Installing and Configuring the 6.40 Startup Framework to Use with SAP J2EE Engine 6.20

Last update: 23.09.2004

SAP J2EE Engine 6.20

Contents

Overview

The startup framework is an infrastructure, situated between the operational system and the Java VM. It takes care of loading the VM and restarting processes, which have died/crashed. By providing support for Windows/Linux/Solaris/and so on, it makes possible to unify the cluster management under different operational systems. This startup framework is delivered as a separate additional tool, which can be installed optionally.

Additional Resources

To set up and run the 6.40 startup framework you will need the following files:

- The *SAPCAR* executable and the *SCSCLIENTEP.SAR* for your platform.
- Windows only – the *sapmmc.SAR* and the *R3DLLINST.SAR* files.

Note: For the location of these files, see SAP Note: 701278 (Location of Files for the Startup Framework for 6.20).

To “unsar” all files from a SAR archive, use the following syntax:

```
SAPCAR –xvf <SAR_file_name>
```

To “unsar” certain files from a SAR archive, use the following syntax:

```
SAPCAR –xvf <SAR_file_name> <filename_1> <filename_2> …
```

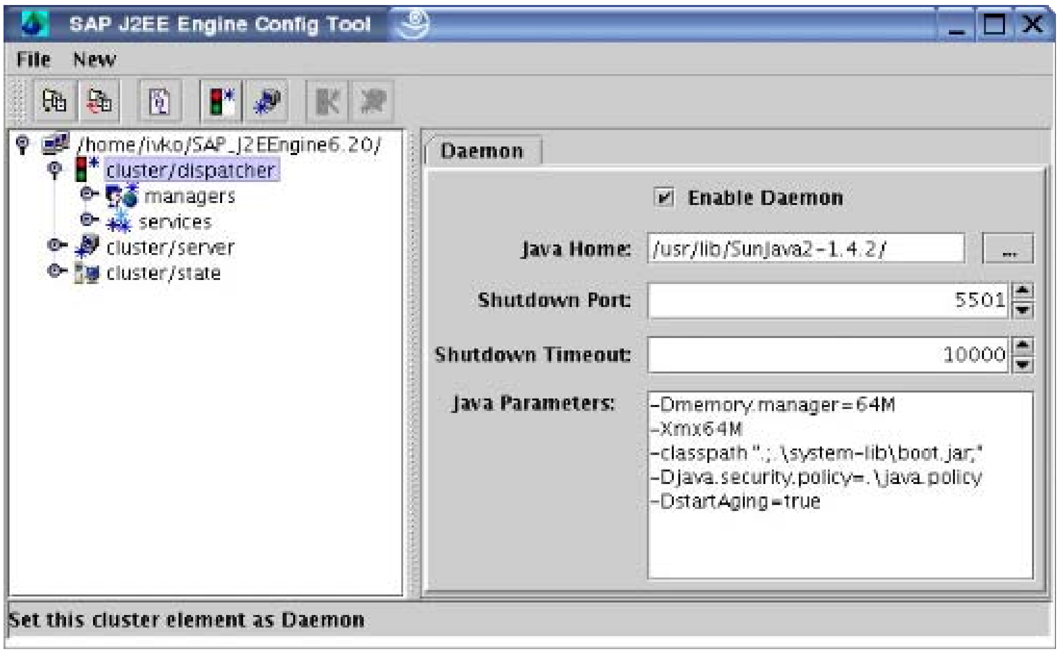
Procedure on UNIX

Preparing the SAP J2EE Engine 6.20

- Log on with the j2eeadm account and start the J2EE Engine Config Tool

```
/usr/sap/<SID>/j2ee/j2ee_<##>/configtool/configtool
```

- In the toolbar, navigate to *cluster/dispatcher*.
- Make sure the “Enable Daemon” flag is activated for all cluster nodes. This will create a required file called *service.ini* in the directory where the Config Tool is located.



Enable Daemon [✓]

Java Home: /usr/lib/SunJava2-1.4.2/

Shutdown Port: 5501

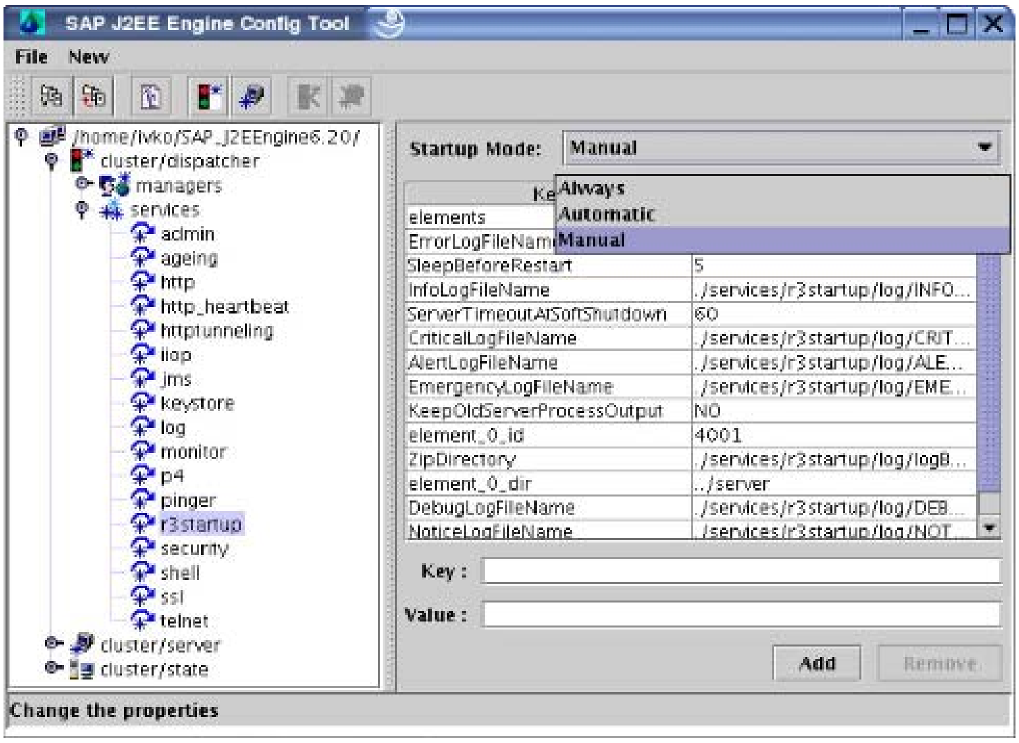
Shutdown Timeout: 10000

Java Parameters:

```
-Dmemory.manager=64M
-Xmx64M
-classpath ".;.\system-lib\boot.jar;"
-Djava.security.policy=.\java.policy
-DstartAging=true
```

Set this cluster element as Daemon

- The R3Startup Service on the dispatcher has to be disabled. Check that the startup mode of the dispatcher’s R3Startup Service is set to “Manual”.



