

◆ ◆ ◆ **4**
CHAPTER 4

Administering GlassFish Server Instances

A GlassFish Server *instance* is a single Virtual Machine for the Java platform (Java Virtual Machine or JVM machine) on a single node in which GlassFish Server is running. A node defines the host where the GlassFish Server instance resides. The JVM machine must be compatible with the Java Platform, Enterprise Edition (Java EE).

GlassFish Server instances form the basis of an application deployment. An instance is a building block in the clustering, load balancing, and session persistence features of GlassFish Server. Each instance belongs to a single domain and has its own directory structure, configuration, and deployed applications. Every instance must contain a reference to a node that defines the host where the instance resides.

The following topics are addressed here:

- [“Types of GlassFish Server Instances” on page 47](#)
- [“Administering GlassFish Server Instances Centrally” on page 48](#)
- [“Administering GlassFish Server Instances Locally” on page 58](#)
- [“Resynchronizing GlassFish Server Instances and the DAS” on page 66](#)
- [“Migrating EJB Timers” on page 66](#)

Types of GlassFish Server Instances

Each GlassFish Server instance is one of the following types of instance:

Standalone instance

A standalone instance does not share its configuration with any other instances or clusters. A standalone instance is created if either of the following conditions is met:

- No configuration or cluster is specified in the command to create the instance.
- A configuration that is not referenced by any other instances or clusters is specified in the command to create the instance.

When no configuration or cluster is specified, a copy of the `default-config` configuration is created for the instance. The name of this configuration is *instance-name-config*, where *instance-name* represents the name of an unclustered server instance.

Shared instance

A shared instance shares its configuration with other instances or clusters. A shared instance is created if a configuration that is referenced by other instances or clusters is specified in the command to create the instance.

Clustered instance

A clustered instance inherits its configuration from the cluster to which the instance belongs and shares its configuration with other instances in the cluster. A clustered instance is created if a cluster is specified in the command to create the instance.

Any instance that is not part of a cluster is considered an unclustered server instance. Therefore, standalone instances and shared instances are unclustered server instances.

Administering GlassFish Server Instances Centrally

Centralized administration requires secure shell (SSH) to be set up. If SSH is set up, you can administer clustered instances without the need to log in to hosts where remote instances reside.

Administering GlassFish Server instances centrally involves the following tasks:

- [“To Create an Instance Centrally” on page 48](#)
- [“To List All Instances in a Domain” on page 51](#)
- [“To Delete an Instance Centrally” on page 52](#)
- [“To Start a Cluster” on page 53](#)
- [“To Stop a Cluster” on page 54](#)
- [“To Start an Individual Instance Centrally” on page 55](#)
- [“To Stop an Individual Instance Centrally” on page 56](#)
- [“To Restart an Individual Instance Centrally” on page 57](#)

▼ To Create an Instance Centrally

Use the `create-instance` subcommand in remote mode to create a GlassFish Server instance centrally.

If the instance is managed by GMS, system properties for the instance that relate to GMS must be configured correctly. To avoid the need to restart the DAS and the instance, configure an instance's system properties that relate to GMS when you create the instance. If you change GMS-related system properties for an existing instance, the DAS and the instance must be restarted to apply the changes.

Before You Begin Ensure that following prerequisites are met:

- The node where the instance is to reside exists and is enabled for remote communication. For more information, see [“To Create an SSH Node” on page 30](#).
- The user of the DAS can use SSH to log in to the host for the node where the instance is to reside.

If any of these prerequisites is not met, create the instance locally as explained in [“To Create an Instance Locally” on page 58](#).

If you are adding the instance to a cluster, ensure that the cluster to which you are adding the instance exists. For information about how to create a cluster, see [“To Create a Cluster” on page 40](#).

If the instance is to reference an existing named configuration, ensure that the configuration exists. For more information, see [“To Create a Named Configuration” on page 71](#).

The instance might be managed by GMS and reside on a node that represents a multihome host. In this situation, ensure that you have the Internet Protocol (IP) address of the network interface to which GMS binds.

1 Ensure that the DAS is running.

Remote subcommands require a running server.

2 Run the `create-instance` subcommand.

Note – Only the options that are required to complete this task are provided in this step. For information about all the options for configuring the instance, see the `create-instance(1)` help page.

