

# Everything about subsystems LAB 1

## Table of Contents

<a href="#">Table of Contents.....</a>	<a href="#">1</a>
<a href="#">Everything about subsystems Lab 1.....</a>	<a href="#">2</a>
<a href="#">    Create a subsystem to handle batch jobs.....</a>	<a href="#">2</a>
<a href="#">Task 1 – Create the subsystem description.....</a>	<a href="#">2</a>
<a href="#">Task 2 – Create a class to hold your run attributes.....</a>	<a href="#">3</a>
<a href="#">Task 3 – Create a job queue for your batch job to use.....</a>	<a href="#">3</a>
<a href="#">Task 4 – Attach the job queue to your subsystem.....</a>	<a href="#">3</a>
<a href="#">Task 5 – Start your subsystem and submit a batch job.....</a>	<a href="#">3</a>
<a href="#">Task 6 - Add a routing entry to your subsystem.....</a>	<a href="#">3</a>
<a href="#">Task 7 – End your subsystem.....</a>	<a href="#">3</a>
<a href="#">Auto start jobs.....</a>	<a href="#">3</a>
<a href="#">Task 8 – Create a job description to define your auto start job.....</a>	<a href="#">3</a>
<a href="#">Task 9 – Add the auto start job entry to your subsystem.....</a>	<a href="#">4</a>
<a href="#">Task 10 – Start your subsystem.....</a>	<a href="#">4</a>
<a href="#">Task 11 – End your subsystem.....</a>	<a href="#">4</a>
<a href="#">Attaching workstations to your subsystem.....</a>	<a href="#">4</a>
<a href="#">Task 12 – Create a named Workstation.....</a>	<a href="#">4</a>
<a href="#">Task 13 – Add a Workstation entry to your subsystem.....</a>	<a href="#">6</a>
<a href="#">Task 14 – Start your subsystem.....</a>	<a href="#">6</a>
<a href="#">Task 15 – End your subsystem.....</a>	<a href="#">6</a>
<a href="#">Changing your sign on Display.....</a>	<a href="#">6</a>
<a href="#">Task 16 – Change the subsystem to use the WMCPA sign-on screen.....</a>	<a href="#">7</a>
<a href="#">Task 17 – Start your subsystem.....</a>	<a href="#">7</a>
<a href="#">Task 18 – End your subsystem.....</a>	<a href="#">7</a>
<a href="#">Task 19 – Create your own sign-in display.....</a>	<a href="#">7</a>

Task 20 – Change the subsystem to use your custom sign-on screen.....	7
Task 21 – Start your subsystem.....	7
Task 22 – End your subsystem.....	7
Solutions.....	9
Create a subsystem to handle batch jobs.....	9
Task 1 – Create the subsystem description.....	9
Task 2 – Create a class to hold your run attributes.....	9
Task 3 – Create a job queue for your batch job to use.....	9
Task 4 – Attach the job queue to your subsystem.....	9
Task 5 – Start your subsystem and submit a batch job.....	9
Task 6 - Add a routing entry to your subsystem.....	9
Task 7 – End your subsystem.....	9
Auto start jobs.....	9
Task 8 – Create a job description to define your auto start job.....	9
Task 9 – Add the auto start job entry to your subsystem.....	10
Task 10 – Start your subsystem.....	10
Task 11 – End your subsystem.....	10
Task 13 – Add a Workstation entry to your subsystem.....	10
Task 16 – Change the subsystem to use the WMCPA sign-on screen.....	10

## Everything about subsystems Lab 1

### Create a subsystem to handle batch jobs

In this task you will create a ‘normal’ subsystem used to handle batch work

#### Task 1 – Create the subsystem description

You should create the subsystem description in your library (SBSCLASSxx) and the subsystem should have one memory pool assigned (\*BASE), put a relevant description on the object

### **Task 2 – Create a class to hold your run attributes**

The class should be created in your library and named CLASS50, define the job to run at priority 50 and get 2000 MS per time slice, the job should wait 30 seconds for any locks it needs, use a relevant description.

