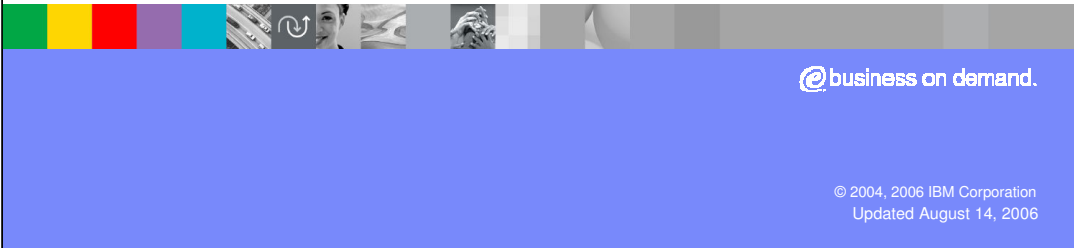




IBM Software Group

IBM WebSphere® Application Server V6

Naming Problem Determination



This presentation will focus on problem determination within the Naming area. Prior to reviewing this presentation, it would be helpful to have reviewed the Naming Introduction and Naming Basics presentations. If you plan on debugging problems in a Network Deployment environment, it would be helpful to have previously reviewed the Advanced Naming presentation.

Goals

- Introduce dumpNameSpace utility
 - ▶ Extremely helpful for debugging name space related problems
- Illustrate usage for dumping various name spaces
 - ▶ Global name space for a single server
 - ▶ java: name space for a specific Enterprise Java™ Bean (EJB) or Web module
 - ▶ local: name space for an application server
 - ▶ java: name space in a J2EE client

WebSphere Application Server V6 provides a utility called dumpNameSpace which, as the name implies, allows you to dump a name space and thus allows you to examine the bindings it contains. The purpose of this presentation is to introduce the tool and illustrate how to invoke it for dumping the different types of name spaces within the V6 Application Server.

dumpNameSpace – Global name space

- Dump contents of
 - ▶ Name space for a single server
 - ▶ Does **not** present full logical view of the name space
 - ▶ Shows CORBA URLs where the name space transitions to another server
- Prevents dumping of cycles in the name space
- Must use correct host and port
 - ▶ Must point at the server you want to dump
 - ▶ Common usage problem is using default port in ND environment
 - This points to the Node Agent
 - Usually you want to dump an Application Server
- Invoked as a command line utility

The most common use of dumpNameSpace is to dump some portion of the global name space. When used to dump the global name space, the utility is invoked on the command line.

To understand the output, you need to recall that the global name space is partially replicated in all servers and partially distributed across servers. This means that the portion of the global name space physically contained within any one server is a subset of the logical global name space and that there are some bindings which identify contexts that are actually in other servers. When using dumpNameSpace, it is the physical portion of the name space contained within the server that is being dumped. Transitions to contexts in other servers are not traversed as they would be in a normal lookup. At the point where a binding identifies a context in another server, the dumpNameSpace utility just jumps the “corbaloc:” URL for that context. This would allow someone to initiate a dumpNameSpace against that server at that context if needing to traverse into that context.

In the name space, there can be multiple references to the same context. For example, the node root context exists in the name space at “nodes/<nodename>”, but there is also a binding to it from the server root context named “thisNode”. The dumpNameSpace utility is smart enough to detect this and would only dump the context at “nodes/<nodename>” and at “thisNode” would provide information that the binding is actually a link to the context. This prevents the utility from entering into endless loops, dumping the same contexts over and over again.

dumpNameSpace – Example Usage

- Get help on options:
 - ▶ `dumpNameSpace -?`
- Dump server on localhost:2809 from cell root
 - ▶ `dumpNameSpace`
- Dump server on localhost:2806 from cell root
 - ▶ `dumpNameSpace -port 2806`
- Dump server on yourhost.ibm.com:2811 from cell root
 - ▶ `dumpNameSpace -port 2811 -host yourhost.ibm.com`
- Dump server on localhost:2809 from server root
 - ▶ `dumpNameSpace -root server`
- Dump server on localhost:2809 from subcontext of server root (dump datasources used with CMP 2.0 beans)
 - ▶ `dumpNameSpace -root server -startAt eis`
- Dump server on localhost:9810, getting a long report
 - ▶ `dumpNameSpace -port 9810 -report long`

This slide gives some sample invocations of the `dumpNameSpace` command line utility for dumping the global name space.

