


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

## APIs by Example: UIM & Work Management APIs, Part 2

[System iNetwork Programming Tips Newsletter](#)

[Carsten Flensburg](#)

Carsten Flensburg

Thu, 06/22/2006 (All day)

This week, I continue my exploration of the User Interface Manager (UIM) APIs and UIM List Panel Groups. To that end, I've written the Work with Subsystem Job Queues (WRKSBSJOBQ) command. Based on the UIM panel group source for the WRKSBSJOBQ list panel group, I guide you through the different sections of UIM tags, together making up the list panel. Last time, I listed and briefly explained the UIM tags involved in creating a list panel group. Now it's time to see how to actually define the UIM tags and their different attributes.

Before we get into the UIM tag details, let me show you what will be the outcome of today's efforts. Here's the WRKSBSJOBQ command prompt:

```

Work with Subsystem Job Queues (WRKSBSJOBQ)
Type choices, press Enter.

Subsystem . . . . . Name
Library . . . . . *LIBL Name, *LIBL, *CURLIB
  
```

Running the WRKSBSJOBQ command or selecting option 12 from the list panel of the Work with Subsystem Activity (WRKSBSACT) presented last time takes you to the following list panel:

```

Work with Subsystem Job Queues WYNDHAMW
16-06-06 17:11:37
Subsystem . . . : QSYSWRK Subsystem status : *ACTIVE

Library . . . : QSYS Max active jobs : *NOMAX

Type options, press Enter.

2=Change 3=Hold 4=Remove 5=Work with job queue jobs
6=Release

Seq Job Max Current Jobs
Opt Nbr Queue Library Status Jobs Active On queue

10 QSYSNOMAX QSYS RLS *NOMAX 45 0

20 QPDAUTOPAR QSYS RLS 5 0 0

30 QESAUTON QSYS RLS 5 0 0
  
```

40	QNMSVQ	QSYS	RLS	5	0	0
70	Q1PSCHQ	QSYS	RLS	1	0	0
80	Q1PSCHQ2	QSYS	RLS	1	1	0
90	Q1PSCHQ3	QSYS	RLS	1	0	0
100	QSJINV	QSYS	RLS	1	0	0
						Bottom
Parameters or command						
===>						
F3=Exit F4=Prompt F5=Refresh F6=Add job queue entry F9=Retrieve						
F10=Work with subsystem jobs F11=Display max F12=Cancel F21=Print						

As you can see, many options and command key functions are available from this panel. You can change, hold, release, and remove any job queue in the list, as well as run yet another new work management command, Work with Job Queue Jobs (WRKJOBQJOB). The WRKJOBQJOB command will be part of the next and final installment of this article series, so until then, that option is a dead end.

Function key 10 runs the WRKSBSJOB command for the current subsystem, and function key 11 toggles between status, max by priority, and text description information for the job queues. A detailed explanation of all available options, function keys, and list panel fields, as well as online cursor-sensitive help, is provided.

Now that I've introduced you to the WRKSBSJOBQ command, let me discuss the UIM tags that make it all happen. I select snippets of the different sections of the UIM list panel group as they appear in the source code that accompanies this article and explain what they do.

```
:PNLGRP
```

The panel group tag defining the beginning of a panel. Likewise, the end of a panel group is signaled by the :EPNLGRP tag.

```
:COPYR.
(C) 2006 Penton Technology Media
```

The COPYR tag presents the specified copyright notice in the message area of the first panel displayed.

```
:IMPORT NAME='*' PNLGRP=CBX157H.
```

IMPORT makes help modules in the specified panel group available to the current panel group. The NAME attribute defines the help module to import. A '\*' special value for this attribute means that any unresolved help modules are to be imported from this panel group.

```

:CLASS      NAME=CLACTION    BASETYPE=ACTION      WIDTH=2.
:ECLASS.
:CLASS      NAME=CLEXITPG    BASETYPE='CHAR 20'.
:ECLASS.
.
:CLASS      NAME=CLALTVIEW   BASETYPE='BIN 15'.
:ECLASS.
.
:CLASS      NAME=CLMAXPTY    BASETYPE='ZONED 2 0'.
:TL.
:TI  VALUE='-1'.*
:ETL.
:ECLASS.

```

The CLASS definition tags in a panel group specify the data types available to the variable (VAR) definitions. So to include a variable of type character and size 20 bytes, a class defining these attributes has to be declared first. As you will see later, this class is then specified as that variable's class name. More variables can share the same class.

