# **User's Guide**



# Inprise Application Server

Inprise Corporation, 100 Enterprise Way Scotts Valley, CA 95066-3249

#### Chapter

# Starting the Inprise Application Server

The Inprise Application Server is a set of services and tools that enable your enterprise to build, deploy, and manage web applications. These applications provide dynamic content by using java, servlets, and Enterprise Java Beans (EJB) technologies.

This chapter gives an overview of starting and stopping the server, changing server configurations, and managing top-level services.

## About the top-level services

Services included with the Inprise Application Server are:

- **EJB Containers:** Use this service to deploy EJBs and monitor EJB performance. Tools include a Deployment wizard and a Deployment Descriptor Editor. You can create and manage as many EJB Containers as desired.
- **CORBA services:** A collection of Inprise CORBA services including the VisiBroker Object Request Broker (ORB) and Smart Agent (osagent).
- Naming Services: The Naming Service associates meaningful names with objects, and then uses those names in a lookup facility. The Inprise Application Server supports two naming services and includes a JNDI interface for EJB applications.
- **JDataStores:** A database service written entirely in Java for embedded, web, and mobile database applications. You can create and manage as many JDataStores as desired.
- **Transaction engines:** Transaction services provide tools to develop and manage transactional applications, including the ability to access a database within a transaction. The Inprise Application Server supports two transaction engines, JTS and ITS.

• Web Service Engines (or Web Engines): Designed to support development and deployment of web applications, this web service and underlying HTTP engine includes a graphical interface for server configuration and administration. The server includes support for both standard (http) and secure (https) Web Service Engines.



Figure 2.1 Using the Inprise Application Server Console to view and manage services

#### Architectures supported

Designed specifically for EJB development, deployment, and management, the Inprise Application Server supports a variety of server architectures for Java and web applications. Common ways enterprise beans can be accessed include:

- Web browser client can invoke a JSP or servlet that invokes the enterprise bean that communicates to a backend process (such as a database).
- Applet, stand-alone Java application, or CORBA client can invoke the enterprise bean directly.



Figure 2.2 Common architectures

### Separation of development and deployment

The Inprise Application Server supports the design model described in version 1.1 of the EJB Specification published by Sun Microsystems. In this model, development tasks are separated from deployment to maximize engineering resources.

The following is an overview of the steps needed to develop and deploy EJBs using the Inprise Application Server. Deployment steps are described in detail in the remaining chapters.

#### Developing the beans

The application developer creates enterprise beans and servlets using JBuilder (or a text editor). All three types of beans are supported: Stateless Session, Stateful Session, and Entity.

- For every enterprise bean, development includes the following three pieces:
  - Enterprise bean class
  - Home interface
  - Remote interface

The home interface is an EJB component interface that enables clients to look up and create EJB objects. The remote interface is an EJB component interface that enables clients to interact with an EJB object on a server.

• The application developer compiles the remote interface, home interface, and implementation class for each enterprise bean.

- Once the appropriate beans are created and compiled, the application developer
  - Creates a deployment descriptor for each enterprise bean
  - Defines a primary key for each Entity bean
  - Bundles the deployment descriptor and enterprise bean classes (bean class, remote interface, home interface, and primary key) into a JAR file

Inprise provides a variety of tools to aid the development of EJBs. For more information, see the Borland JBuilder *User's Guide* and the Inprise Application Server *Enterprise JavaBeans Programmer's Guide*.

Figure 2.3 Using JBuilder to create enterprise bean interfaces

👹 Generate Enterprise JavaB	ieans Interfaces 🛛 🗙
EJB Component: AccountBean	
Methods DeploymentDescriptor	Environment Properties ControlDescriptor
Enterprise Bean Class Name:	AccountBean
Home Interface Class Name:	AccountHome
Remote Interface Class Name:	Account
Bean Home Name:	Account
Session Timeout:	30
State Management Type:	SessionDescriptor.STATELESS_SESSION
	OK Cancel Help

#### Deploying the beans

Attributes for transactions, security, and mapping to EJB-Container data sources all can be handled at deployment. Developers don't have to worry about these details when developing EJBs.

- Deploying an enterprise bean includes the following tasks:
  - Specifying the environment properties that the EJBs require at runtime
  - Deploying the JAR file to an EJB Container
  - Monitoring the deployment
- The Inprise Application Server provides a default EJB Container for deployment. You can create additional Containers as needed. (See "Creating a new Container" on page 3-20.)
- Deployment tools include a Deployment wizard for deploying beans to the Container and Deployment Descriptor Editor for setting or changing information in the deployment descriptor.
- The Deployment wizard creates the Inprise-specific stubs and skeletons necessary for deploying the enterprise bean to the EJB Container.

#### Writing and running clients

- Once the beans are in place on the server, developers write clients to access the beans. The Inprise Application Server supports EJB clients written using RMI or CORBA.
- Users run the client as needed.
  - Figure 2.4 Inprise Application Server and EJB Containers



## Starting the server

The Inprise Application Server can be installed and started either locally or on a shared machine. The server, in turn, starts all services configured for the server. These include the Web Service Engine, database, Smart Agent, EJB Containers, and so on.

### Starting a default server

To start a default version of the server:

- WinNT Click the Start button and choose Application Server from the Inprise Application Server program group, or
- WinNT/ Open a command window and type the following: UNIX

ias

Note To recognize the ias command, you must have your path updated to include the ias bin directory (*install\_dir/*bin), or you can enter the path explicitly.

Once the server has started, you can specify which services are started with the server. See "Managing top-level services" on page 2-10 for more information.

**Important** The ias command includes command options you can use to start the server in something other than the default configuration. For more information, see "Server configurations" on page 2-14.

## Starting the Console

The Inprise Application Server includes a GUI-based Console which functions as the main control point for the Application Server. The Console enables you to view servers on the network, start and stop services, and so on. The Console also enables you to view EJB JAR files and Containers, set deployment properties, and monitor performance.

The Inprise Application Server typically runs on a large shared UNIX or Windows NT machine, while the Console runs on any machine from which users want to view or modify the distributed system. Once the Console is installed, you can deploy to any Application Server on your network.

To start the Console:

1 Make sure an Inprise Application Server is started.

You must start an Application Server before you start the Console.

- **2** Use one of the following methods to start the Console:
- **WinNT** Click the Start button and choose Console from the Inprise Application Server program group.
- Open a command window and enter the following command:

console.sh

**Note** To recognize the console command, you must have your path updated to include the Console bin directory (*install\_dir*/console/bin), or you can enter the path explicitly.

The Console window appears. The Console enables you to view all instances of the Inprise Application Server that share the same Smart Agent port.

For more information on using the Console to view servers and services, See "Managing top-level services" on page 2-10.