### **Task 3 – Create a job queue for your batch job to use**

The job queue should be created in your library and named BATCHJOBS. It should be able to be controlled by the system operator (and anyone else with job control authority).

### **Task 4 – Attach the job queue to your subsystem**

Attach the job queue to your newly created subsystem at sequence 50, it should allow up to 5 jobs to run

### **Task 5 – Start your subsystem and submit a batch job**

Start your subsystem. Do a DSPSBSJOBS to ensure it started correctly.

What memory pool is your subsystem running in? Why is this different than what you stated when you created the subsystem.

Then submit a job that delays 90 seconds (to give you time to see it), named sbsxxTest1 and repeat the DSPSBSJOBS

What do you see? If the job isn't running, look for its joblog. What do you see?

### **Task 6 - Add a routing entry to your subsystem**

Add a routing entry to your subsystem at sequence 9999, it should catch any routing data the user uses, and call the normal command processor (QSYS/QCMD), use the class you created in Task 2. You will allow as many jobs to use this entry as wish to.

Resubmit your batch job (task 5) what happens? If the job is running, look at it and check the run attributes, what is the run priority, time slice, and wait time?

### **Task 7 – End your subsystem**

Issue the command to end your subsystem

### **Auto start jobs**

In this exercise you will add a job that will automatically run whenever you start your subsystem

### **Task 8 – Create a job description to define your auto start job**

The job description should be named SBSCLASSxx/AUTOSTART and submitted to the job queue you have attached to your subsystem, the job should just execute the command DLYJOB DLY(180). Make sure the job will run under your User ID

### Task 9 – Add the auto start job entry to your subsystem

Create a new auto start job entry to your subsystem, the job name should be MYAUTOJOB, and you should use the job description you created in the prior step

### Task 10 – Start your subsystem

Start your subsystem and use the DSPSBSJOBS to see if your auto start job is running.

### Task 11 – End your subsystem

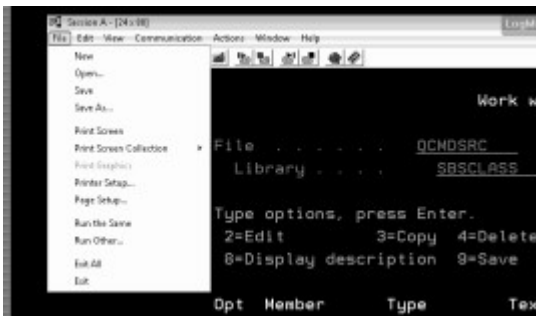
Issue the command to end your subsystem

## Attaching workstations to your subsystem

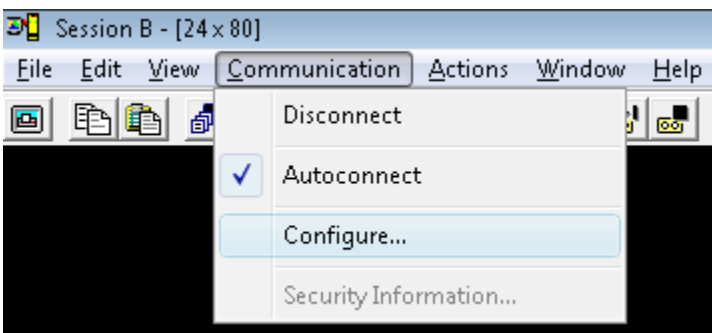
In this task you will learn how to name a workstation in iSeries Nav, then you will attach the workstation to your subsystem