There are many options with `dumpNameSpace` and it is recommended that the “-?” parameter be used to dump the full set of parameters along with an explanation of what they mean. Understanding these parameters will make the tool more useful.

The use of the “-host” and “-port” parameters should be self evident as explained in the examples.

It is often much more useful to dump a subset of the name space rather than the entire name space physically contained in the target server. For example, nine times out of ten what you are looking for will be in the server root context. To start the dump at the server root context, use the “-root server” parameter. You may even know that what you want to see is in a sub-context, such as “eis” which is where the datasources used with CMP 2.0 Entity EJBs are located. In that case, you can use the parameters “-root server -startAt eis” to just dump that context and any of its sub-contexts.

Another useful option is the “-report long” option. It allows you to get additional information about the objects at each binding. Because it dumps several lines for each binding, it is often useful to make use of the “-root” or “-startAt” parameters – or both - to limit the amount of the name space that is dumped.

IBM Software Group IBM

```

C:\was60\0442.23\profiles\Dmgr01\bin>dumpNameSpace -port 9811 -root server

Getting the initial context
Getting the starting context

-----
Name Space Dump
Provider URL: corbaloc:iiop:localhost:9811
Context factory: com.ibm.websphere.naming.WsnInitialContextFactory
Requested root context: server
Starting context: (top)=rInt40Cell/nodes/rInt40Node1/servers/server1
Formatting rules: jndi
Time of dump: Wed Nov 03 15:03:02 CST 2004
-----

Beginning of Name Space Dump
-----
1 (top)
2 (top)/mail
3 (top)/mail/PlantsByWebSphere
4 (top)/cell
4 Linked to context: rInt40Cell
5 (top)/services
6 (top)/services/cache
7 (top)/services/cache/distributedmap
8 (top)/services/cache/basecache
9 (top)/wm
10 (top)/wm/default
11 (top)/jta
12 (top)/jta/usertransaction
13 (top)/tm
14 (top)/tm/default
15 (top)/DefaultDataSource
16 (top)/eis
17 (top)/eis/jdbc
18 (top)/eis/jdbc/PlantsByWebSphereDataSource_CMP
19 (top)/eis/DefaultDataSource_CMP
20 (top)/thisNode
20 Linked to context: rInt40Cell/nodes/rInt40Node1
21 (top)/servername
22 (top)/jdbc
23 (top)/jdbc/DefaultEJBTimerDataSource
24 (top)/jdbc/PlantsByWebSphereDataSource
-----
End of Name Space Dump
-----

```

Example Output – Server Root Context

5

Naming - Problem Determination © 2004, 2006 IBM Corporation

This is an example output from `dumpNameSpace`. It has been invoked as “`dumpNameSpace -port 9811 -root server`”, so what is seen is the server root context for the server on localhost listening to bootstrap port 9811.

Notice that the first thing dumped is header information. The most interesting piece of information in the header is the starting context. This identifies the top level context dumped. It identifies the context using its fully-qualified name starting with the name of the cell. In this example, the cell name is “`rInt40Cell`” and the node name is “`rInt40Node1`” and the server root context is being dumped for “`server1`”. Therefore the starting context shows “`rInt40Cell/nodes/rInt40Node1/servers/server1`”. Also note that the header indicates that “(top)” is used to identify this context in the line by line output in the dump.

You may recall in the Advanced Naming presentation the discussion that from the server root context in any server, you could lookup the string “`cell`” and be positioned at the cell root context. Take a look at line 4 of the output. Notice that the binding “`cell`” is shown as a link to the context “`rInt40Cell`”, which is the cell root context. Line 20 also shows a linked context for the binding “`thisNode`” which shows the link to the context “`rInt40Cell/nodes/rInt40Node1`”.

Having looked at these two linked contexts, you see that there are two lines marked 4 and two lines marked 20. This is because these are not really line numbers, but rather binding numbers, used to help identify the output when a binding spans multiple lines. This especially comes into play when the “`-report long`” option is used.

