

Name create-local-instance – creates a GlassFish Server instance on the machine where the subcommand is run

Synopsis create-local-instance [--help]
 [--node *node-name*] [--nodedir *node-dir*]
 [--config *config-name* | --cluster *cluster-name*]
 [--lbenabled={true|false}]
 [--portbase *port-number*] [--checkports={true|false}]
 [--bootstrap={true|false}] [--savemasterpassword={false|true}]
 [--systemproperties (*name=value*)[:*name=value*]*]
instance-name

Description The create-local-instance command creates a GlassFish Server instance on the machine where the subcommand is run. This subcommand does not require secure shell (SSH) to be configured. You must run this command from the machine where the instance is to reside.

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 machine where the GlassFish Server instance resides. The JVM machine must be compatible with the Java Platform, Enterprise Edition (Java EE).

A GlassFish Server instance requires a reference to the following items:

- The node that defines the machine where the instance resides. The node can be specified in the command to create the instance, but is required only if more than one node exists in the directory where nodes are stored. If no node is specified, the behavior of the subcommand depends on the number of existing nodes in the directory where nodes are stored:
 - If no nodes exist, the subcommand creates a node for the instance. The name of the node is the name of the machine on which the subcommand is run.
 - If only one node exists, the subcommand creates a reference to the existing node for the instance.
 - If two or more nodes exist, an error occurs.
- The named configuration that defines the configuration of the instance. The configuration can be specified in the command to create the instance, but is not required. If no configuration is specified for an instance that is not joining a cluster, the subcommand creates a configuration for the instance. An instance that is joining a cluster receives its configuration from its parent cluster.

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 no configuration or cluster is specified in the command to create the instance.

When a standalone instance is created, 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 named configuration 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.

By default, this subcommand attempts to resolve possible port conflicts for the instance that is being created. The subcommand also assigns ports that are currently not in use and not already assigned to other instances on the same node. The subcommand assigns these ports randomly. Use the `--systemproperties` option to resolve port conflicts for additional instances on the same node. System properties of an instance can be manipulated by using the [create-system-properties\(1\)](#) subcommand and the [delete-system-property\(1\)](#) subcommand.

Options

`--help`

`-?`

Displays the help text for the subcommand.

`--node`

The name of the node that defines the machine where the instance is to be created. The node must be specified only if more than one node exists in the directory where nodes are stored. Otherwise, the node may be omitted. If a node is specified, the node must exist.

If no node is specified, the behavior of the subcommand depends on the number of existing nodes in the directory where nodes are stored:

- If no nodes exist, the subcommand creates a node for the instance. The name of the node is the name of the machine on which the subcommand is run.
- If only one node exists, the subcommand creates a reference to the existing node for the instance.
- If two or more nodes exist, an error occurs.

`--nodedir`

The path to the directory in which the instance's node is to be stored. The default is `as-install/nodes`.

--config

Specifies the named configuration that the instance references. The configuration must exist and must not be named `default-config` or `server-config`. If the configuration name specifies a standalone configuration, an error occurs. Specifying the `--config` option creates a shared instance.

The `--config` option and the `--cluster` option are mutually exclusive. If both options are omitted, a standalone instance is created.

--cluster

Specifies the cluster from which the instance inherits its configuration. Specifying the `--cluster` option creates a clustered instance.

The `--config` option and the `--cluster` option are mutually exclusive. If both options are omitted, a standalone instance is created.

--lbenabled

Specifies whether the instance is enabled for load balancing. Possible values are as follows:

true

The instance is enabled for load balancing (default).

When an instance is enabled for load balancing, a load balancer sends requests to the instance.

false

The instance is disabled for load balancing.

When an instance is disabled for load balancing, a load balancer does not send requests to the instance.

--portbase

Determines the number with which the port assignment should start. An instance uses a certain number of ports that are statically assigned. The *portbase* value determines where the assignment should start. The values for the ports are calculated as follows:

- Administration port: *portbase* + 48
- HTTP listener port: *portbase* + 80
- IIOP listener port: *portbase* + 37
- JMX port: *portbase* + 86

When the `--portbase` option is specified, the output of this subcommand includes a complete list of used ports.

--checkports

Specifies whether to check for the availability of the administration, HTTP, JMS, JMX, and IIOP ports. The default value is `true`.

--bootstrap

Specifies whether the files that are required for secure synchronization with the domain administration server (DAS) are retrieved. The instance is synchronized with the DAS when the instance is started. Possible values are as follows:

true

The files are retrieved (default).

Set this option to `true` if the DAS is configured for secure communication with the instance.

false

The files are not retrieved.

Set this option to `false` only if the DAS is not configured for secure communication with the instance. If this option is `false`, the instance is created more quickly.

--savemasterpassword

Setting this option to `true` allows the master password to be written to the file system. If the master password is written to the file system, the instance can be started without the need to prompt for the password. Because writing the master password to the file system is an insecure practice, the default is `false`.

--systemproperties

Defines system properties for the instance. These properties override property definitions for port settings in the instance's configuration. Predefined port settings must be overridden if, for example, two clustered instances reside on the same machine. In this situation, port settings for one instance must be overridden because both instances share the same configuration.