#### Setting Console preferences

Console preferences enable you to specify the Smart Agent port used by the Console, the default polling interval for performance information displayed in the Console, and so on.

To set Console preferences:

1 Start the Console and choose Preferences from the File menu.

A dialog box appears with a list of preferences.

#### Figure 2.5 General preferences

Preferences	X
General AppServer VisiBroker	
S Inprise Application Server Console	
User Interface:	
Look and Feel: Metal 👻	
HTML Browser:	Browse
✓ Sound beep on error	
Network:	
✓ Lookup host names for IP addresses	
Debuging:	
Debug Level: 0 - None 🔻	
	Ok Cancel

2 Set preferences as desired and click OK.

General preferences include:

- Look and Feel: Sets the display format and behavior of the Console windows (for example, Windows or CDE/Motif).
- **HTML browser:** Specifies the browser you wish to use with the Application Server.
- Sound beep on error: Enables an audible beep if an error occurs.
- **Debug Level:** Sets the level of debug messages displayed in the Console. There are 4 debug levels (0 to 3) with 3 being the most messages displayed.
- Lookup host names for IP addresses: Uses host names as well as IP addresses. If you are not running a DNS to resolve host names on your network, you may wish to uncheck this setting to enhance system performance.

There are three panels of Inprise Application Server preferences.

Figure 2.6 Inprise Application Server preferences

Preferences	X	
Inprise Application Serve	er	
Server Discovery Polling Log Polling: Log File Polling Interval: 60	oooo (ms) 🗆 Disabled	
History: Show complete file Show at most	Preferences     General AppServer VisiBroker     Inprise Application Server     Server Discovery Polling Logs     Polling Intervals:     Server Ping Interval: 2000     Statistics Polling Interval: 5000	00     (ms)     Disabled       00     (ms)     Disable
		Add Server Remove

Log file preferences include:

- Log File Polling Interval: Sets the time interval (in milliseconds) for how often log files are updated. The Disabled checkbox lets you disable polling.
- Show complete file: Enables you to view all log entries since the file was created.
- **Show lines:** Sets the maximum number of message lines (log entries) shown. For example, if you specify 500 lines, the last 500 lines in the log file are shown.

Polling preferences include:

- **Server Ping Interval:** Sets the time interval (in milliseconds) for how often Inprise Application Servers are checked to verify they are active. The Disabled checkbox lets you disable server pinging.
- **Statistics Polling Interval:** Sets the time interval (in milliseconds) for how often objects are sampled for performance information. The Disabled checkbox lets you disable polling.

Server Discovery preferences include:

- Automatic Discovery: Enables the Location Service to automatically locate Inprise Application Servers on the network and display them in the Console.
- **Manual Discovery:** Enables you to manually add the servers you wish to display in the Console. Use this setting to selectively specify which servers appear in the Console.

<sup>開</sup> Professes	
General AppServer VisiBroker	
VisiBroker ORB	
Properties:	
Smart Agent Port: 6669	
	Ok Cancel
	UN Caller

Figure 2.7 VisiBroker preferences

VisiBroker preferences include:

• **Smart Agent Port:** Sets the port the Console uses to communicate with the Smart Agent.

You may have to restart the Console for changes in some preferences to take effect (for example, the Smart Agent port).

# Managing top-level services

Use the Inprise Application Server Console to view servers on the network and manage services associated with a server.

Content pane displays information about a service

Figure 2.8 Managing top-level services

	Navigation pane	or other item selected in the Navigation pa	ine
Click to view Application Servers on the network Server	Inprise Application Server Console         File Tools Help         Image: Servers         Image: Servers	or other item selected in the Navigation participation participation participation         CORBA Service 'osagentservice'         Description:         Smart Agent         Versioning:         Name:       osagentservice         Versioning:         Name:       osagentservice         Version:       4.0         Normal       :         E: Evedback	
	Done		6666

Menu commands you can use with servers and services include

 Table 2.1
 Managing top-level services

Command	Description
Start	Starts a stopped server or service.
Restart	Shuts down and restarts the server or service.
Stop   Normal	Exits the server or service normally. Writes data to disk as needed.
Stop   Kill	Terminates the sever or service immediately.
Ping	Checks to see if the server or service is active.
Edit Properties	Opens a dialog box with a list of server or service properties you can edit. (See "Editing service properties" on page 2-12 for more information.)

**Note** In addition to viewing and managing Inprise Application Servers, the Console enables you to manage VisiBroker services such as Gatekeeper and the Server Manager. (For more information, see the VisiBroker for Java documentation.)

#### Viewing log files

Log files provide a collection of error and status messages associated with a service. If problems occur, these files are useful for troubleshooting.

To view the log file for a service:

1 Use the Navigation pane to display the service you wish to examine. Then, open the folder for the service.

A list of folders and objects associated with the service appears.

- **2** Open the Logs folder for the service and select the log you wish to view.
  - Error log displays all error messages produced by the service.
  - Event log displays all events recorded by the service.

A list of log file properties appears in the Content pane.

**3** Click the View tab to view the log file.

#### Figure 2.9 Viewing log files

S Inprise Application Server Console		E	
P	Log 'event'		
P 🖉 Services	Property	Value	
Web Engines	log.event.description	"Logs startup/shutdown events"	
Inansaction Engines	log.event.destination	rolling_logfile	
<ul> <li>Build biology</li> <li>Build bi</li></ul>	log.event.file.filename	event_log	
🗢 📝 CORBA Services	log.event.formatclass	com.sun.server.log.TraceLog	
🕈 🛅 EJB Containers	log.event.level	2	
P ⊜√ ejbcontainer	log.event.level.0	None	
Y Clubs	log.event.level.1	Start or stop events only	
event	log.event.level.2	All events	
🖻 🐻 bank_beans.jar	log.event.level.3	All events, with detailed messages	
● ♥X northwest	log.event.level.max	3	
INDI Naming Service     INDI Naming Service	log.event.options		
	log.event.rolling_logfile.buffer	8192	
	log.event.rolling_logfile.filename	event_log	
	log.event.rolling_logfile.flushtime	0	
	log.event.rolling_logfile.rollover	1024000	
	log.event.serverNameEnabled	false	
	log.event.socket.host	localhost	
	log.event.socket.port	777	
	log.event.time	local	
	Pronerties View		
J J			leeco
pone			6666
Click to view log properties.	Cli	ck to view the actual log file.	

**Note** You can use Console preferences to control how many message lines are stored in the log files and how often the logs are updated.

### **Editing service properties**

Each service included with the Inprise Application Server has a set of properties you can edit to customize the service. The properties you can edit vary depending on the service.

To view and edit service properties:

- 1 Open the Application Server Console and select the service whose properties you wish to edit.
- 2 Choose Edit Properties from the Context menu.