### Task 12 – Create a named Workstation

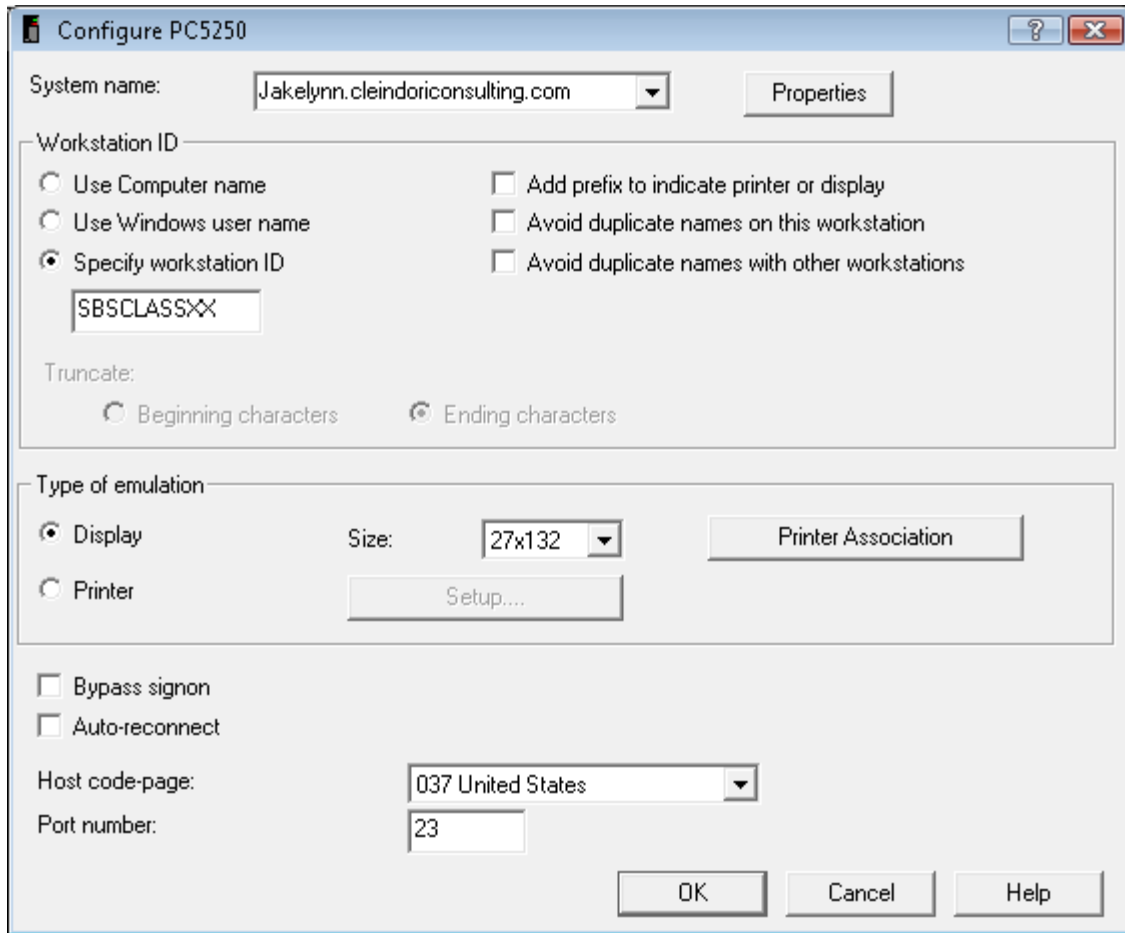
1. From your green screen session select File/Run the Same



