Applies to:
This article applies to anyone working with J2EE.

Summary
This is a general troubleshooting guide to assist you in solving J2EE engine start-up problems. It details the common problems and outlines the best places to look in order to pinpoint the cause of failure.

Author(s): Kenny Tully
Company: SAP
Created on: 23 November 2007

Author Bio
Kenny Tully has worked in the J2EE/NWDI team in GSC Ireland for 1.5 years and has been working in Basis Software Logistics since joining SAP in Jan 2003.
Table of Contents

Start-Up Procedure ....................................................................................................................................... 3
Start-Up Profiles ........................................................................................................................................ 5
Start Up Process ....................................................................................................................................... 6
Processes in Unix....................................................................................................................................... 7
Tools for Start-Up Analysis ......................................................................................................................... 12
Work Directory ......................................................................................................................................... 12
Dev_jcontrol ............................................................................................................................................... 12
Bootstrap Logs .......................................................................................................................................... 12
Dev_server & Dev_dispatcher ..................................................................................................................... 13
Std_server & Std_dispatcher ....................................................................................................................... 13
DefaultTrace.X.trc.................................................................................................................................... 13
Common Problems ..................................................................................................................................... 14
Message Server Connection ...................................................................................................................... 14
Memory Fragmentation ............................................................................................................................... 14
Note 940893 ............................................................................................................................................. 14
Copyright..................................................................................................................................................... 15
Start-Up Procedure

- There are different methods of starting the J2EE in NetWeaver ´04 depending on the installation type the user has.
- With a J2EE-Only installation the user will use the start-up Framework to start the Engine. This is done via the SAP MMC on the Windows server. There are slight differences in the way this works in windows and UNIX installations.
- On a UNIX box, the System is started and stopped by running the 'startsap' and 'stopsap' scripts. These are located in the /usr/sap/<SID>/ SYS/exe/run directory
Start-Up Procedure (continued)

- The above diagram illustrates the start-up procedure of the J2EE Engine.

- The start-up Framework or the UNIX scripts launch 'Jcontrol' which reads profiles located in /usr/sap/<SID>/SYS/profile for parameters required to start the Instance. 'Jcontrol' is in essence, the master program. In Windows, the System is started from the SAP MMC which should be present on the servers desktop. When started this way, the profiles are also read from the same location as they are with the UNIX installation. There are important profile files that should be present in this directory. These profiles are detailed on the next page.
## Start-Up Profiles

- **DEFAULT.PFL** - This contains the SAPSYSTEMNAME, the dbname, the dtype and the dbhost.
- **START_SCS<InstanceNumber>_<host>** e.g. START_SCS01_us4025. This sets global variables and starts the Messaging Service and Locking Service (Enque Server).
- **<SID>_SCS<InstanceNumber>_<host>** e.g. J2E_SCS01_us4025. This contains parameters for the messaging service and enque service.
- **START_JC<InstanceNumber>_<host>** e.g. START_JC00_us4025. This sets Global Variables and start the J2EE (J2EE Engine).
- **<SID>_JC<InstanceNumber>_<host>** e.g. J2E_JC00_us4025. This profile contains information on where to find instance properties and sdm properties. This file also contains the location of ‘jlaunch’ which is required to start the J2EE Engine. ‘Jlaunch’ is normally located in the /usr/sap/<SID>/JC<InstanceNumber>/j2ee/os_libs directory.