Figure 2.10 Editing Service properties



A dialog box appears with a list of Service properties.

**3** Make changes as desired and click OK.

The following sections describes the properties you can edit for each service.

Figure 2.11 Examples of Service properties



#### Chapter

3

# **Using EJB Containers**

This chapter describes how to use the Inprise Application Server Console to view and manage EJB Containers and related services.

## About EJB Containers

The Inprise Application Server Console enables you to view and edit the contents of Containers and enterprise beans.

The Console includes an EJB Deployment wizard. During deployment, the contents of the EJB JAR file are verified and Inprise-specific information is added. The Deployment wizard:

- Creates the required stubs and skeletons for the enterprise bean
- Deploys the JAR to a Container
- Validates the deployment descriptor

After the enterprise bean is deployed, you can use the Deployment Descriptor Editor to change deployment information as desired. For example, you can use the editor to specify or change the Transaction Policies and Security Roles for the enterprise bean.

Stanprise Management Console	
File Tools Help	
• 5 4 4 5	
♥       Application Servers         ♥       ♥         ♥       Ø         ♥       Ø         ♥       Ø         Ø       Ø	EJB Container 'ejbcontainer'         Description:         Enterprise Java Bean Container
e av ejbcontainer	Versioning:
E JNDI Naming Service	Name: ejbcontainer
♥ III VisiBroker Services	Version: 4.0
	Root: D:UAS40
	User: Inprise Corp.
	Vendor: Inprise Corp.
	Group: Inprise Corp.
	Runtime Control: Status: Running
	Contents Version Configuration Properties Roles Performance Memory
	6665

Figure 3.1 EJB Container in the Inprise Application Server Console

#### **Persistence support**

Modeled after version 1.1 of the EJB specification published by Sun Microsystems, the Inprise EJB Container supports built-in persistence. This persistence can be bean-managed or container-managed.

For bean-managed persistence, the developer overrides the ejbLoad and ejbStore methods and provides custom code (for example, using JDBC) to transfer the data from the object's fields to permanent storage and vice versa. Bean-managed persistence ties the enterprise bean to one method of persistence and requires the developer to write and maintain persistence code.

For container-managed persistence, Inprise supplies mapping tools to map the Entity bean's field to fields in the JDataStore. The Entity bean can be persisted into a different data source by remapping the fields in the deployment descriptor.

For more information, see the Inprise Application Server *Enterprise JavaBeans Programmer's Guide*.

# **Viewing EJB Containers**

To view EJB Containers with the Inprise Application Server Console:

1 Start the Application Server and Console.

For details, see Chapter 2, "Starting the Inprise Application Server."

**2** Open the Services folder in the Navigation pane and select EJB Containers.

A list of EJB Containers and JAR files appears in the Navigation pane.

Figure 3.2 EJB Containers



The EJB server includes a default Container that appears in the Console window.

An EJB server can have multiple Containers. You define and configure Containers as needed to deploy objects on the network. You can create custom EJB Containers for various data sources (such as Oracle) and support them separately from enterprise bean development. (For more information, see "Creating a new Container" on page 3-20.)

## Viewing container attributes

Containers include a variety of attributes you can view and edit. These attributes include container contents, configuration information, and so on.

**Note** This section includes information about container attributes. For information about enterprise bean attributes, see "Viewing bean attributes" on page 3-10.

To view container attributes:

• Open the Navigation pane in the Inprise Application Server Console and select the EJB Container you wish to view.

Attributes for the Container appear in the Content pane.

Figure 3.3	Container attributes
------------	----------------------

🍪 Inprise Management Console		
File Tools Help		
• I 🗹 🧐 💱 🔂		
♥     Application Servers       ♥     ♥       ♥     Ø       Ø     Ø       Ø     Ø       Ø     Ø       Ø     Ø       Ø     Ø       Ø     Ø       Ø     Ø       Ø     Ø       Ø     Ø       Ø     Ø       Ø     Ø       Ø     Ø       Ø     Ø       Ø     Ø        Ø     Ø	EJB Container 'ejbcontainer' Conter Description: Enterprise Java Bean Container	nt
e a v ejbcontainer	Versioning:	
	Name: ejbcontainer	
In VisiBroker Services	Version: 4.0	
	Root: D:NAS40	
	User: Inprise Corp.	
	Vendor: Inprise Corp.	
	Group: Inprise Corp.	
	Runtime Control: Status: Running	
	Contents Version Configuration Properties Roles Performance Memory	
	6669	
Click to view Conta	iner attributes.	
Click to view perfo information for the	rmance Container.	

3-4 User's Guide

#### **Contents panel**

The Contents panel shows the JAR files included in the Container.





#### **Version panel**

The Version panel includes:

- General description of the Container
- Version number and root installation directory of the Inprise Application Server running the Container
- Container vendor
- Container runtime information



EJB	Contail	ner 'ejbconta	liner'		
Descript	ion:	an Cantainas			
Versioni					
Namo	ig.	ainar			
Voreion					
version:	4.0	0			
RUOT:	D.IIAS4	0			
User:	Inprise	Corp.			
Vendor:	Inprise	Corp.			
Group:	Inprise	Corp.			
Runtime Status:	Control: Running	(out of proces	s)		

### **Configuration and Properties panels**

The Configuration panel lists server configuration information defined for the Container. This information includes log file locations, JAR repository locations, service enabling attributes, and so on.

The Properties panel lists Java system properties for the Container. This information controls the Java Virtual Machine used to run the Container.