- **If you are creating a standalone instance, do not specify a cluster.**

If the instance is to reference an existing configuration, specify a configuration that no other cluster or instance references.

```
asadmin> create-instance --node node-name
[--config configuration-name] instance-name
```

node-name

The node on which the instance is to reside.

configuration-name

The name of the existing named configuration that the instance will reference.

If you do not require the instance to reference an existing configuration, omit this option. A copy of the `default-config` configuration is created for the instance. The name of this configuration is `instance-name-config`, where `instance-name` is the name of the server instance.

instance-name

Your choice of name for the instance that you are creating.

- **If you are creating a shared instance, specify the configuration that the instance will share with other clusters or instances.**

Do *not* specify a cluster.

```
asadmin> create-instance --node node-name
--config configuration-name instance-name
```

node-name

The node on which the instance is to reside.

configuration-name

The name of the existing named configuration that the instance will reference.

instance-name

Your choice of name for the instance that you are creating.

- **If you are creating a clustered instance, specify the cluster to which the instance will belong.**

If the instance is managed by GMS and resides on a node that represents a multihome host, specify the GMS-BIND-INTERFACE-ADDRESS-*cluster-name* system property.

```
asadmin> create-instance --cluster cluster-name --node node-name
[--systemproperties GMS-BIND-INTERFACE-ADDRESS-cluster-name=bind-address] instance-name
```

cluster-name

The name of the cluster to which you are adding the instance.

node-name

The node on which the instance is to reside.

bind-address

The IP address of the network interface to which GMS binds. Specify this option only if the instance is managed by GMS and resides on a node that represents a multihome host.

instance-name

Your choice of name for the instance that you are creating.

Example 4-1 Creating a Clustered Instance Centrally

This example adds the instance `pmd-i1` to the cluster `pmdclust` in the domain `domain1`. The instance resides on the node `sj01`, which represents the host `sj01.example.com`.

```
asadmin> create-instance --cluster pmdclust --node sj01 pmd-i1
Port Assignments for server instance pmd-i1:
JMX_SYSTEM_CONNECTOR_PORT=28686
JMS_PROVIDER_PORT=27676
HTTP_LISTENER_PORT=28080
ASADMIN_LISTENER_PORT=24848
IIOP_SSL_LISTENER_PORT=23820
```

```

IIOP_LISTENER_PORT=23700
HTTP_SSL_LISTENER_PORT=28181
IIOP_SSL_MUTUALAUTH_PORT=23920
The instance, pmd-i1, was created on host sj01.example.com
Command create-instance executed successfully.

```

- See Also**
- [“To Create an SSH Node” on page 30](#)
 - [“To Create an Instance Locally” on page 58](#)
 - `create-instance(1)`

You can also view the full syntax and options of the subcommand by typing `asadmin help create-instance` at the command line.

- Next Steps** After creating an instance, you can start the instance as explained in the following sections:
- [“To Start an Individual Instance Centrally” on page 55](#)
 - [“To Stop an Individual Instance Locally” on page 64](#)

▼ To List All Instances in a Domain

Use the `list-instances` subcommand in remote mode to obtain information about existing instances in a domain.

1 Ensure that the DAS is running.

Remote subcommands require a running server.

2 Run the `list-instances(1)` subcommand.

```
asadmin> list-instances
```

Example 4-2 Listing Basic Information About All GlassFish Server Instances in a Domain

This example lists the name and status of all GlassFish Server instances in the current domain.

```

asadmin> list-instances
pmd-i2 running
yml-i2 running
pmd-i1 running
yml-i1 running
pmdsa1 not running
Command list-instances executed successfully.

```

Example 4-3 Listing Detailed Information About All GlassFish Server Instances in a Domain

This example lists detailed information about all GlassFish Server instances in the current domain.

```

asadmin> list-instances --long=true
NAME      HOST                PORT  PID    CLUSTER  STATE
pmd-i1    sj01.example.com    24848 31310  pmdcluster  running
yml-i1    sj01.example.com    24849 25355  ymlcluster  running
pmdsa1    localhost           24848 -1     ---        not running
pmd-i2    sj02.example.com    24848 22498  pmdcluster  running
yml-i2    sj02.example.com    24849 20476  ymlcluster  running
ymlsa1    localhost           24849 -1     ---        not running
Command list-instances executed successfully.