Value translation and value validity checks are defined as part of the class definition. In the preceding excerpt, the class CLMAXPTY has a translation list (TL) defined, which performs the conversion between the internal value -1 and the external value '\*'. If any variable based on the CLMAXPTY class has a value of -1, the UIM displays the value '\*' in the list panel. For input fields, the opposite conversion is also performed.

For input fields, you can also include the CHECK tag in the context of the class definition for which the check should apply, as in the following example:

```

:CLASS NAME=CLREPLY  BASETYPE='CHAR 4'  SHIFT=UPPER  CASE=UPPER.
:CHECK VALUES=' '*YES"  "*NO"'  MSGID=USR0001  MSGF=USRMSGF.
:ECLASS.

```

This statement allows only the values \*YES or \*NO for the input fields based on the CLREPLY class. If any other value is entered, the message USR0001 in the USRMSGF message file is issued.

The class definition tag is a comprehensive topic, so I recommend that you consult Appendix A of the Application Display Programming manual (link is provided at the end of this article), which provides a detailed discussion of the many different data types and tag attributes.

```

:VAR      NAME=OPTION    CLASS=CLACTION.
:VAR      NAME=EXITPG    CLASS=CLEXITPG.
.
:VAR      NAME=ALTVIEW   CLASS=CLALTVIEW.
.
:VAR      NAME=MAXPT1    CLASS=CLMAXPTY.
:VAR      NAME=MAXPT2    CLASS=CLMAXPTY.

```

The VAR (Variable Definition) tag declares a dialog variable in the panel group. As I mentioned, the variable's data type is defined by the assigned class name.

```

:VARRCD  NAME=EXPRCD
        VARS='EXITPG'.

```

```

:VARRCD  NAME=HDRRCD
          VARS='DATE TIMZON SBSNAM SBSLIB SBSSTS MAXACT'.
:VARRCD  NAME=DTLRCD
          VARS='OPTION SEQNBR JBQNAM JBQLIB JBQSTS JBQMAX CURACT'
          VARS='JOBONQ JBQDSC MAXPT1 MAXPT2 MAXPT3 MAXPT4 MAXPT5'
          VARS='MAXPT6 MAXPT7 MAXPT8 MAXPT9'.

```

The VARRCD (Variable Record) is used to pass data between the UIM panel group's variable pool and the application. This tag defines the variables included in the specified record and the sequence in which they must appear when passed to the UIM. Data is passed between the application and the UIM using the Get and Put Dialog Variable APIs, QUIGETV and QUIPUTV, respectively.

Here's how the preceding HDRRCD variable record is defined in the RPG IV program that communicates with the UIM:

```

**-- UIM Panel header record:
D HdrRcd          Ds                      Qualified
D  SysDat          7a
D  SysTim          6a
D  TimZon          10a
D  SbsNam          10a
D  SbsLib          10a
D  SbsSts          10a
D  MaxActJob       7s 0

```

The field sequence matches the VARRCD specification, and each field's data type and length matches each variable's CLASS definition.

```

:LISTDEF NAME=DTLLST
          VARS='OPTION SEQNBR JBQNAM JBQLIB JBQSTS JBQMAX CURACT'
          VARS='JOBONQ JBQDSC MAXPT1 MAXPT2 MAXPT3 MAXPT4 MAXPT5'
          VARS='MAXPT6 MAXPT7 MAXPT8 MAXPT9'
          MSGID=CPI0925
          MSGF='QCPFMSG'.

```

The LISTDEF (List Definition) defines the list or data structure that the UIM uses to build the actual panel list. The list definition is based on the variable record defined earlier. The message ID and message file (specified in the preceding code snippet) define the message that the UIM shows in place of the list, if the list is empty when the panel is displayed.

List entries are added to the list panel using the Add List Entry (QUIADDLE) or Add List Multiple Entries (QUIADDLM) APIs. List entries can be changed by the Update List Entry (QUIUPDLE) API, retrieved back into the application by the Get List Entry (QUIGETLE) or Get List Multiple Entries (QUIGETLM) APIs, and removed by the Remove List Entry (QUIRMVLE) API.

```

:COND     NAME=SBSACT      EXPR='SBSSTS *EQ "*"ACTIVE"'.
:COND     NAME=VIEW1       EXPR='ALTVIEW = 0'.
:COND     NAME=VIEW2       EXPR='ALTVIEW = 1'.
:COND     NAME=VIEW3       EXPR='ALTVIEW = 2'.