#### Figure 3.6 Configuration and Properties panels

admin.init args       nohost       images/Ms         admin.ul. Category. 1. config. Setup       images/Ms       images/Ms         admin.ul. Category. 2. monitor. Monitor       admin.ul. config.1. Properties       com.inpris         admin.ul. Category. 2. monitor. Monitor       admin.ul. config.1. Properties       com.inpris         admin.ul. Category. 2. monitor. Monitor       admin.ul. config.1. Properties       com.inpris         com.inpris       com.inpris       com.inpris         ejb. plars ejb. jars       ejb.container. description       None avai         ejb.container. tostaddress       17.2.0.20       ejbcontainer. hostaddress       17.2.0.20         ejb.container. tostaddress       12.20.20       ejbcontainer. tostaddress       17.2.0.20         ejbcontainer. javaproperties. OTSchecked_behavior       false       ejbcontainer. javaproperties. TransactionCheck       false         ejbcontainer. javaproperties. TransactionCheck       false       ejbcontainer. javaproperties. sawt toolkit       sun.awt.w         ejbcontainer. javaproperties. file.encoding.pkg       cum.inpris       ejbcontainer. javaproperties. java. awt.printeriob       sun.awt.W         ejbcontainer. javaproperties. java. awt.printeriob       sun.awt.w       ejbcontainer. javaproperties. java. awt.graphicserv       sun.awt.W         ejbcontainer. javaproperties. java. compiler       sym	Property		Value	
dmin.ui.Category 1: config.Setup       images/Setup         idmin.ui.Category 2: monitor Monitor       images/Setup         idmin.ui.config. 1: Properties       com.inpris         idmin.ui.config. 1: Properties       com.inpris         ijb.enable_jpda_debug       false         ijb.ars (bj.ars (b	idmin.init.args	nohost		
dmin.ul.category.2.monitor.Monitor       images/M         dmin.ul.config.1.Properties       com.sun.s         dmin.ul.config.1.Properties       com.sun.s         jb.jarrepository       ejb_jarse/ejb.ja	dmin.ui.Category.1.config.Setup	images/Se		_ [
dmin.ui.config.1.Properties       com.inpris         dmin.ui.config.1.Properties       com.sun.s         jb arasejota_debug       false         jb jars       jb jars         jb jars       jb jars         jb jars       ejbcontainer.builddate         jb jars       ejbcontainer.description         jb jars       ejbcontainer.hostaddress         jb jss       false         jb jss       true         jb jss       true         jb jss       true         jb jss.       true         jb jss.       true         jb jss.       true         jb security rolemap_path       system/se         js security rolemap_ath       system/se         gacess.description       false         gacess.description       "Logs pag         gacess.description       "Logs pag         gacess.selstination       rue         gacess.stevel       1 <tr< td=""><td>dmin.ui.Category.2.monitor.Monitor</td><td>images/M</td><td></td><td></td></tr<>	dmin.ui.Category.2.monitor.Monitor	images/M		
dmin.uimonitor.1.Log Output       com.sun.s         jb.enable_jpda_debug       false         jb.anable_jpda_debug       false         jb.anable_jpda_debug       false         jb.anable_jpda_debug       false         jb.anable_jpda_debug       false         jb.jars       gibcontainer.hostaddress       172.20.20         jb.jss       frue       gibcontainer.hostaddress       172.20.20         jb.jss       true       gibcontainer.hostaddress       172.20.20         jb.jss       true       gibcontainer.hostaddress       172.20.20         jb.jss       true       gibcontainer.javaproperties.ITSchecked_behavior       false         jb.sc.continer       gibcontainer.javaproperties.Sile.actionCheck       false         gibcontainer.javaproperties.sile.encoding       C2/VINNT         ejbcontainer.javaproperties.file.encoding       C2/VINNT         ejbcontainer.javaproperties.file.encoding       V         ga.access.description       "Logs pag         ga.access.file.filename       access_log       sun.avt.W         ga.access.file.filename       access_log       sun.avt.W         ga.access.file.filename       access_log       sun.avt.W         ga.access.file.filename       access_log       sun.avt.W <td>dmin.ui.config.1.Properties</td> <td>com.inpris</td> <td></td> <td></td>	dmin.ui.config.1.Properties	com.inpris		
jb enable_jpda_debug       false         jb jarr epository       ejb_ars(e)         jb jars (b) ars       ejb_container.builddate         jb jdb       true         jb jdb       true         jb jdb       ejbcontainer.builddate       Sep 17, 12         jb jdb       false         jb jdb       false         jb jbs       false         jb jbs       true         jb security.rolemap_path       system/se         yaserwice.losspath.appent       true         gbcontainer javaproperties.line encoding       CP1252         yaserwice.losspath.appent       true         gbcontainer javaproperties.line encoding       CP1252         ga access.description       "Logs pag         gg access.flemiliename       access_lo         gj access.flemiliename       access_lo	dmin.ui.monitor.1.Log Output	com.sun.s		
(b) Jarsepository       e)b_jars(e)*         (b) Jarse       (c) Jarse         (b) Jarse       (c) Jarse         (b) Jas       false         (c) Jass       (c) Jass         (c) Jass       false         (c) Jass       false         (c) Jass       false         (c) Jass       true         (c) Jass       tr	jb.enable_jpda_debug	false 👍	Distantia	
ib jars       bijordianer       bijordianer       sep 17., 12         jb jdb       true       bijordianer       sec 110, 12       sec 17., 12         jb jdb       true       ejbcontainer.description       None avai         jb jds       true       ejbcontainer.description       None avai         jb jds       true       ejbcontainer.description       None avai         jb jds       true       ejbcontainer.javaproperties.oTSchecked_behavior       false         jb stace_container       ejbcontainer.javaproperties.oTSchecked_behavior       false         ejbcontainer.javaproperties.savt toolkt       su awd.w         vaseervice.entrypoint       corm.inpris       ejbcontainer.javaproperties.file.encoding       C/WINNT         ejbcontainer.javaproperties.file.encoding.pkg       su n.io       ejbcontainer.javaproperties.jile.encoding.pkg       su n.io         ga access.description       "Logs pag       ejbcontainer.javaproperties.java.avd.graphicsenv       su awd.w         ga access.level       1       ejbcontainer.javaproperties.java.class.path       D/J(dk1.2.2         gibcontainer.javaproperties.java.class.version       46.0       ejbcontainer.javaproperties.java.class.version       46.0         gibcontainer.javaproperties.java.class.version       46.0       ejbcontainer.javaproperties.java.class.version	jb.jar.repository	ejb_jars/ej⊧	Property	Oan 47, 4000
ibidb       true       ipicontainer/bdschplund       Notie advalues         ibijsics       false       ipicontainer/bdschplund       Notie advalues         ibijsics       false       ipicontainer/bdschplund       plicontainer/bdschplund         ibijsics       false       ipicontainer/bdschplund       plitzen         ibijsics       true       ipicontainer/bdschplund       glitzen         ibijsics       true       ipicontainer/avaproperties.otTSchecked_behavior       false         ibijsics       true       ipicontainer/avaproperties.TransactionCheck       false         ibiontainer/avaproperties.file encoding       Cplicontainer/avaproperties.file encoding       Cplicontainer/avaproperties.file encoding       Cplicontainer/avaproperties.file encoding         vaserwice jors of the full of true       elibcontainer/avaproperties.file encoding       Cplicontainer/avaproperties.java.avd.fonts         og access flefilename       access.level       1       elibcontainer/avaproperties.java.avd.graphicsenv       sun.avd.W         og access flevel       1       elibcontainer/avaproperties.java.complete       sun.avd.W       elibcontainer/avaproperties.java.complete       sun.avd.W         og access flevel       1       elibcontainer/avaproperties.java.tome       D/J(dkl.2.2         og access flevel       1       elibcontainer/avaprope	jb.jars		ejboontainer.builduate	Sep 17, 1999
