

Using the Ice Service Installer

Ice provides the command-line tool `iceserviceinstall` to assist you in installing and uninstalling the following Windows services:

- [IceGrid registry](#)
- [IceGrid node](#)
- [Glacier2 router](#)

Ice includes other programs that can also be run as Windows services, such as the [IceBox](#) and [IcePatch2](#) servers. Typically it is not necessary to install these programs as Windows services because they can be launched by an IceGrid node service. However, if you wish to run an IceBox or IcePatch2 server as a Windows service without the use of IceGrid, you must [manually install](#) the service.

Here we describe how to use the Ice service installer and discuss its actions and prerequisites.

On this page:

- [iceserviceinstall Command Line Options](#)
- [Security Considerations for Ice Services](#)
- [iceserviceinstall Configuration File](#)
 - [Sample Configuration Files](#)
- [iceserviceinstall Properties](#)
- [Service Installation Process](#)
- [Uninstalling a Service](#)

iceserviceinstall Command Line Options

`iceserviceinstall` supports the following options and arguments:

```
iceserviceinstall [options] service config-file [property ...]
```

Options:

<code>-h, --help</code>	Show this message.
<code>-n, --nopause</code>	Do not call pause after displaying a message.
<code>-v, --version</code>	Display the Ice version.
<code>-u, --uninstall</code>	Uninstall the Windows service.

The *service* and *config-file* arguments are required during installation and uninstallation.

The *service* argument selects the type of service you are installing; use one of the following values:

- `icegridregistry`
- `icegridnode`
- `glacier2router`

Note that the Ice service installer currently does not support the installation of an IceGrid node with a collocated registry, therefore you must install the registry and node separately.

The *config-file* argument specifies the name of an [Ice configuration file](#).

When installing a service, properties can be defined on the command line using the `--name=value` syntax, or they can be defined in the configuration file. The supported properties are described [below](#).

Security Considerations for Ice Services

None of the Ice services require privileges beyond a normal user account. In the case of the IceGrid node service in particular, we do not recommend running it in a [user account](#) with elevated privileges because the service is responsible for launching server executables, and those servers would inherit the node's access rights.

iceserviceinstall Configuration File

The Ice service installer requires that you specify the path name of the Ice configuration file for the service being installed or uninstalled. The tool needs this path name for several reasons:

- During installation, it verifies that the configuration file has sufficient access rights.
- It configures a newly-installed service to load the configuration file using its absolute path name, therefore you must decide in advance where the file will be located.
- It reads the configuration file and examines certain service-specific properties. For example, prior to installing an IceGrid registry service, the tool verifies that the directory specified by the property `IceGrid.Registry.Data` has sufficient access rights.
- The tool supports its own configuration parameters that may also be defined as `properties` in this file.

You may still modify a service's configuration file after installation, but you should uninstall and reinstall the service if you change any of the properties that influence the service installer's actions. The table below describes the service properties that affect the installer:

Property	Service	Description
<code>IceGrid.InstanceName</code>	IceGrid Registry	Value appears in the service name; also included in the default display name if one is not defined.
<code>IceGrid.Node.Data</code>	IceGrid Node	Directory is created if necessary; access rights are verified.
<code>IceGrid.Node.Name</code>	IceGrid Node	Value appears in the service name; also included in the default display name if one is not defined.
<code>IceGrid.Registry.Data</code>	IceGrid Registry	Directory is created if necessary; access rights are verified.
<code>Ice.Default.Locator</code>	IceGrid Node, Glacier2 Router	The IceGrid instance name is derived from the identity in this proxy.
<code>Ice.EventLog.Source</code>	All	Specifies the name of an event log source for the service.

The steps performed by the tool during an installation are described in detail [below](#).

Sample Configuration Files

Ice includes sample configuration files for the IceGrid and Glacier2 services in the `config` subdirectory of your Ice installation. We recommend that you review the comments and settings in these files to familiarize yourself with a typical configuration of each service.