/home/ivko/SAP_J2EEngine6.20/
cluster/dispatcher
managers
services
admin
ageing
http
http_heartbeat
httptunneling
iiop
jms
keystore
log
monitor
p4
pinger
r3startup
security
shell
ssl
telnet
cluster/server
cluster/state

Startup Mode: Manual

Always
Automatic
Manual

Ke	
elements	
ErrorLogFileNam	
SleepBeforeRestart	5
InfoLogFileName	./services/r3startup/log/INFO...
ServerTimeoutAtSoftShutdown	60
CriticalLogFileName	./services/r3startup/log/CRIT...
AlertLogFileName	./services/r3startup/log/ALE...
EmergencyLogFileName	./services/r3startup/log/EME...
KeepOldServerProcessOutput	NO
element_0_id	4001
ZipDirectory	./services/r3startup/log/logB...
element_0_dir	../server
DebugLogFileName	./services/r3startup/log/DEB...
NoticeLogFileName	./services/r3startup/log/NOT...

Key :
Value :

Change the properties

- Open the *service.ini* file with a text editor.
- For each Service described, remove any parameters specifying the VM type (for example, `-server`) from the *Service_n_JavaParameters* line. Note that the parameter names are case-sensitive.
- Other memory tuning parameters (for example, *–XmsnnnM -XX*: *+DisableExplicitGC -XX*:*MaxPermSize=nnM -XX*:*PermSize=nnM -XX*:*MaxNewSize=nnM -XX*:*NewSize=nnM*) must remain in the *Service_n_JavaParameters* line.
- Make sure that the *Service_n_sfSupport* property is set to *yes*. When the property is set to *yes*, the startup framework relies on the J2EE Engine to store information about the nodes (such as, process state, http/https port, telnet port, p4 port, and so on) in the administration shared memory. When the property is set to *no*, the SAP J2EE Engine stores only process state RUNNING.

Preparing the OS Platform

For HP-UX 11.0

- Read the SDK release notes of the HP JDK (*http://www.hp.com/products1/unix/java/java2/sdkrte1_3/infolibrary/sdk_rnotes_1-3-1-11.html#shl_load*) and install patch PHSS_26559.

Installing the 6.40 Startup Framework

- Unpack the *SCSCLIENTEP.SAR* archive (containing the startup framework binaries) on each instance to

```
/usr/sap/<SID>/j2ee/j2ee_<##>/os_libs
```

- Create a sub-directory named "work" underneath this directory.
- Create the profile sub-directory named

```
/usr/sap/<SID>/global/profile
```

- Create a text file in the *profile* sub-directory named

```
/usr/sap/<SID>/global/profile/default.pfl
```

- For each instance, create two text files (so called profiles) in the profile sub-directory. Name the files as follows:

```
/usr/sap/<SID>/global/profile/start_<##>.pfl
/usr/sap/<SID>/global/profile/jcontrol_<##>.pfl
```

The following tables show the contents of these files. Change the paths shown for the *DIR_** variables and the values for the variables *SAPSYSTEM* and *SAPSYSTEMNAME* according to your environment:

default.pfl
<this file has to be empty>
#--- last line -do not delete ---

start_<##>.pfl

```
#-----------------------------------------------------------------------
# Startup Variables
#-----------------------------------------------------------------------

DIR_EXECUTABLE = /usr/sap/<SID>/j2ee/j2ee_<##>/os_libs
DIR_LIBRARY    = /usr/sap/<SID>/j2ee/j2ee_<##>/os_libs
DIR_INSTANCE   = /usr/sap/<SID>/j2ee/j2ee_<##>/os_libs
DIR_PROFILE    = /usr/sap/<SID>/global/profile

# Here, specify the 4 character service SID:
SAPSYSTEMNAME  = <SID>
# Here, specify the instance number of the J2EE Engine 6.20:
SAPSYSTEM      = <##>

# Starts automatically this J2EE instance after restart
# of the physical machine or restart of the SAP Service if the value is
# set to 1. If 0, the automatic start is disabled.
Autostart = 0

#-----------------------------------------------------------------------
# Start Java Dispatcher
#-----------------------------------------------------------------------
```

```
#_JC = jcontrol.exe
#Start_Program_01 = local $(DIR_EXECUTABLE)\$(_JC) pf=$(DIR_PROFILE)\jcontrol_<##>.pfl

#--- last line - do not delete ---
```

Jcontrol_<##>.pfl

```
#-----------------------------------------------------------------------
# Runtime Variables
#-----------------------------------------------------------------------

DIR_EXECUTABLE = /usr/sap/<SID>/j2ee/j2ee_<##>/os_libs
DIR_LIBRARY    = /usr/sap/<SID>/j2ee/j2ee_<##>/os_libs
DIR_INSTANCE   = /usr/sap/<SID>/j2ee/j2ee_<##>/os_libs
DIR_PROFILE    = /usr/sap/<SID>/global/profile

# Here, specify the 4 character service SID:
SAPSYSTEMNAME  = <SID>
# Here, specify the instance number of the J2EE Engine 6.20:
SAPSYSTEM      = <##>

jstartup/instance_properties = /usr/sap/<SID>/j2ee/j2ee_<##>/configtool/service.ini
jstartup/start_mode          = program
jstartup/protocol            = on
jstartup/release             = 6.20
#For SOLARIS, specify as VM type “hotspot”; for HP UX, specify “client”
jstartup/vm/type             = <VM type>

exe/jlaunch = $(DIR_EXECUTABLE)$(DIR_SEP)jlaunchep$(FT_EXE)

rdisp/TRACE = 1

#--- last line - do not delete ---
```

- In the following directory, create for each instance a shell script to start jcontrol

