



APIs by Example: Automatically Starting Jobs and Running Programs on Your System

[System iNetwork Programming Tips Newsletter](#)

[Carsten Flensburg](#)

Carsten Flensburg

Thu, 03/29/2007 (All day)

Writing certain types of programs and utilities (e.g., server programs, monitor facilities, any other never-ending processes) presents you with the challenge of starting the programs at some point. In some cases the system job scheduler takes care of this requirement; the Add Job Schedule Entry (ADDJOBSCDE) and Work with Job Schedule Entries (WRKJOBSCDE) commands let you easily set up jobs to run regularly.

Sometimes, however, you need to ensure that a specific program is running and available at all times whenever your system or specific subsystem is active. In such cases, adding an autostart job entry to the appropriate subsystem often comes in very handy. Contrary to the job scheduler solution, however, only the most basic commands exist to set up and handle autostart job entries.

APIs and some programming effort can make a difference. Two new CL commands are targeted at setting up and testing autostart job entries: Work with Autostart Job Entries (WRKAJE) and Submit Job Description Job (SBMJOBJOB).

The WRKAJE command is based on the Retrieve Subsystem Information (QWDRSBSD) and the List Subsystem Entries (QWDLSE) APIs. The QWDRSBSD API returns information about the operational attributes of a subsystem, including the current subsystem status and the library where the subsystem is located.

The QWDLSE API returns information of the various types of subsystem entries for a given subsystem, such as:

- Routing entries
- Communications entries
- Remote location entries
- Prestart job entries
- Workstation entries
- Autostart job entries

The QWDLSE API returns the selected information to a user space. In this context, the autostart job entries are of interest. Once I have the list of entries for the specified subsystem, I can use the autostart job name and job description of each entry to perform various functions and retrieve even more information based on this data, as you will see later.

For a thorough explanation of how the QWDLSE API call is performed and the returned data is retrieved, please check out a previous article in this series, which used the same APIs to build the Work with Routing Entries (WRKRTGE) command. I've provided a link to that article below.

Here's the WRKAJE command prompt:

Work with Autostart Job Entrs (WRKAJE)

Type choices, press Enter.

Subsystem

Name

Library

*LIBL

Name, *LIBL,

*CURLIB

You simply specify the qualified name of the subsystem whose autostart job entries you want to display or work with, and press Enter. I'll show you an example of the resulting panel displaying the autostart job entries in a moment, but for now let's focus on adding the autostart job entries to a subsystem description.

You add an autostart job entry to a subsystem using the Add Autostart Job Entry (ADDAJE) command. Apart from the qualified subsystem name for which the autostart job entry is added, you specify the name to be used for the job and the qualified name of a job description. I'll show you an example of how this is done later.

Once added to a subsystem, the autostart job starts whenever the subsystem starts. It is possible to add or remove autostart job entries while the subsystem is active, but the change does not take effect until the next time the subsystem is started.

The job description specified for the autostart job entry provides the information necessary to start and run the autostart job. The following job description attributes are especially important:

RTGDTA (Routing data)	— determines the subsystem routing entry used to activate the autostart job
RQSDTA (Request data)	— contains the command or program call to execute; program calls may include parameters
USER (User profile)	— defines the user profile used to run the command or program specified above; the default job description user parameter *RQD is not allowed, so you must specify a specific user profile

Other job attributes are also picked up from the job description (e.g., initial library list), but the ones listed above are the crucial parameters defining the autostart job initiation process. It is important to note that autostart jobs are run as batch jobs, so only commands allowed for batch jobs should be specified in the request data parameter. Furthermore, the autostart job always runs in the subsystem for which it was registered, regardless of the job description's *Job queue* (JOBQ) attribute, as autostart jobs are not queued on a job queue.

Also note that if more than one autostart job is specified for a subsystem, they all start simultaneously. Because all autostart jobs are started when the subsystem starts, the value specified for the maximum number of jobs in the subsystem does not prevent the autostart jobs from starting. However, if the maximum number of jobs is exceeded, no other jobs can start in that subsystem until enough autostart jobs have completed to get the number of running jobs below the subsystem's maximum activity level.

See the *OS/400 Work Management* manual, Chapter 9: Autostart Jobs, for more details. A link is provided at the end of this article.

Here are the steps required to setup an autostart job:

1. Create a job description defining the program call or command to be automatically run:

```
CRTJOB JOB(QUSRSYS/WEB001JD)
      USER(WEBAPPUSR)
      RQSDTA('CALL WEBLIB/WEBAPP01 PARM(''T'')')
      RTGDTA('QCMDI')
```

2. Add an autostart entry to a subsystem description:

```
ADDAJE SBSDB(QGPL/QBATCH)
      JOB(WEBSEVER)
      JOB(QUSRSYS/WEB001JD)
```

3. Start the subsystem:

```
STRSBS SBSDB(QGPL/QBATCH)
```

If anything goes wrong when the subsystem is started and the autostart job fails, look in the job log of the subsystem monitor job for error messages. The subsystem monitor job is named after the subsystem, so to investigate an autostart job problem for subsystem QSYSWRK you would run the command

```
WRKJOB JOB(QSYSWRK)
```

The i5/OS itself also uses autostart jobs as part of the system startup process. You normally specify either subsystem QBASE or subsystem QCTL as the controlling subsystem. You define the controlling subsystem in system value QCTLSBSD. If you look at either subsystem description, you'll notice that an autostart job entry named QSTRUPJD is registered, pointing to a job description by the same name. Looking at job description QSTRUPJD reveals the following request data string:

```
QSYS/CALL QSYS/QWDAJPGM.
```

So, every time the controlling subsystem is started, the program QWDAJPGM in library QSYS is called. The QWDAJPGM program then retrieves the system value QSTRUPPGM (Startup program), and transfers control to the program specified there. It is then the responsibility of the startup program to start all subsystems — and optionally, printer writers — required for the system to operate properly.