The following properties are available:

ASADMIN_LISTENER_PORT

This property specifies the port number of the HTTP port or the HTTPS port for administration. This port is the port in the URL that you specify in your web browser to manage the instance, for example, `http://localhost:4949`. Valid values are 1–65535. On UNIX, creating sockets that listen on ports 1–1024 requires superuser privileges.

HTTP_LISTENER_PORT

This property specifies the port number of the port that is used to listen for HTTP requests. Valid values are 1–65535. On UNIX, creating sockets that listen on ports 1–1024 requires superuser privileges.

HTTP_SSL_LISTENER_PORT

This property specifies the port number of the port that is used to listen for HTTPS requests. Valid values are 1–65535. On UNIX, creating sockets that listen on ports 1–1024 requires superuser privileges.

IIOP_LISTENER_PORT

This property specifies the port number of the port that is used for IIOP connections. Valid values are 1–65535. On UNIX, creating sockets that listen on ports 1–1024 requires superuser privileges.

IIOP_SSL_LISTENER_PORT

This property specifies the port number of the port that is used for secure IIOP connections. Valid values are 1–65535. On UNIX, creating sockets that listen on ports 1–1024 requires superuser privileges.

IIOP_SSL_MUTUALAUTH_PORT

This property specifies the port number of the port that is used for secure IIOP connections with client authentication.

JMS_PROVIDER_PORT

This property specifies the port number for the Java Message Service provider. Valid values are 1–65535. On UNIX, creating sockets that listen on ports 1–1024 requires superuser privileges.

JMX_SYSTEM_CONNECTOR_PORT

This property specifies the port number on which the JMX connector listens. Valid values are 1–65535. On UNIX, creating sockets that listen on ports 1–1024 requires superuser privileges.

Operands *instance-name*

The name of the instance that is being created. Each instance in a domain must have a name that is unique across all nodes, GlassFish Server instances, clusters, and named configurations.

Examples **EXAMPLE 1** Creating a Standalone GlassFish Server Instance

This example creates the standalone instance `il3` on the machine where the command is run. The instance references the only existing node.

```
asadmin> create-local-instance il3
Using DAS host localhost and port 4848 from existing das.properties for
node null. To use a different DAS, create a new node using
create-node-ssh or create-node-config. Then create the instance with the
new node and correc host and port: asadmin --host das_host --port
das_port create-local-instance --node node_name instance_name.
Attempting to rendezvous with DAS on host localhost port 4848
Uptime: 4 hours, 43 minutes, 3 seconds
```

The instance has rendezvoused with the DAS and will be using host localhost port 4848 for future communication.

```
Port Assignments for server instance il3:
JMX_SYSTEM_CONNECTOR_PORT=28691
JMS_PROVIDER_PORT=27682
```

EXAMPLE 1 Creating a Standalone GlassFish Server Instance *(Continued)*

```
ASADMIN_LISTENER_PORT=24854
HTTP_LISTENER_PORT=28085
IIOP_LISTENER_PORT=23705
IIOP_SSL_LISTENER_PORT=23825
HTTP_SSL_LISTENER_PORT=28186
IIOP_SSL_MUTUALAUTH_PORT=23925
```

Command `create-local-instance` executed successfully.

EXAMPLE 2 Creating a Clustered GlassFish Server Instance on a Specific Node

This example creates the clustered instance `yml1` on node `sj02`. The instance is a member of the cluster `ymlclust`.

```
asadmin> create-local-instance --cluster ymlclust --node sj02 yml1
```

```
Using DAS host localhost and port 4848 from existing das.properties
for node sj02. To use a different DAS, create a new node using
create-node-ssh or create-node-config. Then create the instance with
the new node and correct host and port:asadmin --host das_host --port
das_port create-local-instance --node node_name instance_name.
Attempting to rendezvous with DAS on host localhost port 4848
Uptime: 5 hours, 6 minutes, 6 seconds
```

The instance has rendezvoused with the DAS and will be using host `localhost` port `4848` for future communication.

Port Assignments for server instance `yml1`:

```
JMX_SYSTEM_CONNECTOR_PORT=28691
JMS_PROVIDER_PORT=27682
ASADMIN_LISTENER_PORT=24854
HTTP_LISTENER_PORT=28085
IIOP_LISTENER_PORT=23705
IIOP_SSL_LISTENER_PORT=23825
HTTP_SSL_LISTENER_PORT=28186
IIOP_SSL_MUTUALAUTH_PORT=23925
```

Command `create-local-instance` executed successfully.

Exit Status	0	command executed successfully
	1	error in executing the command

See Also [create-instance\(1\)](#), [create-node-config\(1\)](#), [create-node-ssh\(1\)](#), [create-system-properties\(1\)](#), [delete-local-instance\(1\)](#), [delete-system-property\(1\)](#), [list-instances\(1\)](#), [start-local-instance\(1\)](#), [stop-local-instance\(1\)](#)

[asadmin\(1M\)](#)

