application (QUIOPNDA) API:

<http://publib.boulder.ibm.com/infocenter/iseries/v5r3/topic/apis/quiopnda.htm>

Open Print Application (QUIOPNPA) API:

<http://publib.boulder.ibm.com/infocenter/iseries/v5r3/topic/apis/quiopnpa.htm>

Close Application (QUICLOA) API:

<http://publib.boulder.ibm.com/infocenter/iseries/v5r3/topic/apis/quicloa.htm>

Display Panel (QUIDSPP) API:

<http://publib.boulder.ibm.com/infocenter/iseriess/v5r3/topic/apis/quidspp.htm>

Print Panel (QUIPRTP) API:

<http://publib.boulder.ibm.com/infocenter/iseriess/v5r3/topic/apis/quiprtp.htm>

Put Dialog Variable (QUIPUTV) API:

<http://publib.boulder.ibm.com/infocenter/iseriess/v5r3/topic/apis/quiputv.htm>

Get Dialog Variable (QUIGETV) API:

<http://publib.boulder.ibm.com/infocenter/iseriess/v5r3/topic/apis/quigetv.htm>

Add List Entry (QUIADDLE) API:

<http://publib.boulder.ibm.com/infocenter/iseriess/v5r3/topic/apis/quiaddle.htm>

Get List Entry (QUIGETLE) API:

<http://publib.boulder.ibm.com/infocenter/iseriess/v5r3/topic/apis/quigetle.htm>

Set List Attributes (QUISETLA) API:

<http://publib.boulder.ibm.com/infocenter/iseriess/v5r3/topic/apis/quisetla.htm>

Delete List (QUIDLTL) API:

<http://publib.boulder.ibm.com/infocenter/iseriess/v5r3/topic/apis/quidltl.htm>

Display Long Text (QUILNGTX) API:

<http://publib.boulder.ibm.com/infocenter/iseriess/v5r3/topic/apis/quilngtx.htm>

Get Attributes (QpolGetAttr) API:

<http://publib.boulder.ibm.com/infocenter/iseriess/v5r3/topic/apis/qgetattr.htm>

Get Path from File ID (QpolGetPathFromFileID) API:

<http://publib.boulder.ibm.com/infocenter/iseriess/v5r3/topic/apis/getpthff.htm>

Retrieve Job Information (QUSRJOBI) API:

<http://publib.boulder.ibm.com/infocenter/iseriess/v5r3/topic/apis/qusrjobi.htm>

Create User Space (QUSCRTUS) API:

<http://publib.boulder.ibm.com/infocenter/iseriess/v5r3/topic/apis/quscrtus.htm>

Delete User Space (QUSDLTUS) API:

<http://publib.boulder.ibm.com/infocenter/iseriess/v5r3/topic/apis/qusdltus.htm>

Retrieve Pointer to User Space (QUSPTRUS) API:

<http://publib.boulder.ibm.com/infocenter/iseriess/v5r3/topic/apis/qusptrus.htm>

Send Program Message (QMHSNDPM) API:

<http://publib.boulder.ibm.com/infocenter/iseriess/v5r3/topic/apis/QMHSNDPM.htm>

Retrieve Object Description (QUSROBJD) API:

<http://publib.boulder.ibm.com/infocenter/iseriess/v5r3/topic/apis/qustrobjd.htm>

You can retrieve the source code for this API example from

http://www.pentontech.com/IBMContent/Documents/article/53025_108_AuthList2.zip.

Source URL: <http://iprodeveloper.com/rpg-programming/apis-example-authorization-list-apis-and-secured-objects-part-2>