icegridnode

An IceGrid node is a process that activates, monitors, and deactivates registered server processes. You can run any number of nodes in a domain, but typically there is one node per host. A node must be running on each host on which servers are activated automatically, and nodes cannot run without an IceGrid registry.

The IceGrid node server is implemented by the `icegridnode` executable. If you wish to run a registry and node in one process, `icegridnode` is the executable you must use.

On this page:

- Command Line Options for icegridnode
- Configuring Node Endpoints
- Node Security Considerations
- Configuring a Data Directory for the Node
- Node Configuration Example

Command Line Options for `icegridnode`

The node supports the following command-line options:

```
Usage: icegridnode [options]
Options:
-h, --help            Show this message.
-v, --version         Display the Ice version.
--nowarn              Don't print any security warnings.
--readonly            Start the collocated master registry in
                      read-only mode.
--initdb-from-replica=<replica>
                      Initialize the collocated registry database from the
                      given replica.

--deploy DESCRIPTOR [TARGET1 [TARGET2 ...]]
                      Add or update descriptor in file DESCRIPTOR,
                      with optional targets.
```

If you are running the node with a collocated registry, the `--readonly` option prevents any updates to the registry's database; it also prevents slaves from synchronizing their databases with this master. This option is useful when you need to verify that the master registry's database is correct after promoting a slave to become the new master. The `--initdb-from-replica` option, added in Ice 3.5.1, allows you to initialize the database from another registry replica. This option is useful when you need to start a new master with the contents of a slave database.

The `--deploy` option allows an application to be deployed automatically as the node process starts, which can be especially useful during testing. The command expects the name of the XML deployment file, and optionally allows the names of the individual targets within the file to be specified.

Additional command line options are supported, including those that allow the node to run as a Windows service or Unix daemon, and Ice includes a utility to help you install an IceGrid node as a Windows service.

Configuring Node Endpoints

The IceGrid node's endpoints are defined by the `IceGrid.Node.Endpoints` property and must be accessible to the registry. It is not necessary to use a fixed port because each node contacts the registry at startup to provide its current endpoint information.

Node Security Considerations

It is important that you give careful consideration to the permissions of the account under which the node runs. If the servers that the node will activate have no special access requirements, and all of the servers can use the same account, it is recommended that you do not run the node under an account with system privileges, such as the root account on Unix or the Administrator account on Windows.

Configuring a Data Directory for the Node

The node requires an empty directory that it can use to store server files. The pathname of this directory is supplied by the configuration property `IceGrid.Node.Data`. To clear a node's state, first ensure the server is not currently running, then remove all of the files in its data directory and restart the server.

> ⊘ The node's data directory may also contain files and subdirectories used by your application's servers, such as configuration files and Freeze database environments. Before destroying the contents of the node's data directory, make sure that all servers are stopped and any important files are backed up.

When running a collocated node and registry server, we recommend using separate directories for the registry and node data directories.

Node Configuration Example

A minimal node configuration is shown in the following example:

```
IceGrid.Node.Endpoints=tcp
IceGrid.Node.Name=Node1
IceGrid.Node.Data=/opt/ripper/node

Ice.Default.Locator=IceGrid/Locator:tcp -p 4061
```

The value of the `IceGrid.Node.Name` property must match that of a deployed node known by the registry.

The `Ice.Default.Locator` property is used by the node to contact the registry. The value is a proxy that contains the registry's client endpoints.

If you wish to run a collocated registry and node server, enable the property `IceGrid.Node.CollocateRegistry` and include the registry's configuration properties.

The remaining configuration properties are discussed in IceGrid Properties.

See Also

- IceGrid Server Activation
- Promoting a Registry Slave
- IceGrid Persistent Data
- Getting Started with IceGrid
- Using IceGrid Deployment
- Windows Services
- IceGrid Properties