```

See Also `list-instances(1)`

You can also view the full syntax and options of the subcommand by typing `asadmin help list-instances` at the command line.

▼ To Delete an Instance Centrally

Use the `delete-instance` subcommand in remote mode to delete a GlassFish Server instance centrally.



Caution – If you are using a Java Message Service (JMS) cluster with a master broker, do not delete the instance that is associated with the master broker. If this instance must be deleted, use the `change-master-broker(1)` subcommand to assign the master broker to a different instance.

Deleting an instance involves the following:

- Removing the instance from the configuration of the DAS
- Deleting the instance's files from file system

Before You Begin Ensure that the instance that you are deleting is not running. For information about how to stop an instance, see the following sections:

- [“To Stop an Individual Instance Centrally” on page 56](#)
- [“To Stop an Individual Instance Locally” on page 64](#)

1 Ensure that the DAS is running.

Remote subcommands require a running server.

2 Confirm that the instance is not running.

```
asadmin> list-instances instance-name
```

```
instance-name
```

The name of the instance that you are deleting.

3 Run the `delete-instance(1)` subcommand.

```
asadmin> delete-instance instance-name
```

instance-name

The name of the instance that you are deleting.

Example 4-4 Deleting an Instance Centrally

This example confirms that the instance `pmd - i1` is not running and deletes the instance.

```
asadmin> list-instances pmd-i1
pmd-i1  not running
Command list-instances executed successfully.
asadmin> delete-instance pmd-i1
Command _delete-instance-filesystem executed successfully.
The instance, pmd-i1, was deleted from host sj01.example.com
Command delete-instance executed successfully.
```

- See Also**
- [“To Stop an Individual Instance Centrally” on page 56](#)
 - [“To Stop an Individual Instance Locally” on page 64](#)
 - `change-master-broker(1)`
 - `delete-instance(1)`
 - `list-instances(1)`

You can also view the full syntax and options of the subcommands by typing the following commands at the command line:

- `asadmin help delete-instance`
- `asadmin help list-instances`

▼ To Start a Cluster

Use the `start-cluster` subcommand in remote mode to start a cluster.

Starting a cluster starts all instances in the cluster that are not already running.

Before You Begin Ensure that following prerequisites are met:

- All nodes where instances in the cluster reside are enabled for remote communication.
- The user of the DAS can use secure shell (SSH) to log in to the host for any node where instances in the cluster reside.

If any of these prerequisites is not met, start the cluster by starting each instance locally as explained in [“To Start an Individual Instance Locally” on page 63](#).

1 Ensure that the DAS is running.

Remote subcommands require a running server.

2 Run the `start-cluster(1)` subcommand.

```
asadmin> start-cluster cluster-name
```

cluster-name

The name of the cluster that you are starting.

Example 4-5 Starting a Cluster

This example starts the cluster `pmcluster`.

```
asadmin> start-cluster pmcluster
Command start-cluster executed successfully.
```

- See Also**
- [“To Start an Individual Instance Locally” on page 63](#)
 - `start-cluster(1)`

You can also view the full syntax and options of the subcommand by typing `asadmin help start-cluster` at the command line.

Next Steps After starting a cluster, you can deploy applications to the cluster. For more information, see *GlassFish Server Open Source Edition 3.1 Application Deployment Guide*.

▼ To Stop a Cluster

Use the `stop-cluster` subcommand in remote mode to stop a cluster.

Stopping a cluster stops all running instances in the cluster.

1 Ensure that the DAS is running.

Remote subcommands require a running server.

2 Run the `stop-cluster(1)` subcommand.

```
asadmin> stop-cluster cluster-name
```

cluster-name

The name of the cluster that you are stopping.

Example 4-6 Stopping a Cluster

This example stops the cluster `pmcluster`.

```
asadmin> stop-cluster pmcluster
Command stop-cluster executed successfully.
```

- See Also** `stop-cluster(1)`

You can also view the full syntax and options of the subcommand by typing `asadmin help stop-cluster` at the command line.