### Table: Start-Up Profiles

<table>
<thead>
<tr>
<th>Profile Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFAULT.PFL</td>
<td>Contains SAPSYSTEMNAME, dbname, dtype, and dbhost.</td>
</tr>
<tr>
<td>START_SCS&lt;InstanceNumber&gt;_&lt;host&gt;</td>
<td>Sets global variables and starts Messaging Service and Locking Service (Enque Server).</td>
</tr>
<tr>
<td>&lt;SID&gt;<em>SCS&lt;InstanceNumber&gt;</em>&lt;host&gt;</td>
<td>Contains parameters for messaging and enque service.</td>
</tr>
<tr>
<td>START_JC&lt;InstanceNumber&gt;_&lt;host&gt;</td>
<td>Sets Global Variables and starts J2EE Engine.</td>
</tr>
<tr>
<td>&lt;SID&gt;<em>JC&lt;InstanceNumber&gt;</em>&lt;host&gt;</td>
<td>Contains information on instance properties and sdm properties. Includes location of 'jlaunch'.</td>
</tr>
</tbody>
</table>
Start Up Process

startsap (UNIX) / SAP Service (Windows)

-> SAPOsCol

-> Database Instance (if local)

-> Java Message Server

-> Java Enqueue Server

**JControl**
- Reads Java instance description from profile and instance.properties
- Create Shared Memory Segment holding administrative instance data
- Starts/ Stops Java VM hosting processes (JLaunch) for each Java Dispatcher, Java Server Process and SDM

**JLaunch (per VM process)**
- Reads instances specific properties
- Parameterizes, loads and hosts Java VM to start the process
- Attaches instance to SHM segment created by JControl

Web AS Java Startup Framework
Processes in Unix

Window Edit Options

Example of startup of OS processes:

Startup of sapstart (PID 3175) => Reading Start-Profile of SCS-Instance
   ⇒ Starting of Message Server => startsap (PID 3175) is parent (ms.sapJ2E_SCS01= PID 3184 )
   ⇒ Starting of Enqueue Server => startsap (PID 3175) is parent (en.sapJ2E_SCS01= PID 3185 )

Startup of sapstart (PID 3363) => Reading Start-Profile of JC00-Instance
   ⇒ Starting of jcontrol => startsap (PID 3363) is parent (jc.sapJ2E_JC00 = PID 3368)
      ⇒ Starting of jlaunch => jcontrol (PID 3368) is parent (jlaunch = PID 3387)
      ⇒ Starting of jlaunch => jcontrol (PID 3368) is parent (jlaunch = PID 3388)
      ⇒ Starting of jlaunch => jcontrol (PID 3368) is parent (jlaunch = PID xxxx)

Startup of igswd_mt (PID 3411) => Starting the Internet Graphics Server

   On a UNIX box, if the Engine has started properly, then the processes should appear as shown below. They can be viewed by using the following commands:
   • ps –ef|grep jlaunch and ps –ef|grep sapstart etc.
On Windows, you can use TaskManager. As illustrated on the diagram above, the various processes are clearly visible when the J2EE is up and running.
There is also a tool called JCMon that can be used to analyze the condition of the J2EE after Start-Up. As seen in the screenshot above, this tool tells us the state of each of the nodes. Therefore, if one wants to know if the dispatcher starts successfully, they can use this tool. This can be started in the following manner:

```
jcmon "pf=/usr/sap/<SID>/sys/profile/<SAPSID>_<INSTANCE_ID>_<HOST>"
```

Example:
```
jcmon pf=/usr/sap/C11/sys/profile/C11_JC00_testpc.
```

More information on this tool can be found at this location:
```
http://help.sap.com/saphelp_nw04/helpdata/en/d3/4d074147c1f06fe10000000a1550b0/frameset.htm
```
As mentioned, in the J2EE Add-In Installation, the J2EE can also be started using the SMICM transaction in the R/3. In cases where the J2EE is not starting up in this manner, there are a number of parameters that can be checked. On the R/3 System, open transaction RZ10 and choose the profile of the SAP WebAS. Check the parameter rdisp/j2ee_start is set to 1. You should also check the patches to the Start-Up Framework.