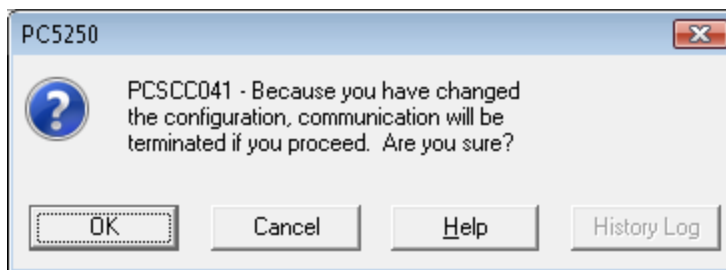
2. Then select Communication/Configure



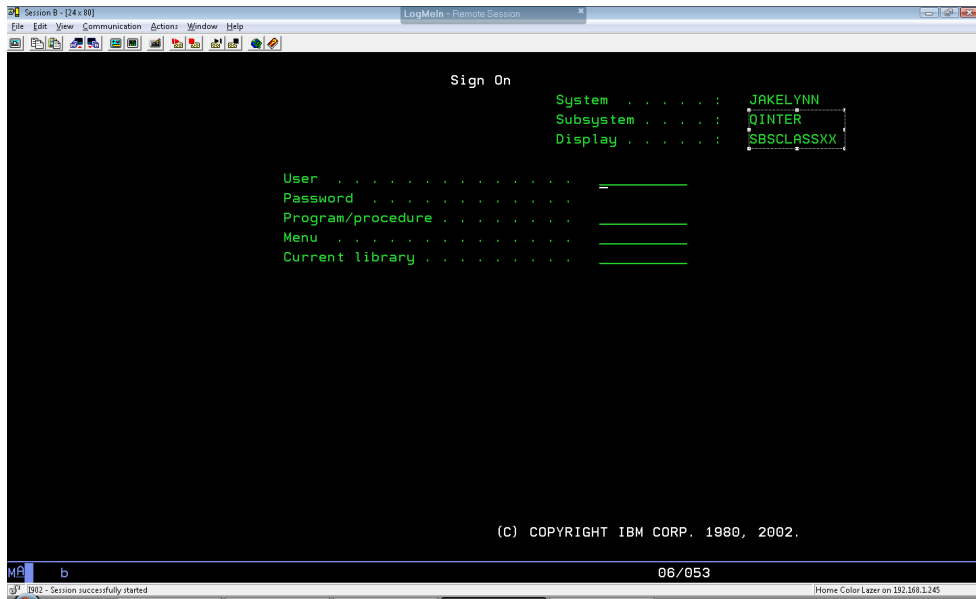
3. Give your display a specific name, name it SBSCLASSxx, then click OK



4. Confirm restart of communications, click OK



5. Confirm name of device, and subsystem you are assigned to



6. Return to your original Green Screen Session

### Task 13 – Add a Workstation entry to your subsystem

Add a workstation entry to your subsystem for the device you just created. Make sure that the allocation parameter indicates allocate the device when the subsystem starts

### Task 14 – Start your subsystem

Start your subsystem and check your named session. What subsystem is it now assigned to? Sign in to the new session. Use DSPSBSJOBS to confirm that you are running in your new subsystem. Sign out of your session, and return to your original one.

### Task 15 – End your subsystem

Issue the command to end your subsystem

## Changing your sign on Display

In this task you will learn how to change your sign-on display. In your library you have the source, and object, for the WMCPA default sign-on display. You will first change the sign-on display for your subsystem to this display, then you will make a copy of the source and modify it as you wish, and then change the display to your custom display.

Changing the default display file requires the \*SECADM special authority so a command has been provided that will change the parameter for you.

### **Task 16 – Change the subsystem to use the WMCPA sign-on screen**

Review the source code in your library for the display file QDSIGNON, note the changes that have been made. Once you are sure that you understand the changes run the command CHGSIGNON DISPLAY(\*WMCPA) this command executes the command CHGSBSD SBSD(SBSCCLASSxx/SBSCCLASSxx) SGNDSPF(SBSCCLASSxx/QDSIGNON)

Display your subsystem description and confirm that it has been changed, you can find the signon display under 'Operational Attributes'

### **Task 17 – Start your subsystem**

Start your subsystem and check your named session. Is the display as you expected? Sign-in to confirm that the sign-in process works. Sign off your session.

### **Task 18 – End your subsystem**

Issue the command to end your subsystem

### **Task 19 – Create your own sign-in display**

Copy the source for QDSIGNON in your library to the new member 'CUSTOM'. Change the source to contain something unique to yourself. Be careful not to change the order of the fields, only add/change attributes and constant fields. You may change the row/column as you wish but remember that you must maintain the integrity of the buffer. After you have completed your changes compile your display file into your library.