Troubleshooting If instances in the cluster have become unresponsive and fail to stop, run the subcommand again with the `--kill` option set to `true`. When this option is `true`, the subcommand uses functionality of the operating system to kill the process for each running instance in the cluster.

▼ To Start an Individual Instance Centrally

Use the `start-instance` subcommand in remote mode to start an individual instance centrally.

Before You Begin Ensure that following prerequisites are met:

- The node where the instance resides is enabled for remote communication.
- The user of the DAS can use SSH to log in to the host for the node where the instance resides.

If any of these prerequisites is not met, start the instance locally as explained in [“To Start an Individual Instance Locally” on page 63](#).

1 Ensure that the DAS is running.

Remote subcommands require a running server.

2 Run the `start-instance` subcommand.

```
asadmin> start-instance instance-name
```

Note – Only the options that are required to complete this task are provided in this step. For information about all the options for controlling the behavior of the instance, see the `start-instance(1)` help page.

instance-name

The name of the instance that you are starting.

Example 4-7 Starting an Individual Instance Centrally

This example starts the instance `pmd-i2`, which resides on the node `sj02`. This node represents the host `sj02.example.com`. The configuration of the instance on this node already matched the configuration of the instance in the DAS when the instance was started.

```
asadmin> start-instance pmd-i2
CLI801 Instance is already synchronized
Waiting for pmd-i2 to start .....
Successfully started the instance: pmd-i2
```

```
instance Location: /export/glassfish3/glassfish/nodes/sj02/pmd-i2
Log File: /export/glassfish3/glassfish/nodes/sj02/pmd-i2/logs/server.log
Admin Port: 24851
Command start-local-instance executed successfully.
The instance, pmd-i2, was started on host sj02.example.com
Command start-instance executed successfully.
```

See Also `start-instance(1)`

You can also view the full syntax and options of the subcommand by typing `asadmin help start-instance` at the command line.

Next Steps After starting an instance, you can deploy applications to the instance. For more information, see *GlassFish Server Open Source Edition 3.1 Application Deployment Guide*.

▼ To Stop an Individual Instance Centrally

Use the `stop-instance` subcommand in remote mode to stop an individual instance centrally.

When an instance is stopped, the instance stops accepting new requests and waits for all outstanding requests to be completed.

- 1 **Ensure that the DAS is running.**
Remote subcommands require a running server.
- 2 **Run the `stop-instance(1)` subcommand.**

Example 4-8 Stopping an Individual Instance Centrally

This example stops the instance `pmd-i2`.

```
asadmin> stop-instance pmd-i2
The instance, pmd-i2, is stopped.
Command stop-instance executed successfully.
```

See Also `stop-instance(1)`

You can also view the full syntax and options of the subcommand by typing `asadmin help stop-instance` at the command line.

Troubleshooting If the instance has become unresponsive and fails to stop, run the subcommand again with the `--kill` option set to `true`. When this option is `true`, the subcommand uses functionality of the operating system to kill the instance process.

▼ To Restart an Individual Instance Centrally

Use the `restart -instance` subcommand in remote mode to start an individual instance centrally.

When this subcommand restarts an instance, the DAS synchronizes the instance with changes since the last synchronization as described in the `restart -instance(1)` help page.

If you require different synchronization behavior, stop and start the instance as explained in the following sections:

- [“To Stop an Individual Instance Centrally” on page 56](#)
- [“To Start an Individual Instance Centrally” on page 55](#)

1 Ensure that the DAS is running.

Remote subcommands require a running server.

2 Run the `restart -instance(1)` subcommand.

```
asadmin> restart-instance instance-name
instance-name
```

The name of the instance that you are restarting.

Example 4–9 Restarting an Individual Instance Centrally

This example restarts the instance `pmd-i2`.

```
asadmin> restart-instance pmd-i2
pmd-i2 was restarted.
Command restart-instance executed successfully.
```

- See Also**
- [“To Stop an Individual Instance Centrally” on page 56](#)
 - [“To Start an Individual Instance Centrally” on page 55](#)
 - `restart-instance(1)`

You can also view the full syntax and options of the subcommand by typing `asadmin help restart -instance` at the command line.