ib ins       false       ib jose       ib jose       gib container.hostanderss       gib container.hostanders       g	jb.jdb	true	ejbcontainer.description	NURE available
jb jsec     false     glucentianer       jb jss     true     glucentianer       jb jss     true     ejbcontainer javaproperties.OTSchecked_behavior     false       jb security.rolemap_path     system/se     ejbcontainer javaproperties.TransactionCheck     false       jb security.rolemap_path     system/se     ejbcontainer javaproperties.arw.toolkt     su awd.w       vaseervice.entrypoint     com.inpris     ejbcontainer javaproperties.env.windows.system_root     C:WINNT       vaseervice.entrypoint     com.inpris     ejbcontainer javaproperties.file.encoding     C:WINNT       vaseervice.entrypoint     com.inpris     ejbcontainer javaproperties.file.encoding     C:WINNT       vaseervice.entrypoint     com.inpris     ejbcontainer javaproperties.file.encoding     C:WINNT       vaseervice.entrypoint     rue     ejbcontainer javaproperties.file.encoding     Su aut.io       vaservice.entrypoint     rue     ejbcontainer javaproperties.java.awt.fonts     V       ejbcontainer javaproperties.java.awt.graphicsenv     su aut.awt.W     su awd.W       ga.access.fleefilename     access.jo     su awt.awt.w     su awd.prisenv     su awd.W       ga.access.fleefilename     access.jo     jbcontainer javaproperties.java.awt.graphicsenv     su awd.W       ga.access.fleefilename     access.jo     jbcontainer javaproperties.java.awt.graphicsenv <td>jb.jns</td> <td>false</td> <td>ejbcontainer.nostadoress</td> <td>172.20.20.197 alitzan</td>	jb.jns	false	ejbcontainer.nostadoress	172.20.20.197 alitzan
jb.jts       true       ejbcontainer javaproperties.OTSchecked_behavior       false         jb.security.rolemap_path       system/se       ejbcontainer javaproperties.TransactionCheck       false         jb.security.rolemap_path       system/se       ejbcontainer javaproperties.TransactionCheck       false         waservice.lsspath.append       true       ejbcontainer javaproperties.env windows system_root       C/WINNT         waservice.jvm.arguments       ejbcontainer javaproperties.file encoding       Cp1252         waservice.jvm.arguments       ejbcontainer javaproperties.java.avt.fonts       ejbcontainer javaproperties.java.avt.fonts         gja.access.file.filename       access.lo       ejbcontainer javaproperties.java.avt.fonts       ejbcontainer javaproperties.java.avt.fonts         gja.access.file.filename       access.lo       ejbcontainer javaproperties.java.avt.fonts       ejbcontainer javaproperties.java.avt.grash       U/Udt.2.2         gja.access.file.filename       access.lo       ejbcontainer javaproperties.java.avt.grash       U/Udt.2.4         gja.access.file.filename       access.lo       ejbcontainer javaproperties.java.toms       U/Udt.2.2         gja.access.file.file.nom       access.lo       ejbcontainer javaproperties.java.toms       D/Udt.2.2         gja.access.file.file.nom       access.lo       ejbcontainer javaproperties.java.toms       D/Udt.2.2	jb.jsec	false	ejboontainer.nostname	gilizen
jb is       true       ejbcontainer javaproperties. Transaction.Checke       false         jb.security.rolemap_path       system/se       ejbcontainer javaproperties. Transaction.Checke       false         ejbcontainer javaproperties.Transaction.Checke       ejbcontainer javaproperties. Transaction.Checke       false         ejbcontainer javaproperties.Transaction.Checke       ejbcontainer javaproperties.Transaction.Checke       sun.awd.w         vasaervice.iog.stdout       true       ejbcontainer javaproperties.Tile.encoding       CP1252         ejbcontainer javaproperties.Tile.encoding.pkg       sun.io         ejbcontainer javaproperties.Jile.esparator       t         ejbcontainer javaproperties.java.awt.graphicsenv       sun.awt.W         ejbcontainer javaproperties.java.awt.graphicsenv       sun.awt.W         ejbcontainer javaproperties.java.awt.graphicsenv       sun.awt.W         ejbcontainer javaproperties.java.awt.graphicsenv       sun.awt.W         ejbcontainer javaproperties.java.ctass.path       D/Jdkl.2.2         ejbcontainer javaproperties.java.ctass.path       D/Jdkl.2.2         ejbcontainer javaproperties.java.ctass.version       46.0         ejbcontainer javaproperties.java.it.trans       D/Jdkl.2.2         ejbcontainer javaproperties.java.it.transproperties.java.it.transproperties.java.it.transproperties.java.it.transproperties.java.it.transproperties.java.it.transproperties.java	jb.jss	true	ejocuritainer.jars	felee
jo security rolemap_path system/se jo brace_ontainer waservice.classpath.append true waservice.classpath.append true waservice.classpath.append true waservice.classpath.append true waservice.classpath.append true waservice.log.stdout ga.ccess.description gb.container javaproperties java.avt.dirs gb.container javaproperties java.dv.dr gb.container javaproperties java.dv.dr gb.container javaproperties java.dv.dr gb.container javaproperties java.dv.dr gb.container javaproperties java.specific.ation.name deb.container javaproperties java.specific.ation.name deb.container javaproperties java.specific.ation.name deb.container javaproperties java.specific.ation.name deb.container javaproperties java.specific.ation.name deb.container javaproperties java.specific.ation.name deb.container javaproperties j	jb.jts	true	ejbcontainer.javaproperties.Or Schecked_behavior	false
jb.trace_container       false       bit.container javaproperties.awi.totikit       Sult.awi.k         waservice.etrypoint       com.inpris       ejbcontainer javaproperties.awi.uotikit       Sult.awi.k         waservice.intrypoint       com.inpris       ejbcontainer javaproperties.awi.uotikit       Sult.awi.k         waservice.intrypoint       com.inpris       ejbcontainer javaproperties.file.encoding.pkg       sun.io         gaccess.description       "Logs pag       ejbcontainer javaproperties.file.eparator       \\         gaccess.description       "Logs pag       ejbcontainer javaproperties.java.awt.graphicsenv       sun.awt.w         gaccess.file.filename       access_lo       ejbcontainer javaproperties.java.awt.graphicsenv       sun.awt.w         gaccess.file.filename       access_lo       ejbcontainer javaproperties.java.class.path       D\ldk1.2.2         gaccess.level       1       ejbcontainer javaproperties.java.class.path       D\ldk1.2.2         ejbcontainer javaproperties.java.class.version       46.0       ejbcontainer javaproperties.java.class.path       D\ldk1.2.2         ejbcontainer javaproperties.java.class.version       46.0       ejbcontainer javaproperties.java.class.path       D\ldk1.2.2         ejbcontainer javaproperties.java.library.path       D\ldk1.2.2       ejbcontainer javaproperties.java.library.path       D\ldk1.2.2 <td< td=""><td>jb.security.rolemap_path</td><td>system/se</td><td>ejocontainer javaproperties, mansaction check</td><td>oup out window</td></td<>	jb.security.rolemap_path	system/se	ejocontainer javaproperties, mansaction check	oup out window