```

The COND (Condition) tag specifies a certain condition used to control the appearance and behavior of the panel group. In the preceding example, the condition SBSACT is on if the value of the SBSSTS variable equals '\*ACTIVE'. This condition is then used to control whether function key F10 should appear in the Function Key List and be valid if pressed by the user:

```
:KEYI      KEY=F10
           HELP=HLPF10
           PRIORITY=20
           COND=SBSACT
           ACTION='CMD WRKSBSJOB SBS (&SBSNAM) '
           .F10=Work with subsystem jobs
```

Likewise, the VIEW $n$  conditions are used to define which text should appear for the F11 function key, in relation to the toggle function performed by this function key. This condition is also applied to the panel group Truth Table definition, which helps the UIM compiler establish all possible panel formats:

```
:TT        NAME=PNLTT
           CONDS= 'VIEW1    VIEW2    VIEW3' .
:TTROW     VALUES='  1      0      0  ' .
:TTROW     VALUES='  0      1      0  ' .
:TTROW     VALUES='  0      0      1  ' .
:ETT .
```

An individual panel format is generated for each possible combination of the VIEW $n$  conditions.

```
:KEYL      NAME=FKEYS
           HELP=HLPFKEYS .
:KEYI      KEY=ENTER
           HELP=HLPENT
           ACTION=ENTER .
.
.
:KEYI      KEY=F3
           HELP=HLPF3
           ACTION='EXIT SET'
           VARUPD=NO
           .F3=Exit
.
.
:KEYI      KEY=F5
           HELP=HLPF5
           PRIORITY=20
           ACTION='RETURN 5'
           .F5=Refresh
:KEYI      KEY=F6
           HELP='HLPF6'
           ACTION='CMD ?ADDJOBQE ?*SBSD (&SBSLIB/&SBSNAM) '
           PRIORITY=20
           .F6=Add job queue entry
:EKEYL .
```

The KEYL (Key List) tag defines sets of available function keys and the functions they perform. The key list is eventually assigned to a panel definition. Between the KEYL and EKEYL (End of Key List) tags, the KEYI (Key List Items) are specified. A key list item defines a single function key and the action that it should perform. A number of UIM predefined actions, also called dialog commands, are available — for example, CHGVIEW, CANCEL, PROMPT, CMD, and CALL.

Please refer to Appendix A of the Application Display Programming manual for a detailed explanation of the KEYL and KEYI tags.

```
:PANEL      NAME=CBX157P
            KEYL=FKEYS
            ENTER='RETURN 500'
            HELP='WRKSBSJOBQ/'
            TT=PNLTT
            TOPSEP=DATETIME
            DATE=DATE
            TIME=TIMZON
            .Work with Subsystem Job Queues
```

The PANEL tag defines the beginning of a display panel format and many of its related attributes, such as panel name, user exit program, panel dialog variables, and much more. The Display Panel (QUIDSPP) API uses the panel name attribute to identify the panel format to display.

The ENTER attribute defines what should occur if Enter is pressed and no option is specified for a list entry and nothing was entered on the command line. In the preceding snippet, the value 500 is returned in the Function Requested parameter of the Display Panel API. The program running this API is then in charge of deciding what should happen next.

The KEYL attribute defines a previously defined key list to be displayed and used with the panel, the TT attribute defines the Truth Table to be applied by the compiler, and the final attribute defines the panel top separator format and the variables holding the date and time information to be used in that format. Again, please check the manual for all the details.

Between the PANEL and EPANEL (End of Panel) tags are the following tag declarations, which apply to the panel in question:

```
:DATA      DEPTH=3
            LAYOUT=2
            BOTSEP=SPACE.

:DATACOL   WIDTH=18.
:DATACOL   WIDTH='*'.
:DATAGRP   GRPSEP=QINDENT
            COMPACT.

:DATAI     VAR=SBSNAM
            USAGE=OUT
            HELP='WRKSBSJOBQ/SBSNAM'
            .Subsystem

:DATAI     VAR=SBSLIB
            USAGE=OUT
            HELP='WRKSBSJOBQ/SBSLIB'
            .Library
```

```
:EDATAGRP.
.
:EDATA.
```

The DATA presentation area describes the header information in the WRKSBSJOBQ list panel. This is where the specified subsystem name and related information is displayed. Again, different attributes are available to define presentation area layout and appearance.