By default, the CL-program QSTRUP in library QSYS is defined as startup program, but you can replace or adapt it as required. If you choose to replace or adapt the default startup program, it is

recommended that you place the new version in a different library than QSYS and change the QSTRUPPGM system value accordingly.

You can see other examples of operating system uses of autostart jobs by running the new Work with Autostart Job Entries (WRKAJE) command against subsystem QSYSWRK:

WRKAJE SBS(QSYSWRK)

Running this command results in the display of a panel similar to the following:

```

                                Work with Autostart Job Entries

WYNDHAMW                                                                24-03-07

19:02:34
  Subsystem . . . : QSYSWRK                      Subsystem status :
*ACTIVE
  Library . . . : QSYS

Type options, press Enter.

  2=Change   3=Copy           4=Remove   5=Work with job description
  8=Work with active jobs   9=Submit job

                                Job

Opt  Job          Description  Library   Text
-----
      QSYSWRKJOB  QSYSWRK     QSYS      SYSTEM SUBSYSTEM JOB
DESCRIPTION
      QFSIOPJOB   QFSIOPWK     QSYS      FILE SEVER I/O PROCESSOR
JOB DESCRIP
      QDB2MULTI   QQQTEMPS     QSYS      DB2/400 JOB DESCRIPTION
USED BY QSYS
      QXOTJOB     QXOTJOB      QSYS      Object Translation Service
job descr
      QPASVRP     QPASVR       QSYS      Target Pass-through Server
Job Descr
      QNEOSOEM    QNEOJOB      QSYS      Alternate Communications
Path Job De
      QZBSEVTM    QZBSEJBD     QSYS      ZBS Event Monitor Job
Description

More...
Parameters or command

===>

F3=Exit   F4=Prompt   F5=Refresh   F6=Add autostart job entry
F9=Retrieve
F10=Work with subsystem jobs   F12=Cancel   F21=Print list
F24=More keys

```

The above panel lets you run all native Autostart Job Entry commands, namely the Add Autostart Job Entry (ADDAJE), Change Autostart Job Entry (CHGAJE), and Remove Autostart Job Entry (RMVAJE) commands.

The panel also offers entry points to Work with Job Description (WRKJOBDD) for the autostart job entry's job description, Work with Active Jobs (WRKACTJOB) for all jobs having the autostart job entry's job name, and the new Submit Job Description Job (SBMJOBDDJOB) command, which allows you to submit the autostart job entry without having to stop and start the subsystem. Please note that because the SBSJOBDDJOB command actually submits the job to a job queue, the job description should specify a job queue pointing to the correct subsystem.

Command functions keys also let you run the Work with Subsystem Jobs (WRKSBSJOB) and Work with Subsystem Description (WRKSBSD) commands for the selected subsystem. For more information, refer to the online help text provided for both CL commands and the work with panel.

This APIs by Example includes the following sources:

```
CBX1711  -- RPGLE  -- Work with Autostart Job Entries - CCP

CBX1711E -- RPGLE  -- Work with Autostart Job Entries - UIM Exit
Program
CBX1711H -- PNLGRP -- Work with Autostart Job Entries - Help

CBX1711P -- PNLGRP -- Work with Autostart Job Entries - Panel Group

CBX1711V -- RPGLE  -- Work with Autostart Job Entries - VCP

CBX1711X -- CMD     -- Work with Autostart Job Entries

CBX1712  -- CLP     -- Submit Job Description Job - CPP

CBX1712H -- PNLGRP -- Submit Job Description Job - Help
CBX1712V -- RPGLE  -- Submit Job Description Job - VCP
CBX1712X -- CMD     -- Submit Job Description Job

CBX171M  -- CLP     -- Work with Autostart Job Entries - Build commands
```

To create the above objects, compile and run CBX171M. You can find the compilation instructions in the source headers, as usual.

IBM documentation:

OS/400 Work Management — Chapter 9. Autostart Jobs:

<http://publib.boulder.ibm.com/infocenter/iserics/v5r3/topic/books/c4153063.pdf>

Autostart jobs:

<http://publib.boulder.ibm.com/infocenter/iserics/v5r4/topic/rzaks/rzaksaboutautostartjob.htm>

Subsystems shipped with the system:

<http://publib.boulder.ibm.com/infocenter/iserics/v5r4/topic/rzaks/rzakssbsshipped.htm>

Change the IPL startup program:

<http://publib.boulder.ibm.com/infocenter/iserics/v5r4/topic/rzal2/rzal2chgiplstartup.htm>

Previously published Subsystem Entry CL Commands:

Work with Subsystem Job Queues:

<http://www.systeminetwork.com/article.cfm?id=52739>

Work with Routing Entries:

<http://www.systeminetwork.com/article.cfm?id=53255>

This article demonstrates the following Work Management APIs:

Retrieve Subsystem Information (QWDRSBSD) API:

<http://publib.boulder.ibm.com/infocenter/iserics/v5r3/topic/apis/qwdrsbsd.htm>

List Subsystem Entries (QWDLSE) API:

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

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

http://www.pentontech.com/IBMContent/Documents/article/54373_188_AutoStart.zip

Source URL: <http://iprodeveloper.com/rpg-programming/apis-example-automatically-starting-jobs-and-running-programs-your-system>