waservice.classpath.append       true       epicontainer javaproperties file encoding pkg       c.w.minto         waservice.ing.stdout       true       epicontainer javaproperties file encoding pkg       sun.io         waservice.log.stdout       true       epicontainer javaproperties file encoding pkg       sun.io         og.access.description       "Logs pag       epicontainer javaproperties file encoding pkg       sun.io         og.access.description       "Logs pag       epicontainer javaproperties java.awt.fonts       epicontainer javaproperties java.awt.fonts         og.access.fle.filename       access_lo       epicontainer javaproperties java.awt.graphicsenv       sun.awt.W         og.access.flevel       1       epicontainer javaproperties java.compiler       sun.awt.W         og.access.level       1       epicontainer javaproperties java.compiler       synchia         og.access.level       1       epicontainer javaproperties java.compiler       D/ldkl.2.2         opicontainer javaproperties java.omming.factory.initial       D/ldkl.2.2       epicontainer javaproperties java.anming.factory.initial       com.inpris         opicontainer javaproperties java.anming.factory.initial       com.inpris       epicontainer javaproperties java.anming.factory.initial       com.inpris	jb.trace_container	false	ejocontainer.javaproperties.awi.tookit	C-WAININT
avaservice, invasirice entrypoint     corm.inpris       avaservice, invasirice and structure     epicontainer javaproperties file expansion       avaservice, log, stdout     true       ga access, description     "Logs pag       ga access, description     "Logs pag       ga access, description     access, log       ga access, formatclass     corm.sun.s       ga access, level     1       ag access, level     1       ag access, level     numetrice       ag access, level     numetris       ag access, level     <	avaservice.classpath.append	true	ejocontainer.javaproperties.env.windows.system_toot	C:000000
avaservice.jvm.arguments       epicontainer javaproperties jie separator       k         avaservice.jvm.arguments       epicontainer javaproperties jie separator       k         gaccess.description       "Logs pag       epicontainer javaproperties jie separator       k         ggaccess.description       "Logs pag       epicontainer javaproperties jiava.awt.graphicsenv       sun.awt.W         ggaccess.description       rolling_log       epicontainer javaproperties jiava.awt.graphicsenv       sun.awt.W         ggaccess.fevel       1       epicontainer javaproperties java.awt.graphicsenv       sun.awt.W         ggaccess.fevel       1       epicontainer javaproperties java.class.path       D\idk1.2.2         Contents       Version       Configuration       Properties         ejocontainer javaproperties java.home       D\idk1.2.2       ejocontainer javaproperties java.class.path       D\idk1.2.2         ejocontainer javaproperties java.home       D\idk1.2.2       ejocontainer javaproperties java.tome       D\idk1.2.2         ejocontainer javaproperties java.home       D\idk1.2.2       ejocontainer javaproperties java.tompiler       Symcjitt         ejocontainer javaproperties java.noming.factory.initial       com.inpris       com.inpris       com.inpris         ejocontainer javaproperties java.specification.name       Java Platfor       E       E	avaservice.entrypoint	com.inpris	ejocontainer javaproperties file encoding n/g	cunio
avaservice log stdout true og.access.description "Logs pag og.access.description "Olling_log og.access.destination rolling_log og.access.destination rolling_log og.access.file.filename access_lo og.access.level 1 og.access.level 1 Mone Contents Version Configuration Properties identified access.gene to the state of	avaservice.jvm.arguments		ejocontainer javaproperties ille cenorator	Sunno N
og access description "Logs pag og access destination rolling.Jog og access destination rolling.Jog og access destination rolling.Jog og access formatclass corm.sun.s og access level 1 og access level 1 None elbcontainer javaproperties java.awt graphicsenv sun.awt.W elbcontainer javaproperties java.class.path D\Jdk1.2.2 elbcontainer javaproperties java.class.version 46.0 elbcontainer javaproperties java.class.version 46.0 elbcontainer javaproperties java.class.version 46.0 elbcontainer javaproperties java.compiler symcit elbcontainer javaproperties java.compiler D\Jdk1.2.2 elbcontainer javaproperties java.lot.moder D\Jdk1.2.2 elbcontainer javaproperties java.secofication.name Java Platfor	avaservice.log.stdout	true	ejocontainer javaproperties inte separator	,
og access destination rolling_log og access destination rolling_log og access file.filename access_log og access file.filename access file.filename access file.filename access og access file.filename access file.file.filename access file.file.file.file.file.file.file.file.	og.access.description	"Logs pag	eibcontainer javaproperties java awt granhicsenv	sun swit Min 32(
ga access file filename access_lo og access formatclass com.sun.s og access level 1 ga access level 0 Contents Version Configuration Properties eibcontainer javaproperties java.class path D\ldk1.2.2 eibcontainer javaproperties java.class version 46.0 eibcontainer javaproperties java.class version 90/(dk1.2.2 eibcontainer javaproperties java.dlars D.ldk1.2.2 eibcontainer javaproperties java.dlars D.ldk1.2.2 eibcontainer javaproperties java.compiler D.ldk1.2.2 eibcontainer javaproperties java.lot trapdir C.1TEMPU eibcontainer javaproperties java.anaming.factory.initial com.inpris eibcontainer javaproperties java.anaming.factory.url.pkgs com.inpris	og.access.destination	rolling_log	eibcontainer javaproperties java awt printerich	sun awtwindow
ga access formatclass       corm.sun.s       0 portinition (parpoperties) jovacitors synchronic (parpoperties)	og.access.file.filename	access_lo	eibcontainer javaproperties java class nath	Driidk1 2 21libite
ga access level 1 bg access level 1 bg access level 0 bg access level 0 Contents Version Configuration Properties version Configuration	og.access.formatclass	com.sun.s	eibrontainer javanronerties java class version	46.0
og access level. U None optionaring inder option	og.access.level	1 80	eibcontainer javaproperties java compiler	sympit
Contents Version Configuration Properties e)bcontainer javaproperties java. hormé D.\ldk1.2.2 e)bcontainer javaproperties java. hormé D.\ldk1.2.2 e)bcontainer javaproperties java. ibt normé D.\ldk1.2.2 e)bcontainer jav	og.access.level.0	None	eibcontainer javaproperties java evt dirs	Driidk1 2 2iireili
Contents Version Configuration Properties ejbcontainer javaproperties java.io.tmpdir CATEMPV ejbcontainer javaproperties java.io.tmpdir CATEMPV ejbcontainer javaproperties java.naming.factory.initial com.inpris ejbcontainer javaproperties java.axing.factory.url.pkgs com.inpris ejbcontainer javaproperties java.axing.factory.url.pkgs com.inpris			eibcontainer javaproperties java home	Drijdk1 2 2)ire
ejbcontainer javaproperties java. Ibirary path D1/dk1.2.2 ejbcontainer javaproperties java. naming factory.initial com.inpris ejbcontainer javaproperties java. naming factory. url.pkgs com.inpris ejbcontainer javaproperties java. specification. name Java Platfor	Contents Version Configuration	Properties	eibcontainer javaproperties java jo tmpdir	C:\TEMP\
ejbcontainer javaproperties java. naming factory. Initial com. Inpris ejbcontainer javaproperties java. naming factory. url. pkgs com. Inpris ejbcontainer javaproperties java. specification. name Java Platfo	Sector (Sector )		eibcontainer, javaproperties, java, library, path	D:\idk1.2.2\ire\b
ejbcontainer javaproperties java.naming factory url.pkgs com.inpris ejbcontainer javaproperties java.specification.name Java Platfo			ejbcontainer, javaproperties, java, naming, factory, initial	com.inprise.eib
ejbcontainer javaproperties java.specification.name Java Platfo		1000	ejbcontainer, javaproperties, java, naming, factory, url, pkgs	com.inprise.ejb
			ejbcontainer javaproperties java specification name	Java Platform A
2. I there are a second s				•

