

**Name** create-service – configures the starting of a DAS or a GlassFish Server instance on an unattended boot

**Synopsis** create-service [--help] [--name *name*]  
 [--serviceproperties *serviceproperties*]  
 [--dry-run={false|true}] [--force[={false|true}]]  
 [--serviceuser *serviceuser*]  
 [--domaindir *domain-dir*]  
 [--nodedir *node-dir*] [--node *node*]  
 [*domain-or-instance-name*]

**Description** The create-service subcommand configures the starting of a domain administration server (DAS) or a GlassFish Server instance on an unattended boot on Windows, Ubuntu Linux, and Oracle Solaris systems.

If no arguments are specified, the subcommand configures the starting of the default domain.

If the operand specifies an instance, the create-service subcommand does not contact the domain administration server (DAS) to determine the node on which the instance resides. To determine the node on which the instance resides, the subcommand searches the directory that contains the node directories. If multiple node directories exist, the node must be specified as an option of the subcommand.

The subcommand can determine whether the supplied operand is a DAS or an instance.

This subcommand is supported in local mode only.

**Behavior of create-service on Windows Systems** On Windows systems, the create-service subcommand creates a Windows service to represent the DAS or instance. After this subcommand creates the service, you must use the Windows Services Manager or the Windows Services Wrapper to start, stop, uninstall, or install the service.

The subcommand creates the following Windows Services Wrapper files for the service in the *domain-dir*\bin directory or the *instance-dir*\bin directory:

- Configuration file: *service-name*Service.xml
- Executable file: *service-name*Service.exe

On Windows systems, this subcommand requires the [Microsoft .NET Framework](#). Otherwise, the subcommand fails.

**Behavior of create-service on Linux Systems** On Linux systems, the create-service subcommand creates a System-V-style initialization script /etc/init.d/GlassFish\_*domain-or-instance-name* and installs a link to this script. After this subcommand creates the script, you must use this script to start, stop, or restart the domain or instance.

Behavior of  
create-service on  
Oracle Solaris Systems

On Oracle Solaris systems, the `create-service` subcommand creates a Service Management Facility (SMF) service to represent the DAS or instance. After this subcommand creates the service, you must use SMF commands to start, enable, disable, delete, or stop the service.

This subcommand must be run as the OS-level user with superuser privileges. The configuration file for the DAS or instance must be stored in a directory to which the superuser has access and cannot be stored on a network file system. The service that is created is controlled by the OS-level user who owns the directory where the configuration of the DAS or instance resides.

To run this subcommand, you must have `solaris.smf.*` authorization. For information about how to grant authorizations to users, see the `useradd(1M)` and `usermod(1M)` man pages.

On Oracle Solaris, the manifest file is created in the following directory:

```
/var/svc/manifest/application/GlassFish/domain-or-instance-name_domain-or-instance-root-dir
```

You must also have write permission in the directory tree

```
/var/svc/manifest/application/GlassFish.
```

Usually, the superuser has both these permissions. To run these commands as non-root user, the system administrator must be contacted so that the relevant authorizations are granted. You must also ensure that the following conditions are met:

- Oracle Solaris 10 administration commands such as `svccfg(1M)`, `svcs(1)`, and `auths(1)` are available through the `PATH` statement, so that these commands can be executed. A simple test to do so is to run the command `which svccfg` in the shell.
- You must have write permission for the path `/var/svc/manifest/application/GlassFish.`

If you delete a service that you created by using the `create-service` subcommand, you must delete the directory that contains the manifest file and the entire contents of the directory. Otherwise, an attempt to re-create the service by using the `create-service` subcommand fails. The Oracle Solaris command `svccfg` does *not* delete this directory.

## Options

`--help`

`-?`

Displays the help text for the subcommand.

`--name`

(Windows and Oracle Solaris systems only) The name of the service that you will use when administering the service through Oracle Solaris SMF commands or the service management features of the Windows operating system. If a default is present, this name overrides the default.

`--serviceproperties`

(Oracle Solaris systems only) Specifies a colon(:)-separated list of various properties that are specific to the service. For Oracle Solaris 10, if you specify `net_privaddr`, the service's

processes will be able to bind to the privileged ports (<1024) on the platform. You can bind to ports < 1024 only if the owner of the service is superuser, otherwise, this is not allowed.

**--dry-run**

Previews your attempt to create a service. Indicates issues and the outcome that will occur if you run the command without using the `--dry-run` option. Nothing is actually configured. Default is false.

**--force**

Specifies whether the service is created even if validation of the service fails.

Possible values are as follows:

**true**

The service is created even if validation of the service fails.

**false**

The service is not created (default).

**--serviceuser**

(Linux systems only) The user that is to run the GlassFish Server software when the service is started. The default is the user that is running the subcommand. Specify this option if the GlassFish Server software is to be run by a user other than the root user.