You can modify a configuration file to suit your needs or copy one to use as a starting point for your own configuration.

iceserviceinstall Properties

The Ice service installer uses a set of optional properties that customize the installation process. These properties can be defined in the service's configuration file as discussed above, or they can be defined on the command line using the familiar `--name=value` syntax:

```
iceserviceinstall --DependOnRegistry=1 ...
```

The installer's properties are listed below:

- `AutoStart=num`
If num is a value greater than zero, the service is configured to start automatically at system start up. Set this property to zero to configure the service to start on demand instead. If not specified, the default value is 1.
- `DependOnRegistry=num`
If num is a value greater than zero, the service is configured to depend on the IceGrid registry, meaning Windows will start the registry prior to starting this service. Enabling this feature also requires that the property `Ice.Default.Locator` be defined in *config-file*. If not specified, the default value is zero.
- `Description=value`
A brief description of the service. If not specified, a general description is used.
- `DisplayName=name`
The friendly name that identifies the service to the user. If not specified, `iceserviceinstall` composes a default display name.
- `EventLog=name`
The name of the event log used by the service. If not specified, the default value is `Application`.
- `ImagePath=path`
The path name of the service executable. If not specified, `iceserviceinstall` assumes the service executable resides in the same directory as itself and fails if the executable is not found.

- `ObjectName=name`
Specifies the account used to run the service. If not specified, the default value is `NT Authority\LocalService`.
- `Password=value`
The password required by the account specified in `ObjectName`.

Service Installation Process

The Ice service installer performs a number of steps to install a service. As discussed [earlier](#), you must specify the path name of the service's configuration file because the service installer uses certain properties during the installation process. The actions taken by the service installer are described below:

- Obtain the service's *instance name* from the configuration file. The instance name is specified by the property `IceGrid.InstanceName` or `Glacier2.InstanceName`. If an instance name is not specified, the default value is `IceGrid` or `Glacier2`, respectively. If the service being installed depends on the IceGrid registry, the IceGrid instance name is derived from the value of the `Ice.Default.Locator` property.
- For an IceGrid node, obtain the node's name from the property `IceGrid.Node.Name`. This property must be defined when installing a node.
- Compose the service name from the service type, instance name, and node name (for an IceGrid node). For example, the default service name for an IceGrid registry is `icegridregistry.IceGrid`. Note that the service name is not the same as the display name.
- Resolve the user account specified by `ObjectName`.
- For an IceGrid registry, create the data directory specified by the property `IceGrid.Registry.Data` and ensure that the user account specified by `ObjectName` has read/write access to the directory.
- For an IceGrid node, create the data directory specified by the property `IceGrid.Node.Data` and ensure that the user account specified by `ObjectName` has read/write access to the directory.
- For an IceGrid node, ensure that the user account specified by `ObjectName` has read access to the following registry key:
`HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib`
This key allows the node to access [CPU utilization statistics](#).
- Ensure that the user account specified by `ObjectName` has read/write access to the configuration file.
- Create a new Windows event log by adding the registry key specified by `EventLog`.
- Add an event log source under `EventLog` for the source name specified by `Ice.EventLog.Source`. If this property is not defined, the service name is used as the source name.
- Install the service, including command line arguments that specify the service name (`--service name`) and the absolute path name of the configuration file (`--Ice.Config=config-file`).

The Ice service installer currently does *not* perform these tasks:

- Modify access rights for the service's executable or its DLL dependencies
- Verify that the user account specified by `ObjectName` has the right to "Log on as a service"

Uninstalling a Service

When uninstalling an existing service, the Ice service installer first ensures that the service is stopped, then proceeds to remove the service. The service's event log source is removed and, if the service is not using the `Application` log, the event log registry key is also removed.

See Also

- [icegridregistry](#)
- [icegridnode](#)
- [Getting Started with Glacier2](#)
- [IceBox](#)
- [IcePatch2](#)
- [Installing a Windows Service](#)
- [Manually Installing a Service](#)
- [Troubleshooting Windows Services](#)
- [IceGrid Properties](#)
- [Glacier2 Properties](#)