

**Name** add-resources – creates the resources specified in an XML file

**Synopsis** add-resources [--help] [--target *target*]  
[--upload={false|true}] *xml-file-name*

**Description** The add-resources subcommand creates the resources named in the specified XML file. The resources that can be created with this subcommand are listed in See Also in this help page.

The --target option specifies the target for which you are creating the resources. If this option specifies the domain, the resources are added only to the configuration of the domain administration server (DAS). If this option specifies any other target, the resources are added to the configuration of the DAS and references are added to the resources from the specified target.

The *xml-file-name* operand is the path to the XML file that contains the resources to be created. The DOCTYPE must be specified as [http://glassfish.org/dtds/glassfish-resources\\_1\\_5.dtd](http://glassfish.org/dtds/glassfish-resources_1_5.dtd) in the resources.xml file.

This subcommand is supported in remote mode only.

**Options** --help

-?

Displays the help text for the subcommand.

--target

Specifies the target for which you are creating the resources.

Valid values are as follows:

*server*

Creates the resources for the default server instance *server* and is the default value.

*domain*

Creates the resources for the domain.

*cluster-name*

Creates the resources for every server instance in the cluster.

*instance-name*

Creates the resources for a particular GlassFish Server instance.

--upload

Specifies whether the subcommand uploads the file to the DAS. In most situations, this option can be omitted.

Valid values are as follows:

*false*

The subcommand does not upload the file and attempts to access the file through the specified file name. If the DAS cannot access the file, the subcommand fails.

For example, the DAS might be running as a different user than the administration user and does not have read access to the file. In this situation, the subcommand fails if the `--upload` option is `false`.

`true`

The subcommand uploads the file to the DAS over the network connection.

The default value depends on whether the DAS is on the host where the subcommand is run or is on a remote host.

- If the DAS is on the host where the subcommand is run, the default is `false`.
- If the DAS is on a remote host, the default is `true`.

### Operands *xml-file-name*

The path to the XML file that contains the resources that are to be created. You can specify an absolute path, only the file name, or a relative path.

- If you specify an absolute path, the XML file can be anywhere.
- If you specify only the file name, the XML file must reside in the `as-install/domains/domain1/config` directory on the DAS host. This requirement must be met even if you run the subcommand from another host.
- If you specify a relative path, the XML file must be in the relative directory.

An example XML file follows.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE resources PUBLIC
  "-//GlassFish.org//DTD GlassFish Application Server 3.1 Resource Definitions //EN"
  "http://glassfish.org/dtds/glassfish-resources_1_5.dtd">
<resources>
  <jdbc-connection-pool name="SPECjPool" steady-pool-size="100"
    max-pool-size="150" max-wait-time-in-millis="60000"
    pool-resize-quantity="2" idle-timeout-in-seconds="300"
    is-isolation-level-guaranteed="true"
    is-connection-validation-required="false"
    connection-validation-method="auto-commit"
    fail-all-connections="false"
    datasource-classname="oracle.jdbc.pool.OracleDataSource">
    <property name="URL"
      value="jdbc:oracle:thin:@iasperfsol12:1521:specdb"/>
    <property name="User" value="spec"/>
    <property name="Password" value="spec"/>
    <property name="MaxStatements" value="200"/>
    <property name="ImplicitCachingEnabled" value="true"/>
  </jdbc-connection-pool>
  <jdbc-resource enabled="true" pool-name="SPECjPool"
    jndi-name="jdbc/SPECjDB"/>
</resources>
```

**Examples** EXAMPLE 1 Adding Resources

This example creates resources using the contents of the XML file `resource.xml`.

```
asadmin> add-resources resource.xml
Command : Connector connection pool jms/testQFactoryPool created.
Command : Administered object jms/testQ created.
Command : Connector resource jms/testQFactory created.
Command : Resource adapter config myResAdapterConfig created successfully
Command : JDBC connection pool DerbyPoolA created successfully.
Command : JDBC resource jdbc/__defaultA created successfully.
Command add-resources executed successfully.
```

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

**See Also** [create-jdbc-connection-pool\(1\)](#), [create-jdbc-resource\(1\)](#), [create-jms-resource\(1\)](#), [create-jndi-resource\(1\)](#), [create-javamail-resource\(1\)](#), [create-custom-resource\(1\)](#), [create-connector-resource\(1\)](#), [create-connector-work-security-map\(1\)](#), [create-admin-object\(1\)](#), [create-resource-adapter-config\(1\)](#)

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