### **Task 20 – Change the subsystem to use your custom sign-on screen**

Run the command CHGSIGNON DISPLAY(\*CUSTOM) this command executes the command CHGSBSD SBSD(SBSCCLASSxx/SBSCCLASSxx) SGNDSPF(SBSCCLASSxx/CUSTOM)

Display your subsystem description and confirm that it has been changed, you can find the signon display under 'Operational Attributes'

### **Task 21 – Start your subsystem**

Start your subsystem and check your named session. Is the display as you expected? Sign-in to confirm that the sign-in process works. Sign off your session.

### **Task 22 – End your subsystem**

Issue the command to end your subsystem





## Solutions

### Create a subsystem to handle batch jobs

#### Task 1 – Create the subsystem description

```
CRTSBSD SBSD(SBSCCLASSXX/SBSCCLASSXX) POOLS((1 *BASE)) TEXT('SBSCCLASSxx subsystem')
```

#### Task 2 – Create a class to hold your run attributes

```
CRTCLS CLS(SBSCCLASSXX/CLASS50) RUNPTY(50) TIMESLICE(2000) DFTWAIT(30)
```

#### Task 3 – Create a job queue for your batch job to use

```
CRTJOBQ JOBQ(SBSCCLASSXX/BATCHJOBS) OPRCTL(*YES) AUTCHK(*OWNER)
```

#### Task 4 – Attach the job queue to your subsystem

```
ADDJOBQE SBSD(SBSCCLASSXX/SBSCCLASSXX) JOBQ(SBSCCLASSXX/BATCHJOBS) MAXACT(5) SEQNBR(50)
```

#### Task 5 – Start your subsystem and submit a batch job

```
strsbs sbsclassxx/sbsclassxx
```

The WRKACTJOB (and other commands) use the actual memory pool #, when you create your subsystem the memory pool # is just a sequence #, the actual pool number as shown by wrkshrpool or wrksyssts should match the \*BASE pool ID (always 2)

```
SBMJOB CMD(DLYJOB DLY(90)) JOB(SBSXXTEST1) JOBQ(SBSCCLASSXX/BATCHJOBS)
```

Job isn't running. Job log shows message CPC1117, no routing entry

#### Task 6 - Add a routing entry to your subsystem

```
ADDRTGE SBSD(SBSCCLASSXX/SBSCCLASSXX) SEQNBR(9999) CMPVAL(*ANY) PGM(QSYSL/QCMD)
```

```
CLS(SBSCCLASSXX/CLASS50) MAXACT(*NOMAX)
```

You can see your batch job running. Attributes should match your class

#### Task 7 – End your subsystem

```
ENDSBS SBS(SBSCCLASSXX)
```

## Auto start jobs

#### Task 8 – Create a job description to define your auto start job

```
CRTJOBQ JOBQ(SBSCCLASS/AUTOSTART) JOBQ(SBSCCLASSXX/BATCHJOBS) USER(SBSCCLASSXX)
```

```
TEXT('Autostart job for SBSCLASSxx') RQSDTA('DLYJOB DLY(180)')
```

**Task 9 – Add the auto start job entry to your subsystem**

```
ADDAJE SBSD(SBSCLASSXX/SBSCLASSXX) JOB(MYAUTOJOB) JOBD(SBSCLASSXX/AUTOSTART)
```

**Task 10 – Start your subsystem**

```
STRSBS SBSCLASSxx/SBSCLASSxx
```

**Task 11 – End your subsystem**

```
ENDSBS SBS(SBSCLASSXX)
```

**Task 13 – Add a Workstation entry to your subsystem**

```
ADDWSE SBSD(SBSCLASSXX/SBSCLASSXX) WRKSTN(SBSCLASSXX) AT(*SIGNON)
```

**Task 16 – Change the subsystem to use the WMCPA sign-on screen**

```
DSPSBSD SBSD(SBSCLASSXX/SBSCLASSXX)
```