**--domaindir**

The absolute path of the directory on the disk that contains the configuration of the domain. This is the If this option is specified, the operand must specify a domain.

**--nodedir**

Specifies the directory that contains the instance's node directory. The instance's files are stored in the instance's node directory. The default is *as-install/nodes*. If this option is specified, the operand must specify an instance.

**--node**

Specifies the node on which the instance resides. This option may be omitted only if the directory that the `--nodedir` option specifies contains only one node directory. Otherwise, this option is required. If this option is specified, the operand must specify an instance.

**Operands** *domain-or-instance-name*

The name of the domain or instance to configure. If no operand is specified, the default domain is used.

**Examples** **EXAMPLE 1** Creating a Service on a Windows System

This example creates a service for the default domain on a system that is running Windows.

```
asadmin> create-service
```

```
Found the Windows Service and successfully uninstalled it.
```

```
The Windows Service was created successfully. It is ready to be started. Here are the details:
```

```
ID of the service: domain1
```

**EXAMPLE 1** Creating a Service on a Windows System    *(Continued)*

```

Display Name of the service:domain1 GlassFish Server
Domain Directory: C:\glassfishv3\glassfish\domains\domain1
Configuration file for Windows Services Wrapper: C:\glassfishv3\glassfish\domains\
domain1\bin\domain1Service.xml
The service can be controlled using the Windows Services Manager or you can use the
Windows Services Wrapper instead:
Start Command: C:\glassfishv3\glassfish\domains\domain1\bin\domain1Service.exe start
Stop Command: C:\glassfishv3\glassfish\domains\domain1\bin\domain1Service.exe stop
Uninstall Command: C:\glassfishv3\glassfish\domains\domain1\bin\domain1Service.exe
uninstall
Install Command: C:\glassfishv3\glassfish\domains\domain1\bin\domain1Service.exe
install

```

This message is also available in a file named PlatformServices.log in the domain's root directory  
 Command create-service executed successfully.

**EXAMPLE 2** Creating a Service on a Linux System

This example creates a service for the default domain on a system that is running Linux.

```

asadmin> create-service
Adding system startup for /etc/init.d/GlassFish_domain1 ...
/etc/rc0.d/K20GlassFish_domain1 -> ../init.d/GlassFish_domain1
/etc/rc1.d/K20GlassFish_domain1 -> ../init.d/GlassFish_domain1
/etc/rc6.d/K20GlassFish_domain1 -> ../init.d/GlassFish_domain1
/etc/rc2.d/S20GlassFish_domain1 -> ../init.d/GlassFish_domain1
/etc/rc3.d/S20GlassFish_domain1 -> ../init.d/GlassFish_domain1
/etc/rc4.d/S20GlassFish_domain1 -> ../init.d/GlassFish_domain1
/etc/rc5.d/S20GlassFish_domain1 -> ../init.d/GlassFish_domain1
The Service was created successfully. Here are the details:
Name of the service:domain1
Type of the service:Instance
Configuration location of the service:/etc/init.d/GlassFish_domain1
User account that will run the service: root
You have created the service but you need to start it yourself. Here
are the most typical Linux commands of interest:

* /etc/init.d/GlassFish_domain1 start
* /etc/init.d/GlassFish_domain1 stop
* /etc/init.d/GlassFish_domain1 restart
* update-rc.d GlassFish_domain1 remove // make sure to delete
/etc/init.d/GlassFish_domain1 first
For your convenience this message has also been saved to this file:
/export/glassfish3/glassfish/nodes/localhost/domain1
Command create-service executed successfully.

```

**EXAMPLE 3** Creating a Service on an Oracle Solaris System

This example creates a service for the default domain on a system that is running Oracle Solaris.

```
asadmin> create-service
```

The Service was created successfully. Here are the details:

Name of the service:application/GlassFish/domain1

Type of the service:Domain

Configuration location of the service:/home/gfuser/glassfish-installations  
/glassfishv3/glassfish/domains

Manifest file location on the system:/var/svc/manifest/application  
/GlassFish/domain1\_home\_gfuser\_glassfish-installations\_glassfishv3  
\_glassfish\_domains/Domain-service-smf.xml.

You have created the service but you need to start it yourself.

Here are the most typical Solaris commands of interest:

```
* /usr/bin/svcs -a | grep domain1 // status
* /usr/sbin/svcadm enable domain1 // start
* /usr/sbin/svcadm disable domain1 // stop
* /usr/sbin/svccfg delete domain1 // uninstall
```

Command create-service executed successfully.

<b>Exit Status</b>	0	subcommand executed successfully
	1	error in executing the subcommand

**See Also** [asadmin\(1M\)](#)

[auths\(1\)](#), [svcs\(1\)](#)

[svccfg\(1M\)](#), [useradd\(1M\)](#), [usermod\(1M\)](#)

[Microsoft .NET Framework \(http://www.microsoft.com/net/\)](http://www.microsoft.com/net/)