#### **Roles panel**

The Roles panel displays the contents of the Roles DB file defined for the Container. You can use the Roles panel to create a new Roles DB file or edit the contents of an existing file. For more information about this file, see the Inprise Application Server *Security Service Guide*.

To edit the file:

- 1 Open the Roles panel, and click to set an insertion point.
- 2 Enter new information as desired.
- **3** Click Save to save the changes.





# Viewing performance information

In addition to container attributes, you can use the Console to view performance information about Containers. Use the Performance and Memory tabs in the Content pane to view this information.

Performance information is useful for monitoring the amount of time the Container spends on deployment activities such as Dispatching or Activation. Use this information to troubleshoot network performance problems (for example, when a Container spends a much larger percentage of time on dispatching instead of business logic).

You can view performance information in a variety of formats including: bar charts, line charts, 3D charts, and numerical tables.





Use the Memory tab to display memory used by the Container. Memory usage includes:

	-
Activity	Description
Current used	Memory used at a given point in time.
Total used	Total memory available at a given point in time.
Maximum used	Maximum amount of memory used by the Container since it started.
Maximum total	Maximum amount of total memory available since the Container started.

 Table 3.1
 Memory usage information





# Viewing bean attributes

The Inprise EJB Container supports three types of enterprise beans:

- **Stateless Session beans:** In these beans, any state (if required) is maintained by the client or in an external location such as a database. Because no state is maintained, stateless Session beans aren't tied to any specific client; any available instance of a stateless Session bean can be used to service a client.
- **Stateful Session beans:** In these beans, state is maintained by the enterprise bean. This means the application server manages client-bean pairs. Each instance of the enterprise bean is created on behalf of a client and is intended to be a private resource to that client. Stateful Session beans can access persistent resources (such

as databases and files) but unlike Entity beans, they don't actually represent the data.

• Entity beans: These beans are persistent objects and represent an object view of data stored in permanent storage. An Entity bean lives in an EJB Container in much the same way a record lives in a database. Unlike Stateful Session beans, Entity beans can be accessed by multiple clients concurrently. This concurrency is managed by the EJB Container.

Beans include a variety of attributes you can view. These attributes include bean identity information, environment settings, and so on.

**Note** This section includes information about bean attributes. For information about container attributes, see "Viewing container attributes" on page 3-3.

To view bean attributes:

• Open the Navigation pane in the Inprise Application Server Console and select the enterprise bean you wish to view.

Content pane

Attributes for the enterprise bean appear in the Content pane.

Figure 3.10 Bean attributes

<ul> <li></li></ul>	Entity Bean ' Description: This bean c General: Bean Name:	container* orresponds to the Container entity. container	
♥ () pigs_beans.jar ♥ (Contained ♥ () contained ♥ () pert_beans.jar ♥ () sort_beans.jar ♥ () sort_beans.jar ♥ () mortgage.jar	Home Interface: Remote Interface: Bean Class: Home Name:	ContainerHome Container ContainerBean pigs/Container	
E JNUINaming service ▶ @ VisiBroker Services	Type: Primary Key Class: Reentrant:	Bean Managed Persistence	
	Properties Configura	ation Persistence Methods Bean States	66

**Note** Not all beans display all the attributes specified. For example, the Persistence property only appears if you have selected an Entity bean.

#### **Properties panel**

The Properties panel displays baseline information about the enterprise bean. This includes the bean's class name and the class names for the bean's home and remote interface.

For Entity beans, it also indicates whether the enterprise bean is reentrant. A reentrant bean can accept more than one call in the same transaction context. (Inprise recommends you avoid making a bean reentrant.)

For Stateful Session beans, timeout information is also provided.

💙 Entity Bean '	savings'	
Description:		
This Entity b	ean is an example of Bean Managed Persistence	
General:		
Bean Name:	savings	
Home Interface:	AccountHome	
Remote Interface:	Account	
Bean Class:	SavingsAccount	
Home Name:	accounts/savings	
Persistence/Sessio	n:	
Type:	Bean Managed Persistence	Entity beans on
Primary Key Class:	AccountPK	
Reentrant:	×	

## **Configuration panel**

The Configuration panel lists Inprise-specific environment and configuration information defined for the enterprise bean. This information includes persistence storage information, database location and login information, and so on.

#### Figure 3.12 Configuration information

C Entity Bean container	
Property	Value
contained-home ejb.cmp.jdbc.password ejb.cmp.jdbc.primaryKeyFieldName	pigs/Contained tiger name
ejb.cmp.jdbc.table ejb.cmp.jdbc.url	Container jdbc:inprise:its_direct:borland:dslocal:

#### **Persistence** panel

This panel appears for Entity beans only. It does not appear for Session beans.

For container-managed persistence, the panel displays:

- Names of database fields managed by the Container
- Class name of the enterprise bean's primary key

For Entity EJBs, a primary key is an object of user-defined type that can be used to look up a reference to the enterprise bean.

For bean-managed persistence, the panel displays the bean's primary key.