Missing J2EE Settings in R/3: 741289
Tools for Start-Up Analysis

Work Directory
On a standalone is located at: /usr/sap/<SID>/JC<Instance Number>/work
On a Add-In, it is located at /usr/sap/<SID>/DVEMBGS<Instance Numbers>/work
On a dialog, it is located at /usr/sap/<SID>/J<Instance Number>/work
The first thing to ask the customer for is the complete work directory. Just ask them to archive (Zip) it up and attach it to the message
Because files are over written at each start-up, sometimes it is best that you ask the customer to restart the J2EE once more before attaching the logs. Then you have a fresh set of logs to look at.

Dev_jcontrol
This is the trace for the Jcontrol process.
It will tell you how far the start-up process has progressed and approximately where it has failed.
Example 1
Dev_Jcontrol
[Thr 3316] JControlICheckProcessList: process server0 started (PID:3972)
N.B. The memory settings are included in the dev_jcontrol

Bootstrap Logs
There are a number of logs/traces for each bootstrap
- Main bootstrap for instance
- Bootstraps for each Node
Example 2
Dev_bootstrap
[Thr 3420] JLaunchIExitJava: exit hook is called (rc=0)
[Thr 3420] JLaunchCloseProgram: good bye (exitcode=0)
**Dev_server & Dev_dispatcher**
These are the trace files of the JLaunch processes
Useful for checking the JVM parameters for each process
- Memory settings, etc.
Also useful for checking at which stage the start-up process failed if it gets this far
- [Waiting for start (1)] to [Starting (2)]
- [Starting (2)] to [Starting applications (10)]

Example 3
Dev_server0
[Thr 4788] JLaunchISetState: change state from [Starting applications (10)] to [Running (3)]

**Std_server & Std_dispatcher**
Output of the JVM
Shows which services failed during start-up
Core Services (e.g. Security - com.sap.security.core.ume.service) are required for successful start-up
Example 4
Std_dispatcher0.out
ServiceManager started for 16303 ms.
Framework started for 25666 ms.
SAP J2EE Engine Version 6.40   PatchLevel 100627.313 is running!
PatchLevel 100627.313 December 14, 2005 20:06 GMT

**DefaultTrace.X.trc**
DefaultTrace for the server is located at: /usr/sap/<SID>/<Instance>/j2ee/cluster/serverX/log
Useful for determining if the problem is actually within an application starting up
Check if the customer can load the ‘Default’ page
- http://<J2EE_Host>:<port>
If so, it is a problem with the application starting and check the defaultTrace
Also in the dev_serverX, never goes from:
- [Starting applications (10)] to [Running (3)]
If it seems to hang/get stuck between ‘Starting Applications’ and ‘Running’ it is useful to trigger thread dumps
#
- If it seems to hang/get stuck between ‘Starting Applications’ and ‘Running’ it is useful to trigger thread dumps.
- Such cases are common with XI start-Ups. From the thread dumps we can often see that there are not enough Application Threads allocated
Common Problems

Message Server Connection

Jcontrol cannot connect to the Message Server host/port
Often the actual port number is wrong
Check what port the MS is listening on in:
- /usr/sap/<SID>/SCS<Inst>/work/dev_ms
- I listen to internal port 3900 (3900)

Compare with the port that Jcontrol tries to connect to in the Dev_jcontrol
- Instance properties
  - -> ms host    : usilap34
  - -> ms port    : 3601

If incorrect, they need to change this in the configtool
Make sure that they Save any changes and Restart

Memory Fragmentation

In the Std_bootstrap:
Error occurred during initialization of VM. Could not reserve enough space for object heap
Fixed through Note 835704
Must start jlaunch in this Note and this may not be possible.
Solution:
Reduce the value of the MaxHeapSize property of the server node via the configtool, navigate to Instance_ID# -> Server_ID# -> Bootstrap (256Mb)
Restart Jlaunch and Rebase
Increase XMX again if required

Note 940893

JDK Detection Issues (One not mentioned)
- JStart-upIGetJDKInfo: Cannot find Java class
- Permission issue

Java VM Shared library issues
Java VM Initialization Issues
Java Terminated with a Non-zero Exit code