  `/usr/sap/<SID>/j2ee/j2ee_<##>/os_libs`

  Replace the bold parts with your local settings:

start620Engine

```
#!/bin/csh

set SAPSYSTEMNAME= <SID>
set SAPSYSTEM= <##>

switch ( `uname` )
  case "HP-UX":
    setenv SHLIB_PATH /usr/sap/${SAPSYSTEMNAME}/j2ee/j2ee_${SAPSYSTEM}/os_libs
    setenv LD_PRELOAD libjvm.sl
    breaksw

  case "AIX":
    setenv LIBPATH /usr/sap/${SAPSYSTEMNAME}/j2ee/j2ee_${SAPSYSTEM}/os_libs
    breaksw

  case "Linux":
```

```
  case "SunOS":
    setenv LD_LIBRARY_PATH /usr/sap/${SAPSYSTEMNAME}/j2ee/j2ee_${SAPSYSTEM}/os_libs
    breaksw
endsw

cd /usr/sap/${SAPSYSTEMNAME}/j2ee/j2ee_${SAPSYSTEM}/os_libs
nohup ./jcontrol pf=/usr/sap/${SAPSYSTEMNAME}/global/profile/jcontrol_${SAPSYSTEM}.pfl &
```

stop620Engine

```
#!/bin/csh

set SAPSYSTEMNAME= <SID>
set SAPSYSTEM= <##>

switch ( `uname` )
  case "HP-UX":
    setenv SHLIB_PATH /usr/sap/${SAPSYSTEMNAME}/j2ee/j2ee_${SAPSYSTEM}/os_libs
    setenv LD_PRELOAD libjvm.sl
    breaksw

  case "AIX":
    setenv LIBPATH /usr/sap/${SAPSYSTEMNAME}/j2ee/j2ee_${SAPSYSTEM}/os_libs
    breaksw

  case "Linux":
  case "SunOS":
    setenv LD_LIBRARY_PATH /usr/sap/${SAPSYSTEMNAME}/j2ee/j2ee_${SAPSYSTEM}/os_libs
    breaksw
endsw

cd /usr/sap/${SAPSYSTEMNAME}/j2ee/j2ee_${SAPSYSTEM}/os_libs

./jcmon pf=/usr/sap/${SAPSYSTEMNAME}/global/profile/jcontrol_${SAPSYSTEM}.pfl > jcmon.out << EOT
20
2
y
0
0
EOT
```

- Grant ownership and execute permissions for the scripts to the j2eeadm account.
- Use these shell scripts to start and stop the SAP J2EE Engine.

Procedure on Windows 2000/XP

General Preparation

The following section describes how to set up a SAP startup Windows service for a SAP J2EE Engine 6.20 using the 6.40 Startup Framework. The service needs a **3-character system name** (letters and digits; must start with a letter), also called SID, for the SAP J2EE Engine 6.20 to be started. Since the standalone SAP J2EE Engine 6.20 comes with J2EE system names of 4 to 20 characters, you will need to decide about another 3 character name for your SAP J2EE Engine 6.20 that you pass on to the service.

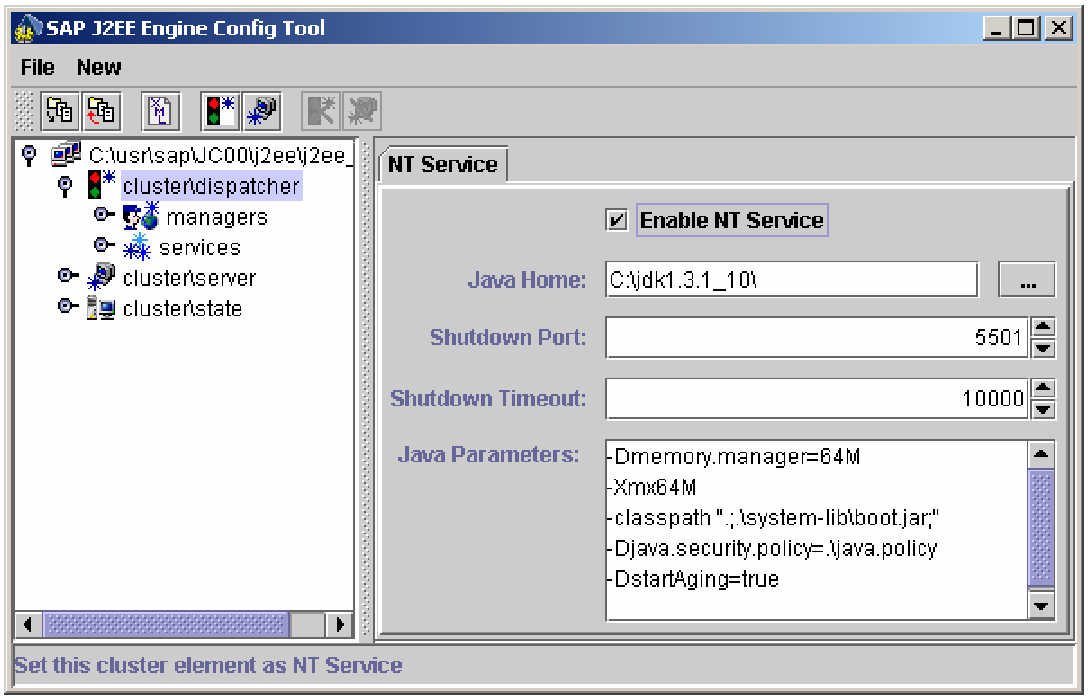
To not get confused with the system names, the 3-character system name required for the service will be called service system name, or service SID, whilst the 4 – 20 character system name for the SAP J2EE Engine 6.20 will be called J2EE system name, or J2EE SID.

Preparing the SAP J2EE Engine 6.20

- Start the SAP J2EE Engine Config Tool

  ```
  <drive>:\usr\sap\<J2EE SID>\j2ee\j2ee_<##>\configtool\configt
  ool.bat
  ```

  and make sure the “Enable NT Service” flag is activated for the cluster nodes. This will create a required file called *service.ini* in the directory where the Config Tool is located.