#### Figure 3.13 Persistence information

Entity Bean 'savings'	
Persistence:	
Type: Bean	
Primary Key Class Name: AccountPK	
	Persistence panel only appears for
	Entity beans
Properties Configuration Persistence Methods Bean States	

# Methods panel

The Methods panel lists all the methods included in the enterprise bean and referenced by the bean. It also displays the Transaction Policies assigned to each method.



Transaction De	marcation Type: O Bean	
Methods:		
Transaction P	olicy Signatu	е
Required	void transfer(Account, Account, float)	-

### State panel

The State panel displays an overview of the enterprise bean's life cycle and the proportion of time a bean spends in each phase. For example, the State panel shows how much time a bean spends on invocations of its business methods vs. other tasks. This information is useful for troubleshooting potential network and performance problems.

You can view State information as a pie chart or line chart. To help you interpret the information provided, state diagrams of the bean's life cycle are provided for each type of bean: Stateless Session, Stateful Session, and Entity.

For more information on EJB state diagrams and life cycle, see the Inprise Application Server *Enterprise JavaBeans Programmer's Guide*.





# **Deploying EBJs**

Deploying an enterprise bean means installing the EJB JAR file in a Container. Installation includes:

- Merging support files that make up the enterprise bean
- Registration of the enterprise bean with a naming service
- Executing the Transaction Policies and Security Roles
- Adding Inprise-specific information

You can install any number of beans in a Container.

You can use the Deployment wizard included with the Inprise Application Server Console to deploy the enterprise bean, or you can use commands in the EJB toolkit. The following section describes how to use the Deployment wizard to deploy beans. For more information on the EJB toolkit, see the Inprise Application Server *Enterprise JavaBeans Programmer's Guide*.

To deploy an EJB:

1 Open the Application Server Console and choose Deployment Wizard from the Tools menu.

The EJB Deployment wizard starts.

Figure 3.16 Deploying an EJB JAR file



- **2** Enter the name of the JAR file that contains the beans you want to deploy. You can type in the name or use the Browse button to locate the JAR file.
- **3** Select the Container you want to deploy the enterprise bean to.

The Container pop-up menu includes a list of Containers that are running.

4 Click the Deploy JAR to Container button and follow the onscreen instructions.

During deployment, the contents of the JAR file is verified and Inprise-specific information is added. The Deployment wizard:

- Creates the required stubs and skeletons for the enterprise bean
- Deploys the JAR to a Container
- Validates the deployment descriptor

When the enterprise bean is deployed successfully, the JAR file appears in the Container folder.





After the enterprise bean is deployed, you can use the Deployment Descriptor Editor to change deployment information as desired. For example, you can use the Deployment Descriptor Editor to specify or change the Transaction Policies and Security Roles for the enterprise bean. For more information, see Chapter 4.

**Note** If errors are found during deployment, the Deployment Descriptor Editor opens automatically to enable you to correct problems.

## Converting EJB 1.0 to EJB 1.1

The Deployment wizard converts an EJB 1.0 serialized deployment descriptor (or a set of descriptors) to an EJB 1.1 XML deployment descriptor. This wizard can be used to migrate enterprise beans from other EJB vendors to the Inprise EJB Container.

The wizard generates the DTD embedded within the XML deployment descriptor, which means that the Inprise Deployment Descriptor Editor (or any standard XML editor) can edit the generated XML deployment descriptor.

See the Inprise Application Server *Enterprise JavaBeans Programmer's Guide* for more information about changing EJB 1.0 deployment descriptors to EJB 1.1 deployment descriptors.

# **Removing EJBs from a Container**

To remove an EJB JAR file from a Container:

- 1 Open the Container folder in the Console.
- 2 Select the JAR file you wish to delete and choose Remove from the Context menu.

Figure 3.18 Removing a JAR file



The JAR file is removed from the Container.

# **Restarting a Container**

Occasionally, you may need to restart the Container for changes to take effect (for example, to make sure changes to the EJB JAR file are updated on the server).

To restart a Container:

- 1 Locate the Container in the Navigation pane.
- 2 Right-click the Container and choose Restart from the Context menu.

Figure 3.19 Restarting the Container



After a few moments, the Container restarts. When the Container has completely restarted, a check mark appears next to the Container.

# **Creating a new Container**

If desired, you can create additional versions of the Inprise Application Server's EJB Container and run these Containers out of process.

To create a new Container:

1 Using a file system navigation tool, locate the Server properties directory for the Inprise Application Server

install\_dir/properties/server/server\_name/

where *install\_dir* is the Inprise Application Server installation directory and *server\_name* is the name of the server.

**2** Copy the ebjcontainer directory and give the directory a new name. (This is the name of the new Container.)

🔍 Exploring - ejbcontainer			_ 🗆 ×
<u>File E</u> dit ⊻iew <u>I</u> ools <u>H</u> elp			
🕞 ejbcontainer 🔹 🗈 🍋 🍋 🛣 👗 🗈 🛱			
All Folders	Contents of 'eibcontainer'		
Binny (A:)	Name	Size Tupe	Modified 4
	adminUI.properties	1KB PROPERTIES File	7/7/99 4:08 PM
🕒 🕞 (D:)	(a) ejb.properties	1KB PROPERTIES File	9/17/99 10:45 AM
AppServer1	javaservice.properties	1KB PROPERTIES File	9/24/99 6:59 PM
	service.properties	1KB PROPERTIES File	9/17/99 8:41 AM
ti dm	systemDefaults.properties	5KB PROPERTIES File	7/16/99 10:49 AM
te- Cin doc			
examples			
i → intml			
E - I logs			
i properties		-	
i ⊡ alitzen	To create a new		
- adminservice	Container conv		
ejbcontainer			
itsservice	and rename this		
idatastoreservice	directory		
namingservice			
		-	
sessionservice			
webpageservice			
ervice_defaults	1		
i realms	1		
servietbeans	1		
serviets	1		
The system system			
5 object(s) 6.87KB (Disk free space: 746MB)	,		

Figure 3.20 Container properties folder

- **3** In the directory you just created, locate the server.properties file and use a text editor to change the following:
  - Change the service.name entry to the name of the new Container.
  - Change the service.description entry to a description of the new Container.

The following is a partial example of a server.properties file.

```
#Service information
service.name=mycontainer
service.description=Custom Container for EJBs
service.vendor=Inprise Corp.
service.version=4.0
...
```

- 4 Locate the administer\_services.properties file (*install\_dir*/properties/server/ *server\_name*/administer\_services.properties) and add the new container name to the server.service.administer entry.
- **5** Restart the server. Then, use the Console (or Web Administration tool) to start the new Container.
- **Note** After you create a new Container, you can use the Console to change container attributes, if desired. See "Viewing container attributes" on page 3-3.