Troubleshooting If the instance has become unresponsive and fails to stop, run the subcommand again with the `--kill` option set to `true`. When this option is `true`, the subcommand uses functionality of the operating system to kill the instance process before restarting the instance.

Administering GlassFish Server Instances Locally

Local administration does not require SSH to be set up. If SSH is not set up, you must log in to each host where remote instances reside and administer the instances individually.

Administering GlassFish Server instances locally involves the following tasks:

- [“To Create an Instance Locally” on page 58](#)
- [“To Delete an Instance Locally” on page 61](#)
- [“To Start an Individual Instance Locally” on page 63](#)
- [“To Stop an Individual Instance Locally” on page 64](#)
- [“To Restart an Individual Instance Locally” on page 65](#)

Note – Even if SSH is not set up, you can obtain information about instances in a domain without logging in to each host where remote instances reside. For instructions, see [“To List All Instances in a Domain” on page 51](#).

▼ To Create an Instance Locally

Use the `create-local-instance` subcommand in remote mode to create a GlassFish Server instance locally.

If the instance is managed by GMS, system properties for the instance that relate to GMS must be configured correctly. To avoid the need to restart the DAS and the instance, configure an instance's system properties that relate to GMS when you create the instance. If you change GMS-related system properties for an existing instance, the DAS and the instance must be restarted to apply the changes.

Before You Begin Ensure that the node on which the instance is to reside exists. For information about how to create a node, see the following sections:

- [“To Create an SSH Node” on page 30](#)
- [“To Create a CONFIG Node” on page 30](#)

If you are adding the instance to a cluster, ensure that the cluster to which you are adding the instance exists. For information about how to create a cluster, see [“To Create a Cluster” on page 40](#).

If the instance is to reference an existing named configuration, ensure that the configuration exists. For more information, see [“To Create a Named Configuration” on page 71](#).

The instance might be managed by GMS and reside on a node that represents a multihome host. In this situation, ensure that you have the Internet Protocol (IP) address of the network interface to which GMS binds.

1 Ensure that the DAS is running.

Remote subcommands require a running server.

2 Log in to the host that is represented by the node where the instance is to reside.**3 Run the `create-local-instance` subcommand.**

Note – Only the options that are required to complete this task are provided in this step. For information about all the options for configuring the instance, see the `create-local-instance(1)` help page.

- **If you are creating a standalone instance, do not specify a cluster.**

If the instance is to reference an existing configuration, specify a configuration that no other cluster or instance references.

```
$ asadmin --host das-host [--port admin-port]
--node node-name [--config configuration-name] instance-name
```

das-host

The name of the host where the DAS is running.

admin-port

The HTTP or HTTPS port on which the DAS listens for administration requests. If the DAS listens on the default port for administration requests, you may omit this option.

node-name

The node on which the instance is to reside.

configuration-name

The name of the existing named configuration that the instance will reference.

If you do not require the instance to reference an existing configuration, omit this option. A copy of the `default-config` configuration is created for the instance. The name of this configuration is *instance-name-config*, where *instance-name* is the name of the server instance.

instance-name

Your choice of name for the instance that you are creating.

- **If you are creating a shared instance, specify the configuration that the instance will share with other clusters or instances.**

Do *not* specify a cluster.

```
$ asadmin --host das-host [--port admin-port]
--node node-name --config configuration-name instance-name
```

das-host

The name of the host where the DAS is running.

admin-port

The HTTP or HTTPS port on which the DAS listens for administration requests. If the DAS listens on the default port for administration requests, you may omit this option.

node-name

The node on which the instance is to reside.

configuration-name

The name of the existing named configuration that the instance will reference.

instance-name

Your choice of name for the instance that you are creating.

- **If you are creating a clustered instance, specify the cluster to which the instance will belong.**

If the instance is managed by GMS and resides on a node that represents a multihome host, specify the `GMS-BIND-INTERFACE-ADDRESS-cluster-name` system property.

```
$ asadmin --host das-host [--port admin-port]
create-local-instance --cluster cluster-name --node node-name
[--systemproperties GMS-BIND-INTERFACE-ADDRESS-cluster-name=bind-address] instance-name
```

das-host

The name of the host where the DAS is running.

admin-port

The HTTP or HTTPS port on which the DAS listens for administration requests. If the DAS listens on the default port for administration requests, you may omit this option.

cluster-name

The name of the cluster to which you are adding the instance.

node-name

The node on which the instance is to reside.

bind-address

The IP address of the network interface to which GMS binds. Specify this option only if the instance is managed by GMS and resides on a node that represents a multihome host.

instance-name

Your choice of name for the instance that you are creating.