The manual contains many examples of how to compose different types of data presentation areas. Look up the DATA tag section in Appendix A of the manual for further details.

```
:LIST      DEPTH='*'
           LISTDEF=DTLLST
           MAXHEAD=2
           VIEW=ALTVIEW
           ACTOR=UIM
           SCROLL=YES
           PARMS=CMDPRM.
```

The LIST area tag defines the list area of the panel, such as separator attributes, view toggle dialog variable, parameters dialog variable, as well as the DTLLST list specification previously defined. The ACTOR keyword specifies who gains control when the actions defined by the LISTACT tags (see the following) are performed. In this case, the UIM is in charge. The ELIST tag specifies the end of a list area definition.

```
:TOPINST .Type options, press Enter.
```

The TOPINST tag specifies the text that appears in the top section of the list, right above the list area.

```
:LISTACT  ENTER='CMD CHGJOBQE SBSD(&SBSLIB/&SBSNAM) '
           ENTER='JOBQ(&JBQLIB/&JBQNAM) MAXACT(&JBQMAX) '
           ENTER='SEQNBR(&SEQNBR) &CMDPRM'
           PROMPT='CMD ?CHGJOBQE ?*SBSD(&SBSLIB/&SBSNAM) '
           PROMPT='?*JOBQ(&JBQLIB/&JBQNAM) MAXACT(&JBQMAX) '
           PROMPT='SEQNBR(&SEQNBR) &CMDPRM'
           HELP='WRKSBSJOBQ/OPTCHG'
           OPTION=2
           NOCMD=PROMPT
           USREXIT='CALL EXITPG'
           .2=Change
```

The LISTACT (List Action) tag defines an option available to perform against one or more of the list entries.

For the preceding option 2 definition, the ENTER attribute specifies what happens for the list entry if option 2 is entered and Enter is pressed, whereas the following PROMPT attribute does the exact same if the function key assigned to the PROMPT dialog command is pressed. The latter is typically done by function key F4.

The command specified in the USREXIT attribute is performed when the ENTER or PROMPT action has completed. The program being called is defined by the content of the EXITPG variable, previously loaded to the variable pool. The parameter interface used by the UIM during this process

is documented in the Exit Program section of the UIM API manual (link is provided at the end of this article).

In this API example, the General UIM Exit Program CBX157E handles all types of UIM exit program calls, including this duty, which for the option 2 exit implies updating the changed list entry status.

```
:LISTCOL  VAR=OPTION
          USAGE=INOUT
          MAXWIDTH=3
          HELP= 'WRKSBSJOBQ/OPTION'
          .Opt
:LISTCOL  VAR=SEQNBR
          USAGE=OUT
          MAXWIDTH=5
          HELP= 'WRKSBSJOBQ/SEQNBR'
          . 'Seq' 'Nbr'
```

The LISTCOL (List Column) tag specifies each list column and a number of attributes related to it. The actual combination and sequence of the fields appearing in the list is, however, defined by the LISTVIEW tag:

```
:LISTVIEW COLS='OPTION SEQNBR JBQNAM JBQLIB JBQSTS JBQMAX CURACT
              JOBONQ' .
:LISTVIEW COLS='OPTION SEQNBR JBQNAM JBQLIB MAXPTY' .
:LISTVIEW COLS='OPTION SEQNBR JBQNAM JBQLIB JBQDSC' .
```

The LISTVIEW (List View) tag defines the different list views, meaning the actual constellation of list columns and list groups, and the order in which they appear.

For the WRKSBSJOBQ list panel, three alternate list views are available. The user can use function key F11 to toggle between each view. Function key F11 has been assigned the special CHGVIEW function key dialog command. Every time this key is pressed, the UIM updates the ALTVIEW variable, which was defined by the list's VIEW attribute, accordingly. This in turn sets the VIEWn condition previously specified, thereby controlling the text that appears for function key F11.

```
:CMDLINE  SIZE=SHORT
          .Parameters or command
```

The CMDLINE (Command Line) tag defines the command-line area where commands and list action parameters are entered.

Besides the main list panel just described, the CBX157P panel group also includes a confirmation list panel that the UIM uses to perform the confirmation dialog for list option 4, Remove Job Queue.

Similar to the display list, a couple of print panel definitions are also included. The PRTHEAD (Print Header) tag describes the print list header definition, and the PRTPNL (Print Panel) tag defines one or more area definitions for the print panel. This works pretty much along the same lines as the previous display panel, so because of space constraints, I do not go into more detail here. Please refer to the manual and the panel group source for these.



In the final installment of this article series, I will look closer at the UIM API dialog with the panel group and also deliver the missing WRKJOBQJOB command, to complete the suite of my new work management commands.

This APIs by Example includes the following sources:

```
CBX157  -- Work with Subsystem Job Queues - CCP
CBX157E -- Work with Subsystem Job Queues - UIM Exit Program
CBX157H -- Work with Subsystem Job Queues - Help
CBX157P -- Work with Subsystem Job Queues - Panel Group
CBX157V -- Work with Subsystem Job Queues - VCP
CBX157X -- Work with Subsystem Job Queues

CBX157M -- Work with Subsystem Job Queues - Build command
```

To create all these objects, compile and run CBX157M. Compilation instructions are in the source headers, as usual.