Example Output – corbaloc URL to Other Server

```

C:\was60\0442.23\profiles\Dmgr01\bin>dumpNameSpace -port 9811 -startAt nodes/rInt40Node1/servers
Getting the initial context
Getting the starting context
-----
Name Space Dump
Provider URL: corbaloc:iiop:localhost:9811
Context factory: com.ibm.websphere.naming.WsnInitialContextFactory
Requested root context: cell
Starting context: (top)=rInt40Cell/nodes/rInt40Node1/servers
Formatting rules: jndi
Time of dump: Wed Nov 03 15:34:57 CST 2004
-----
Beginning of Name Space Dump
-----
1 (top)
2 (top)/myServerB1                               javax.naming.Context
3   Linked to URL: corbaloc::rInt40.austin.ibm.com:9814/NameServiceServerRoot
4 (top)/nodeagent                               javax.naming.Context
5   Linked to URL: corbaloc::rInt40.austin.ibm.com:2809/NameServiceServerRoot
6 (top)/myServerB1                               javax.naming.Context
7   Linked to URL: corbaloc::rInt40.austin.ibm.com:9813/NameServiceServerRoot
8 (top)/server1                                 javax.naming.Context
9 (top)/server1/mail                            javax.naming.Context
10 (top)/server1/mail/PlantsByWebSphere         javax.mail.Session
11 (top)/server1/cell                            javax.naming.Context
12   Linked to context: rInt40Cell
13 (top)/server1/services                       javax.naming.Context
14 (top)/server1/services/cache                 javax.naming.Context
15 (top)/server1/services/cache/distributedmap  com.ibm.websphere.cache.DistributedObjectCache
16 (top)/server1/services/cache/basecache      com.ibm.websphere.cache.DistributedObjectCache
17 (top)/server1/vm                             javax.naming.Context
18 (top)/server1/vm/default                     com.ibm.websphere.asynchbeans.WorkManager
19 (top)/server1/its                             javax.naming.Context

```



Remember that `dumpNameSpace` will only dump the contents of the physical name space in the server being dumped. This example shows what a binding looks like when it is bound to a context physically contained in another server. In this case, notice that the dump is starting at the “`rInt40Cell/nodes/rInt40Node1/server`” context, or in other words, dumping all the servers on node “`rInt30Node1`”. Notice that the dump is not from the server listening on port 9811, which in this case happens to be “`server1`”. Look at the binding on line 2 which is to the server root context for server “`myServerB1`”. The output of `dumpNameSpace` is to show this as a link to URL “`corbaloc::rInt40.austin.ibm.com:9814/NameServiceServerRoot`”. Now look at line 5. This is the binding to the server root context for “`server1`”, and since this is “`server1`” being dumped it is not shown as a link but is traversed as shown in lines 6-15.

dumpNameSpace – java: & local:

- Dump the in-memory name spaces
- Every EJB and Web Module has a java: name space
- Every Application Server has a local: name space
- Dump using wsadmin and NameService mBean
 - ▶ Establish wsadmin prompt, connected to the proper server
`wsadmin -conntype RMI -port 9810`
 - ▶ Get the NameServer mBean
`set mbean [$AdminControl completeObjectName WebSphere:*.type=NameServer]`
 - ▶ Dump the local: name space
`$AdminControl invoke $mbean dumpLocalNameSpace {{options}}`
 - ▶ Dump java: for appName/modName/componentName
`$AdminControl invoke $mbean dumpJavaNameSpace {{appName}{modName}{compName}{options}}`

In the Naming Introduction and the Basic Naming presentations, the concepts of the “java:” and “local:” name spaces were explained. These name spaces are only accessible from within a server and therefore cannot be bootstrapped into the global name space. This slide shows how to dump these name spaces using wsadmin commands. Since there are multiple “java:” name spaces in a server, the Application Name, Module Name and Component Name must be used as qualifiers.

dumpNameSpace – J2EE Clients

- Dump the java: name space in a J2EE client

```
launchClient myClientApp.ear -CCdumpJavaNameSpace=true
```

- -CCdumpJavaNameSpace=xxx, where xxx:
 - ▶ false – do not dump name space (default)
 - ▶ true – dump using short report format
 - ▶ long – dump using long report format

A J2EE client process also has a “java:” name space. It is dumped using a parameter to the launchClient command as is illustrated.

Summary

- Introduced dumpNameSpace utility
 - ▶ What it does
 - ▶ Sample command line invocations
 - ▶ Sample output
- Usage for
 - ▶ Global name space
 - ▶ java: name spaces, client and server
 - ▶ local: name space

This presentation examined the dumpNameSpace utility and how to use it for debugging name space problems.

Trademarks, Copyrights, and Disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM	CICS	IMS	MQSeries	Tivoli
IBM (logo)	Cloudscape	Informix	OS/390	WebSphere
e (logo) business	DB2	iSeries	OS/400	xSeries
AIX	DB2 Universal Database	Lotus	pSeries	zSeries

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds.

Other company, product and service names may be trademarks or service marks of others.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or program(s) described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2004, 2006. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.