Example 4–10 Creating a Clustered Instance Locally

This example adds the instance `yml-i1` to the cluster `ymlcluster` locally. The instance resides on the node `sj01`. The DAS is running on the host `das1.example.com` and listens for administration requests on the default port.

```
$ asadmin --host das1.example.com
create-local-instance --cluster ymlcluster --node sj01 yml-i1
Rendezvoused with DAS on das1.example.com:4848.
Port Assignments for server instance yml-i1:
```

```
JMX_SYSTEM_CONNECTOR_PORT=28687
JMS_PROVIDER_PORT=27677
HTTP_LISTENER_PORT=28081
ASADMIN_LISTENER_PORT=24849
JAVA_DEBUGGER_PORT=29009
IIOP_SSL_LISTENER_PORT=23820
IIOP_LISTENER_PORT=23700
OSGI_SHELL_TELNET_PORT=26666
HTTP_SSL_LISTENER_PORT=28182
IIOP_SSL_MUTUALAUTH_PORT=23920
Command create-local-instance executed successfully.
```

- See Also**
- [“To Create an SSH Node” on page 30](#)
 - [“To Create a CONFIG Node” on page 30](#)
 - `create-local-instance(1)`

You can also view the full syntax and options of the subcommand by typing `asadmin help create-local-instance` at the command line.

- Next Steps** After creating an instance, you can start the instance as explained in the following sections:
- [“To Start an Individual Instance Centrally” on page 55](#)
 - [“To Stop an Individual Instance Locally” on page 64](#)

▼ To Delete an Instance Locally

Use the `delete-local-instance` subcommand in remote mode to delete a GlassFish Server instance locally.



Caution – If you are using a Java Message Service (JMS) cluster with a master broker, do not delete the instance that is associated with the master broker. If this instance must be deleted, use the `change-master-broker(1)` subcommand to assign the master broker to a different instance.

Deleting an instance involves the following:

- Removing the instance from the configuration of the DAS
- Deleting the instance's files from file system

Before You Begin Ensure that the instance that you are deleting is not running. For information about how to stop an instance, see the following sections:

- [“To Stop an Individual Instance Centrally” on page 56](#)
- [“To Stop an Individual Instance Locally” on page 64](#)

1 Ensure that the DAS is running.

Remote subcommands require a running server.

2 Log in to the host that is represented by the node where the instance resides.**3 Confirm that the instance is not running.**

```
$ asadmin --host das-host [--port admin-port]
list-instances instance-name
```

das-host

The name of the host where the DAS is running.

admin-port

The HTTP or HTTPS port on which the DAS listens for administration requests. If the DAS listens on the default port for administration requests, you may omit this option.

instance-name

The name of the instance that you are deleting.

4 Run the delete-local-instance(1) subcommand.

```
$ asadmin --host das-host [--port admin-port]
delete-local-instance [--node node-name] instance-name
```

das-host

The name of the host where the DAS is running.

admin-port

The HTTP or HTTPS port on which the DAS listens for administration requests. If the DAS listens on the default port for administration requests, you may omit this option.

node-name

The node on which the instance resides. If only one node is defined in the domain, you may omit this option.

instance-name

The name of the instance that you are deleting.

Example 4–11 Deleting an Instance Locally

This example confirms that the instance `yml-i1` is not running and deletes the instance.

```
$ asadmin --host das1.example.com list-instances yml-i1
yml-i1 not running
Command list-instances executed successfully.
$ asadmin --host das1.example.com delete-local-instance --node sj01 yml-i1
Command delete-local-instance executed successfully.
```

- See Also**
- [“To Stop an Individual Instance Centrally” on page 56](#)
 - [“To Stop an Individual Instance Locally” on page 64](#)
 - `change-master-broker(1)`
 - `delete-local-instance(1)`
 - `list-instances(1)`

You can also view the full syntax and options of the subcommands by typing the following commands at the command line:

- `asadmin help delete-local-instance`
- `asadmin help list-instances`

▼ To Start an Individual Instance Locally

Use the `start-local-instance` subcommand in local mode to start an individual instance locally.