[✓] Enable NT Service

Java Home: C:\jdk1.3.1_10\

Shutdown Port: 5501

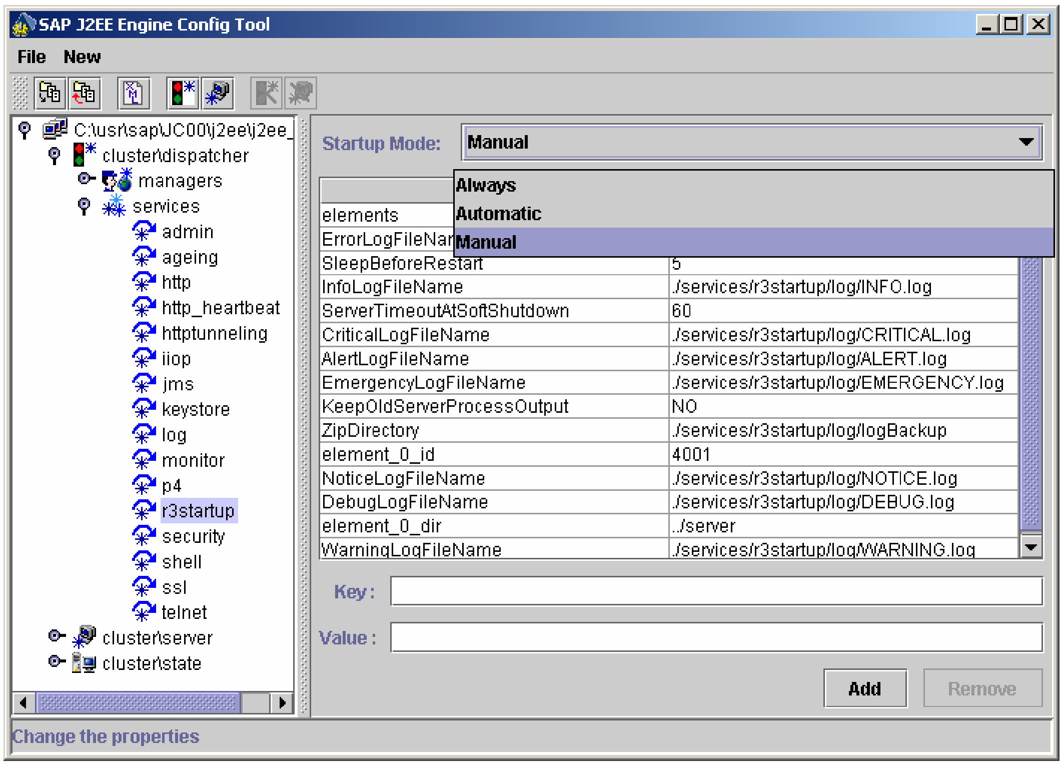
Shutdown Timeout: 10000

Java Parameters:

```
-Dmemory.manager=64M
-Xmx64M
-classpath ".;.\system-lib\boot.jar;"
-Djava.security.policy=.\java.policy
-DstartAging=true
```

Set this cluster element as NT Service

- The R3Startup Service on the dispatcher node has to be disabled. Make sure that the startup mode of the dispatcher’s R3Startup Service is set to “manual”.



Startup Mode: Manual

elements	
ErrorLogFileNar	
SleepBeforeRestart	5
InfoLogFileName	./services/r3startup/log/INFO.log
ServerTimeoutAtSoftShutdown	60
CriticalLogFileName	./services/r3startup/log/CRITICAL.log
AlertLogFileName	./services/r3startup/log/ALERT.log
EmergencyLogFileName	./services/r3startup/log/EMERGENCY.log
KeepOldServerProcessOutput	NO
ZipDirectory	./services/r3startup/log/logBackup
element_0_id	4001
NoticeLogFileName	./services/r3startup/log/NOTICE.log
DebugLogFileName	./services/r3startup/log/DEBUG.log
element_0_dir	../server
WarningLogFileName	./services/r3startup/log/WARNING.log

- Open the *service.ini* file with a text editor.
- For each described service, remove any parameters specifying the VM type (for example, *–server*), from the *Service_n_JavaParameters* line. Note that the parameter names are case-sensitive.
- Make sure that the *Service_n_sfSupport* property is set to *yes*. When the property is set to *yes*, the startup framework relies on the J2EE Engine to store information about the nodes (such as, process state, http/https port, telnet port, p4 port, and so on) in the administration shared memory. When the property is set to *no*, the SAP J2EE Engine stores only process state RUNNING.

Installing the 6.40 Startup Framework and the SAP Service

- Unpack the *SCSCLIENTEP.SAR* archive (containing the startup framework binaries) on each instance to

```
<drive>:\usr\sap\<J2EE SID>\j2ee\j2ee_<##>\os_libs
```

- Upack the *sapmmc.SAR* archive in the *%windir%\system32* directory

```
sapcar.exe –xvf sapmmc.SAR
```

Note: The *sapmmc.SAR* archive is located in the *MMC*.SAR*, which you have downloaded from the Service Marketplace. See SAP Note 701278. That is, you will first have to unpack the *MMC*.SAR* archive and then the *sapmmc.SAR* archive as described above.

- Upack the *R3DLLINST.SAR* to a temporary folder and run *R3DLLINS.EXE* to install the Microsoft runtime DLLs.
- Move *sapmmc.chm* from *%windir%\system32* to *%windir%\help*
- Move *sapstartsrv.exe* from *%windir%\system32* to

```
<drive>:\usr\sap\<J2EE SID>\j2ee\j2ee_<##>\os_libs
```

- Create a sub-directory named "work" underneath this directory.
- Create the profile sub-directory named

```
<drive>:\usr\sap\<J2EE SID>\global\profile
```

- Create a text file in the profile sub-directory named

```
<drive>:\usr\sap\<J2EE SID>\global\profile\default.pfl
```

- For each instance, create two text files (so called profiles) in the profile subdirectory, named

```
<drive>:\usr\sap\<J2EE SID>\global\profile\start_<##>.pfl
<drive>:\usr\sap\<J2EE SID>\global\profile\jcontrol_<##>.
pfl
```

The following tables show the contents of these files. Adapt the paths shown for the *DIR_** variables and the value for the variables *SAPSYSTEM* and *SAPSYSTEMNAME* according to your environment.

default.pfl

```
<this file has to be empty>
#--- last line - do not delete ---
```

start_<##>.pfl

```
#--------------------------------------------------------------------
# Startup Variables
#--------------------------------------------------------------------

DIR_EXECUTABLE = D:\usr\sap\<J2EE SID>\j2ee\j2ee_<##>\os_libs
DIR_LIBRARY    = D:\usr\sap\<J2EE SID>\j2ee\j2ee_<##>\os_libs
DIR_INSTANCE   = D:\usr\sap\<J2EE SID>\j2ee\j2ee_<##>\os_libs
DIR_PROFILE    = D:\usr\sap\<J2EE SID>\global\profile

# Here, specify the 3 character service SID:
SAPSYSTEMNAME  = <service SID>
# Here, specify the instance number of the J2EE Engine 6.20:
SAPSYSTEM      = <##>
# Here, specify the instance name:
INSTANCE_NAME = JC<##>
```

```
# Starts automatically this J2EE instance after restart
# of the physical machine or restart of the SAP Service if the value is
# set to 1. If 0, the automatic start is disabled.
Autostart = 0

#-----------------------------------------------------------------------
# Start Java Dispatcher
#-----------------------------------------------------------------------
_JC = jcontrol.exe
Start_Program_01 = local $(DIR_EXECUTABLE)\$(_JC) pf=$(DIR_PROFILE)\jcontrol_<##>.pfl

#--- last line - do not delete ---
```

