

Name create-resource-adapter-config – creates the configuration information for the connector module

Synopsis create-resource-adapter-config [--help] [--threadpoolid *threadpool*]
[--objecttype *object-type*] [--property (*property-name=value*)[:*name=value*]*]
raname

Description The create-resource-adapter-config subcommand creates configuration information for the connector module. This subcommand can be run before deploying a resource adapter, so that the configuration information is available at the time of deployment. The resource adapter configuration can also be created after the resource adapter is deployed. In this case, the resource adapter is restarted with the new configuration. You must first create a thread pool, using the create-threadpool subcommand, and then identify that thread pool value as the ID in the --threadpoolid option.

This subcommand is supported in remote mode only.

Application Scoped Resources The create-resource-adapter-config subcommand can be scoped such that the resource configuration is available only to the application for which it is defined. Configurations that are scoped in this way cannot be referred to or looked up by other applications. If an application tries to look up a scoped resource that does not belong to the application (defined in another application) or module (defined in another module), a naming exception is thrown.

To scope a resource to a specific application, define the resource in a `glassfish-resources.xml` file in the RAR archive for the corresponding connector module. The `glassfish-resources.xml` should be a valid XML file that conforms to the `glassfish-resources.dtd`.

The scoping can be applied at the application level or module level.

- To scope a resource at the application level, place the `glassfish-resources.xml` containing the resource definition in the top-level `META-INF` directory of the RAR archive (`META-INF/glassfish-resources.xml`). The resource can be referenced through lookups with the `java:app` prefix. The resource will be available only to the application (and its modules) for which it is defined.
- To scope a resource at the module level, place the `glassfish-resources.xml` containing the resource definition in the RAR bundled at the module level. The resource can be referenced through lookups with the `java:module` prefix. The resource will be available only to the module for which it is defined.

The `jndi-name` for `application-scoped-resources` or `module-scoped-resources` are specified using the format `java:app/jdbc/myDataSource` or `java:module/jdbc/myModuleLevelDataSource`. This naming scope is defined in the [Java EE 6 Specification](http://download.oracle.com/javase/6/api/) (<http://download.oracle.com/javase/6/api/>).

Note – Note the following restrictions:

- If the prefix `java:app/` or `java:module/` is not specified in the resource name, the prefix will be applied appropriately and persisted in `domain.xml`.
- The `java:global` prefix is not supported in GlassFish Server 3.1.
- The `create-resource-adapter-config` configuration can only be defined in the `glassfish-resources.xml` for a corresponding connector module (standalone RAR). In the case of an EAR archive, the `create-resource-adapter-config` configuration can only be defined in a `glassfish-resources.xml` in a RAR that is bundled in the EAR. It is not permissible to define the resource in a RAR that is not part of the EAR with which the resource will be used.

Options `--help`

`-?`

Displays the help text for the subcommand.

`--target`

This option has been deprecated.

`--threadpoolid`

The thread pool ID from which the work manager gets the thread. This option takes only one thread pool ID.

`--objecttype`

The default is `user`.

`--property`

Keyword-value pairs that specify additional configuration properties of the resource adapter Java bean. The keyword-value pairs are separated by a colon (:). The properties are the names of setter methods of the class that is referenced by the `resourceadapter-class` element in the `ra.xml` file.

Operands *raname*

Indicates the connector module name. It is the value of the `resource-adapter-name` in the `domain.xml` file.

Examples EXAMPLE 1 Creating a Resource Adapter Configuration

This example creates a resource adapter configuration for `ra1`.

```
asadmin> create-resource-adapter-config --property foo=bar --threadpoolid
mycustomerthreadpool ra1
```

Command `create-resource-adapter-config` executed successfully

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

See Also `create-threadpool(1)`, `delete-resource-adapter-config(1)`,
`list-resource-adapter-configs(1)`
`asadmin(1M)`