- 1 **Log in to the host that is represented by the node where the instance resides.**
- 2 **Run the `start-local-instance` subcommand.**

```
$ asadmin start-local-instance [--node node-name] instance-name
```

Note – Only the options that are required to complete this task are provided in this step. For information about all the options for controlling the behavior of the instance, see the `start-local-instance(1)` help page.

node-name

The node on which the instance resides. If only one node is defined in the domain, you may omit this option.

instance-name

The name of the instance that you are starting.

Example 4–12 Starting an Individual Instance Locally

This example starts the instance `yml-i1` locally. The instance resides on the node `sj01`.

```
$ asadmin start-local-instance --node sj01 yml-i1
Waiting for yml-i1 to start .....
Successfully started the instance: yml-i1
Instance Location: /export/glassfish3/glassfish/nodes/sj01/yml-i1
Log File: /export/glassfish3/glassfish/nodes/sj01/yml-i1/logs/server.log
Admin Port: 24849
Command start-local-instance executed successfully.
```

See Also `start-local-instance(1)`

You can also view the full syntax and options of the subcommand by typing `asadmin help start-local-instance` at the command line.

Next Steps After starting an instance, you can deploy applications to the instance. For more information, see *GlassFish Server Open Source Edition 3.1 Application Deployment Guide*.

▼ To Stop an Individual Instance Locally

Use the `stop-local-instance` subcommand in local mode to stop an individual instance locally.

When an instance is stopped, the instance stops accepting new requests and waits for all outstanding requests to be completed.

1 Log in to the host that is represented by the node where the instance resides.

2 Run the `stop-local-instance(1)` subcommand.

```
$ asadmin stop-local-instance [--node node-name] instance-name
```

node-name

The node on which the instance resides. If only one node is defined in the domain, you may omit this option.

instance-name

The name of the instance that you are stopping.

Example 4–13 Stopping an Individual Instance Locally

This example stops the instance `yml-i1` locally. The instance resides on the node `sj01`. The DAS is running on the host `das1.example.com` and listens for administration requests on the default port.

```
$ asadmin --host das1.example.com stop-local-instance --node sj01 yml-i1
Waiting for the instance to stop ....
Command stop-local-instance executed successfully.
```

See Also `stop-local-instance(1)`

You can also view the full syntax and options of the subcommand by typing `asadmin help stop-local-instance` at the command line.

Troubleshooting If the instance has become unresponsive and fails to stop, run the subcommand again with the `--kill` option set to `true`. When this option is `true`, the subcommand uses functionality of the operating system to kill the instance process.

▼ To Restart an Individual Instance Locally

Use the `restart-local-instance` subcommand in local mode to restart an individual instance locally.

When this subcommand restarts an instance, the DAS synchronizes the instance with changes since the last synchronization as described in the `restart-local-instance(1)` help page.

If you require different synchronization behavior, stop and start the instance as explained in the following sections:

- [“To Stop an Individual Instance Locally” on page 64](#)
- [“To Start an Individual Instance Locally” on page 63](#)

1 Log in to the host that is represented by the node where the instance resides.

2 Run the `restart-local-instance` subcommand.

```
$ asadmin restart-local-instance [--node node-name] instance-name  
node-name
```

The node on which the instance resides. If only one node is defined in the domain, you may omit this option.

```
instance-name
```

The name of the instance that you are restarting.

Example 4–14 Retarting an Individual Instance Locally

This example restarts the instance `ym1-i1` locally. The instance resides on the node `sj01`.

```
$ asadmin restart-local-instance --node sj01 yml-i1  
Command restart-local-instance executed successfully.
```

See Also `restart-local-instance(1)`

You can also view the full syntax and options of the subcommand by typing `asadmin help restart-local-instance` at the command line.

Troubleshooting If the instance has become unresponsive and fails to stop, run the subcommand again with the `--kill` option set to `true`. When this option is `true`, the subcommand uses functionality of the operating system to kill the instance process before restarting the instance.