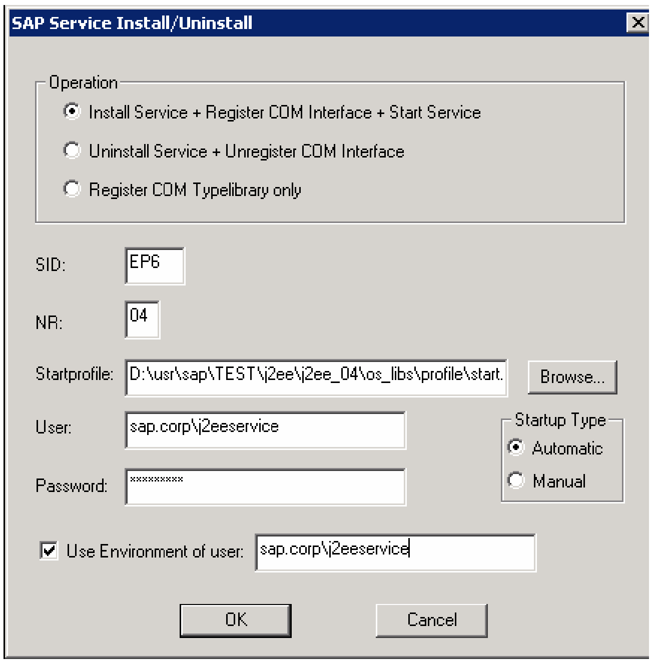
jcontrol_<##>.pfl

```
#-----------------------------------------------------------------------
# Runtime Variables
#-----------------------------------------------------------------------

DIR_EXECUTABLE = D:\usr\sap\<J2EE SID>\j2ee\j2ee_<##>\os_libs
DIR_LIBRARY    = D:\usr\sap\<J2EE SID>\j2ee\j2ee_<##>\os_libs
DIR_INSTANCE   = D:\usr\sap\<J2EE SID>\j2ee\j2ee_<##>\os_libs
DIR_PROFILE    = D:\usr\sap\<J2EE SID>\global\profile

# Here, specify the 3 character service SID:
SAPSYSTEMNAME  = <service SID>
# Here, specify the instance number of the J2EE Engine 6.20:
SAPSYSTEM      = <##>
# Here, specify the instance name:
INSTANCE_NAME = JC<##>

jstartup/instance_properties = D:\usr\sap\<J2EE SID>\j2ee\j2ee_<##>\configtool\service.ini

jstartup/start_mode          = program
jstartup/protocol            = on
jstartup/release             = 6.20
jstartup/vm/type             = hotspot

exe/jlaunch = $(DIR_EXECUTABLE)$(DIR_SEP)jlaunchep$(FT_EXE)

rdisp/TRACE = 1

#--- last line - do not delete ---
```

- For each instance in the cluster, open a command prompt and go to the folder shown above. In the command prompt, execute the following command:

  ```
  sapstartsrv.exe
  ```

  The following service configuration screen will appear:


SAP Service Install/Uninstall

Operation
[✓ Install Service + Register COM Interface + Start Service]
Uninstall Service + Unregister COM Interface
Register COM Typelibrary only

SID: EP6
NR: 04
Startprofile: D:\usr\sap\TEST\j2ee\j2ee_04\os_libs\profile\start.
User: sap.corp\j2eeservice
Password: *********

Startup Type
[✓ Automatic]
Manual

[✓ Use Environment of user:] sap.corp\j2eeservice

- Select the *Install Service + Register COM Interface + Start Service* option from the ***Operation*** pane
- In the ***SID*** field, enter the 3-character service SID that you have chosen for your installation
- In the ***NR*** field, specify the <##> instance number of your SAP J2EE Engine 6.20
- In the ***Startprofile*** field, use the *Browse...* button to navigate to your *start_<##>.pfl* file created previously
- In the ***User*** field, specify the Windows user name. If you want to start it with a domain user, use the following format:

```
<windows_domain>\<user_name>
```

- In the ***Password*** field, enter the user's password
- Choose the startup type of the service from the options in the ***Startup Type*** pane
- To run the SAP J2EE Engine 6.20 processes in a certain user environment, enable the ***Use Environment of user*** option and specify the user name of the user whose environment should be used. This is required, if you need access to the DLLs in the *os_libs* directory. In this case, you have to either specify this path in the Windows system *PATH* variable, or specify a user that has this path configured in its local *PATH* variable
- Choose ***OK*** to install and start the service.

Configuring the Service Control MMC Snap-In

To control the service with respect to starting/stopping one or more J2EE Engines, you have to add the SAP R/3 Systems Manager Snap-In to the Microsoft Management Console. Afterwards, you can control a set of local and remote SAP J2EE Engines 6.20 with the help of this snap-in.