Previously published UIM related APIs by Example articles:

<http://www2.systeminetwork.com/article.cfm?id=52457>

<http://www2.systeminetwork.com/article.cfm?id=52574>

<http://www2.systeminetwork.com/article.cfm?id=52665>

Application Display Programming — printable PDF:

<http://publib.boulder.ibm.com/infocenter/iseres/v5r4/topic/books/sc415715.pdf>

UIM APIs — printable PDF:

[http://publib.boulder.ibm.com/infocenter/iseres/v5r4/topic/apis\\_web/uim.pdf](http://publib.boulder.ibm.com/infocenter/iseres/v5r4/topic/apis_web/uim.pdf)

This article demonstrates the following APIs:

Open List of Job Queues (QSPOLJBQ) API:

<http://publib.boulder.ibm.com/infocenter/iseres/v5r3/topic/apis/qspoljbq.htm>

Retrieve Job Queue Information (QSPRJQBQ) API:

<http://publib.boulder.ibm.com/infocenter/iseres/v5r3/topic/apis/qsprjobq.htm>

Retrieve Subsystem Information (QWDRSBSD) API:

<http://publib.boulder.ibm.com/infocenter/iseres/v5r3/topic/apis/qwdrsbds.htm>

Open List of Objects (QGYOLOBJ) API:

<http://publib.boulder.ibm.com/infocenter/iseres/v5r3/topic/apis/qgyolobj.htm>

Get List Entries (QGYGTLE) API:

<http://publib.boulder.ibm.com/infocenter/iseres/v5r3/topic/apis/qgygtle.htm>

Close List (QGYCLST) API:

<http://publib.boulder.ibm.com/infocenter/iseres/v5r3/topic/apis/qgyclst.htm>

Open Display Application (QUIOPNDA) API:

<http://publib.boulder.ibm.com/infocenter/iseres/v5r4/topic/apis/quiopnda.htm>

Close Application (QUICLOA) API:

<http://publib.boulder.ibm.com/infocenter/iseres/v5r4/topic/apis/quicloa.htm>

Display Panel (QUIDSPP) API:

<http://publib.boulder.ibm.com/infocenter/iseriess/v5r4/topic/apis/quidspp.htm>

Display Long Text (QUILNGTX) API:

<http://publib.boulder.ibm.com/infocenter/iseriess/v5r4/topic/apis/quilngtx.htm>

Put Dialog Variable (QUIPUTV) API:

<http://publib.boulder.ibm.com/infocenter/iseriess/v5r4/topic/apis/quiputv.htm>

Get Dialog Variable (QUIGETV) API:

<http://publib.boulder.ibm.com/infocenter/iseriess/v5r4/topic/apis/quigetv.htm>

Add List Entry (QUIADDLE) API:

<http://publib.boulder.ibm.com/infocenter/iseriess/v5r4/topic/apis/quiaddle.htm>

Get List Entry (QUIGETLE) API:

<http://publib.boulder.ibm.com/infocenter/iseriess/v5r4/topic/apis/quigetle.htm>

Update List Entry (QUIUPDLE) API:

<http://publib.boulder.ibm.com/infocenter/iseriess/v5r4/topic/apis/quiupdle.htm>

Remove List Entry (QUIRMVLE) API:

<http://publib.boulder.ibm.com/infocenter/iseriess/v5r4/topic/apis/quirmvle.htm>

Retrieve List Attributes (QUIRTVLA) API:

<http://publib.boulder.ibm.com/infocenter/iseriess/v5r4/topic/apis/quirtvla.htm>

Set List Attributes (QUISETLA) API:

<http://publib.boulder.ibm.com/infocenter/iseriess/v5r4/topic/apis/quisetla.htm>

Delete List (QUIDLTL) API:

<http://publib.boulder.ibm.com/infocenter/iseriess/v5r4/topic/apis/quidltl.htm>

Print Panel (QUIPRTP) API:

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

Add Print Application (QUIADDPA) API:

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

Remove Print Application (QUIRMVPA) API:

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

Process Commands (QCAPCMD) API:

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

Retrieve Message (QMHRTVM) API:

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

You can retrieve the source code for this API example from:

[http://www.pentontech.com/IBMContent/Documents/article/52739\\_76\\_UimWrkMgt2.zip](http://www.pentontech.com/IBMContent/Documents/article/52739_76_UimWrkMgt2.zip)

**Source URL:** <http://iprodeveloper.com/rpg-programming/apis-example-uim-work-management-apis-part-2>