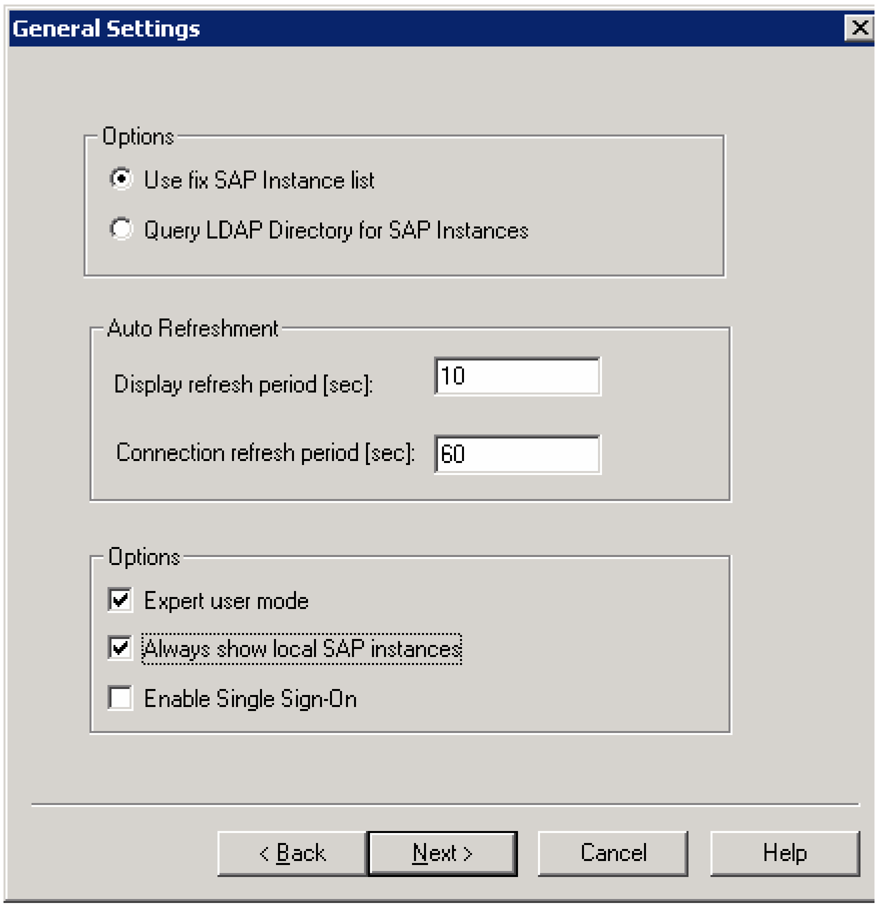
- Open a command prompt, go to the *%windir%\system32* directory of your Windows installation and run:

  ```
  regsvr32 sapmmc.dll
  ```

  Confirm the message dialog box stating that the registration of the DLL was successful. Leave the command prompt open for the next step. This will register the SAP R/3 systems manager snap-in.

- In the command prompt, run the *mmc* command. The Microsoft Management Console (MMC) will open. From the menu, choose *Console -> Add/Remove Snap-In*. In the *Add/Remove Snap-in* dialog box, choose the *Add...* button. From the shown selection list, choose the *SAP Systems Manager* Snap-In. Then, press the *Add* button.

  A configuration window will appear. Choose the corresponding options as shown below:


General Settings

Options
- [✓ Use fix SAP Instance list]
- Query LDAP Directory for SAP Instances

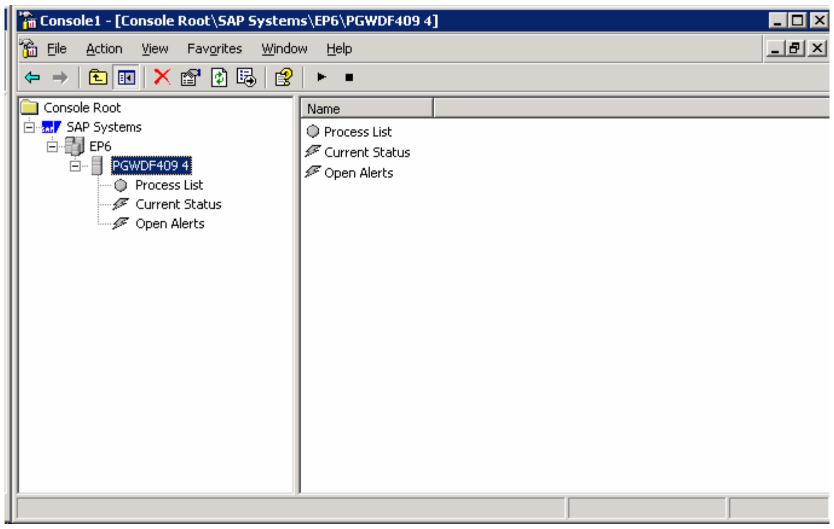
Auto Refreshment

Display refresh period [sec]: 10

Connection refresh period [sec]: 60

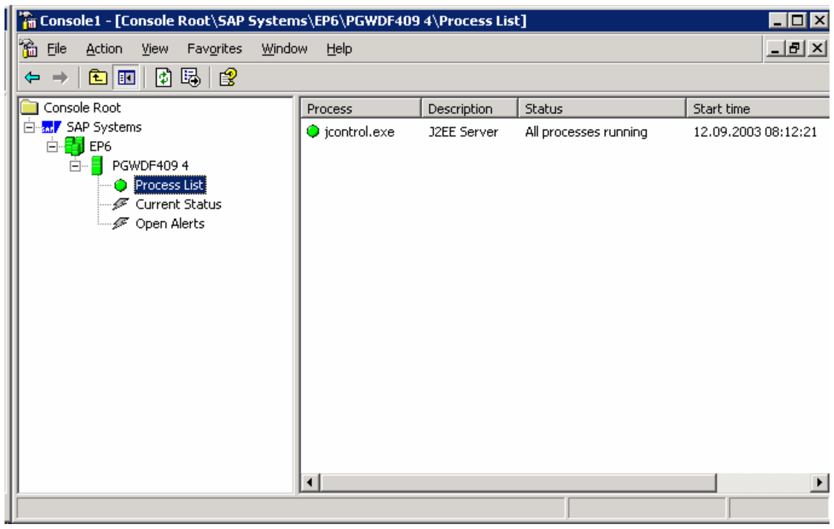
Options
- ✓ Expert user mode
- ✓ Always show local SAP instances
- Enable Single Sign-On

With these settings, all local instances of the SAP startup service will be displayed. Also notice the refresh rate of the snap-in information display and choose *Next >*. In the next window called *Fixed Server List*, choose *Finish*. Confirm all open windows. A Snap-in like the following should appear in the MMC window:



- Choose the small black triangle (▸) in the tool bar to let the service start the SAP J2EE Engine 6.20. Choose the *Process List* node to get detailed information about the current startup status. The color of the node will change from grey (no processes running), through yellow (some processes are running), to green (all processes are running).

  After a while (the service refreshes the display every 10 seconds) the nodes should appear green:



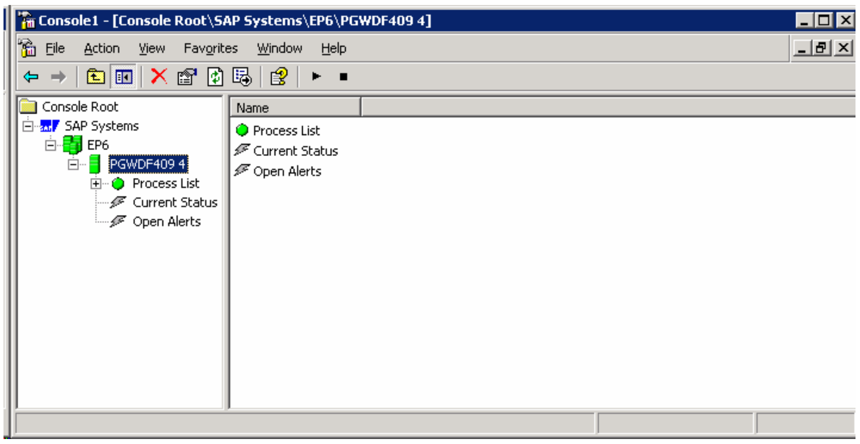
Process	Description	Status	Start time
jcontrol.exe	J2EE Server	All processes running	12.09.2003 08:12:21

- You can now save the MMC configuration into a file using the *Console* menu option *Save As…*. Later, by choosing the created file, the MMC will open your last view on the node tree and will start refreshing the display every 10 seconds (as long as the service has been started).

Managing the SAP Startup Service via the SAP MMC Snap-In

By using the SAP MMC Snap-In you can manage the SAP startup service in the following way:

- The start/stop/display statuses of the 6.20 J2EE instances
- The display log and trace files of the service, as well as of jcontrol and jlaunchep



1. Starting/Stopping SAP J2EE Engine 6.20

Select the host and instance number specific console node (here “PGWDF409 04”) and choose the ▸ button to start or the ■ button to stop the J2EE Engine.

2. Starting a Set of SAP J2EE Engines 6.20 Belonging to One Service SID

Select the SAP service SID specific console node (here “EP6”) and choose the ▸ button to start or the ■ button to stop all J2EE Engines belonging to this SID.

3. Starting All SAP J2EE Engines 6.20 Configured in the Current View of the *SAP Systems* Snap-In

Select the console node named *SAP Systems* and choose the ▸ button to start or the ■ button to stop all available J2EE Engines.

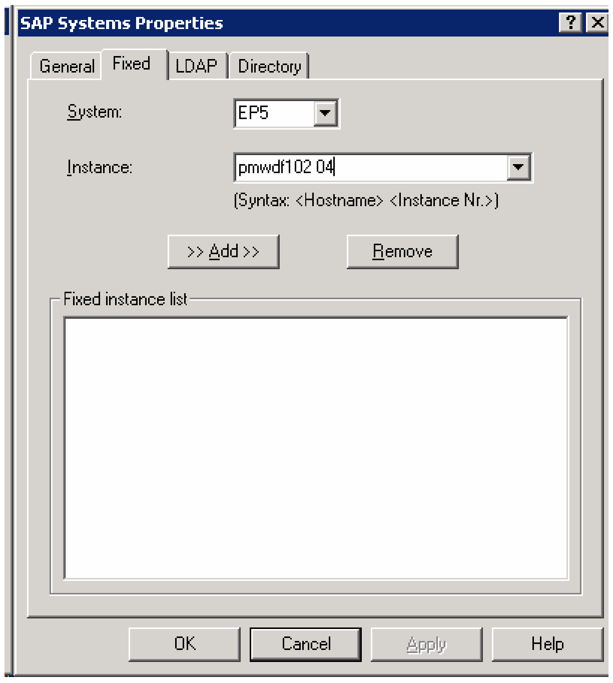
4. Displaying the Trace and Log Files

Display the trace or log files of individual processes, such as the service itself, as well as of jcontrol and jlaunchep, by right-clicking on the host and instance number specific console node (here, “PGWDF409 04”) and selecting *All Tasks*. In the context menu, you can choose the files you want to be displayed.

Adding More SAP J2EE Engines to the SAP MMC Snap-In

You can add more SAP J2EE Engines 6.20 to your current MMC view. The prerequisite for that is that you have set up startup services for those SAP J2EE Engines, as well.

- To add another **local** SAP J2EE Engine 6.20, you actually only have to configure the startup service for this J2EE Engine. Since the snap-in was configured to display all local instances of the SAP startup service, the new J2EE Engine will automatically appear.
- To add another **remote** SAP J2EE Engine6.20, right-click on the node named *SAP Systems* and select *Properties*. Choose the *Fixed* tab. The following configuration window will appear:

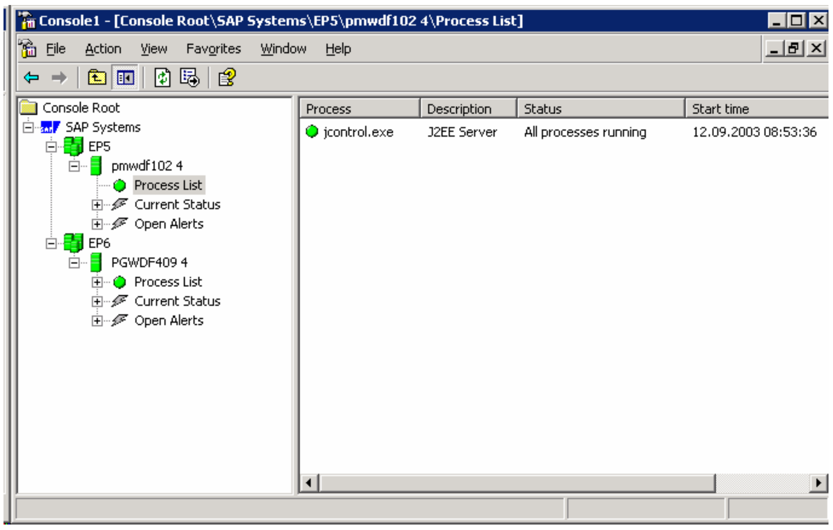


System: EP5

Instance: pmwdf102 04

(Syntax: <Hostname> <Instance Nr.>)

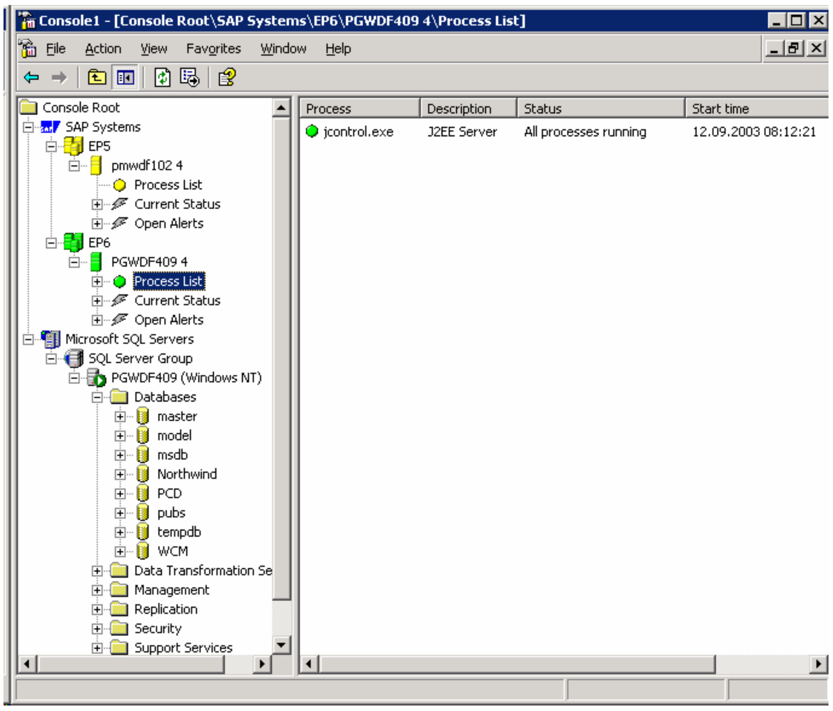
To add another system, enter the 3-character service SID used for the SAP startup service of the SAP J2EE Engine 6.20. In the *Instance* field, specify the remote host name, and a blank space, followed by the instance number of the SAP J2EE Engine 6.20. Choose *>> Add >>* to add it to the list named *Fixed instance list* displaying the remotely controlled instances. Make sure the remote SAP startup service is running before choosing *OK*. Choose *OK* when ready.



Process	Description	Status	Start time
jcontrol.exe	J2EE Server	All processes running	12.09.2003 08:53:36

You have now added a remote SAP J2EE Engine that you can start and stop via the current MMC view. Save the MMC view by using the *Console->Save As...* menu option to make the new settings permanent.

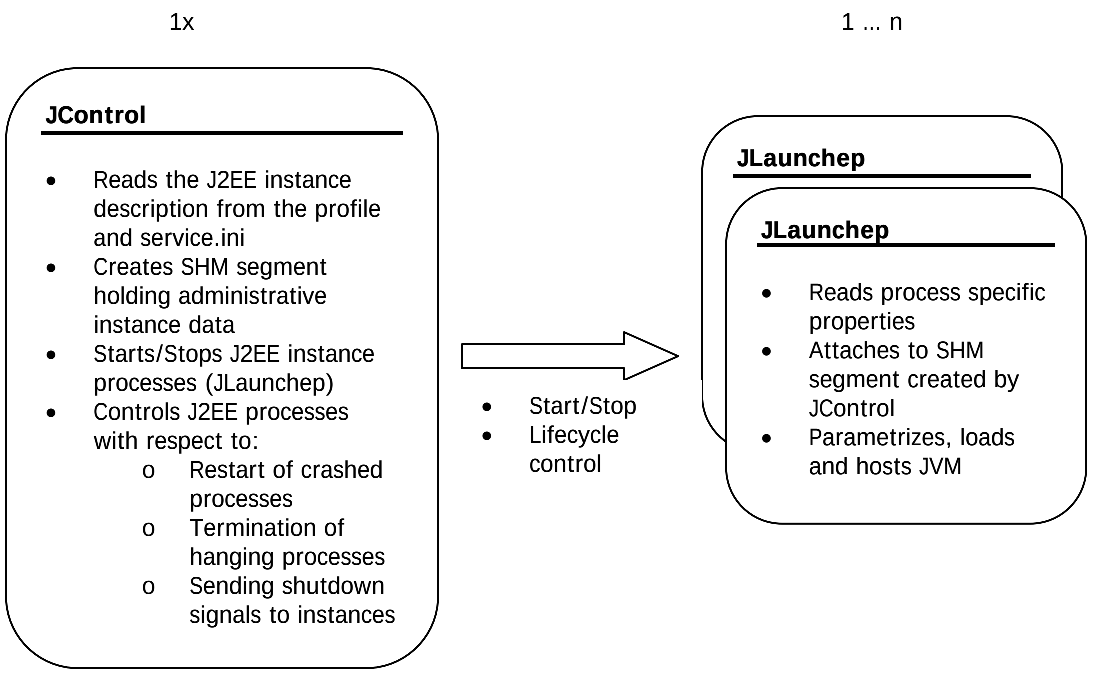
- You can also add any available snap-in into your current MMC view to manage your complete SAP J2EE Engine 6.20 cluster or a set of SAP J2EE Engine 6.20 and 6.30/6.40 clusters and databases from one single point of administration.



Process	Description	Status	Start time
jcontrol.exe	J2EE Server	All processes running	12.09.2003 08:12:21

General Information

Architecture


1x
1 ... n
JControl
• Reads the J2EE instance description from the profile and service.ini
• Creates SHM segment holding administrative instance data
• Starts/Stops J2EE instance processes (JLaunchep)
• Controls J2EE processes with respect to:
o Restart of crashed processes
o Termination of hanging processes
o Sending shutdown signals to instances
• Start/Stop
• Lifecycle control
JLaunchep
JLaunchep
• Reads process specific properties
• Attaches to SHM segment created by JControl
• Parametrizes, loads and hosts JVM

Operation

The service|deamon creates trace and log files in the following directory:

- Windows
  `<drive>:\usr\sap\<SID>\j2ee\j2ee_00\os_libs\work`
- UNIX
  `/usr/sap/<SID>/j2ee/j2ee_00/os_libs/work`

The files are:

sapstart.log

A log file of the SAP service, containing the log entries about the last start request sent to the service.

sapstart.trc

A trace file of the SAP service, holding trace information about the control request signals sent to the SAP service since its last restart.

dev_jcontrol

A trace file of the jcontrol process.

dev_<n>

A trace file of the SAP J2EE cluster nodes' jlaunchep processes.

You can set the current trace level for the trace files with the profile variable ***rdisp/trace*** in the *jcontrol.pfl*. The trace levels can be defined from 0 (lowest) to 3 (highest).