

# Server Descriptor

A server represents an Ice server deployed on a node as part of an application. IceGrid supports three kinds of servers:

- **Plain Server**  
A normal Ice server, with typically a single Ice communicator. IceGrid generates a single Ice configuration file for this server.
- **IceBox Server**  
An IceBox server, with usually a number of IceBox services deployed in it. IceGrid generates an Ice configuration file for the IceBox server itself, and also a separate Ice configuration for each IceBox service.
- **Server Instance**  
A server (either plain server or IceBox server) defined using a server template.

On this page:

- [Plain Server](#)
  - [Properties](#)
  - [Children](#)
- [IceBox Server](#)
  - [Properties](#)
  - [Children](#)
- [Server Instance](#)
  - [Properties](#)
  - [Children](#)

## Plain Server

### Properties

The Server Properties panel offers the following fields:

- **Server ID**  
The ID of the server; corresponds to the `Ice.Admin.ServerId` property. Each server must have a unique ID within an IceGrid deployment.
- **Description**  
A free-text description of this server.
- **Property Sets**  
List of property-set IDs; you refer to a [property set](#) to "include" all its properties in the server.
- **Properties**  
Ice properties private to this server.
- **Log Files**  
This table can be used to declare a number of log files used by this server. Path is the path to the log file (a relative path is relative to the IceGrid node working directory); when Property is set, IceGrid generates a property with this name and the log file path as value. You declare log files to be able to conveniently retrieve them using IceGrid Admin (in the [Live Deployment](#) tab) or with the `icegridadmin` command-line utility.
- **Path to Executable**  
Path to the server's executable; cannot be blank. A relative path is relative to the IceGrid node working directory.
- **Ice Version**  
The Ice version of this server. If you don't provide a value, IceGrid assumes it's the same version as the IceGrid registry.
- **Working Directory**  
The working directory for the server when started by the IceGrid node.
- **Command Arguments**  
The command-line arguments given to the server when started by the IceGrid node.
- **Run as**  
On Linux and Unix, when IceGrid node is running as root, it is possible to run the server under any username. Enter the desired username in this field.  
When not set (the default), the server runs as the same user as the IceGrid node, except when IceGrid node runs as root. In this case, the

server runs as nobody.

- **Environment Variables**

The environment variables for the server when started by the IceGrid node. These variables are in addition to variables defined in the IceGrid node own environment.

- **Activation Mode**

The server's activation mode. Must be one of:

Activation Mode	Description
always	IceGrid keeps this server running all the time
manual	This server is started "manually", using IceGrid Admin or the <code>icegridadmin</code> command-line utility
on-demand	IceGrid starts this server when it resolves the object-adapter ID of an object adapter defined in this server
session	IceGrid starts a separate instance of this server for each IceGrid session that allocates this server

The Activation Mode can also be a variable or a combination of variables that resolves to one of the values above.

- **Activation Timeout**

When activating a server, IceGrid gives timeout seconds to object adapters with server lifetime to register their endpoints with the IceGrid registry. During this time, lookup for the corresponding adapter IDs are delayed.

If not set or set to 0, the IceGrid node uses the value of its `IceGrid.Node.WaitTime` property.

- **Deactivation Timeout**

When deactivating a server, IceGrid gives timeout seconds to the server to exit gracefully. After this timeout, the server process is killed.

If not set or set to 0, the IceGrid node uses the value of its `IceGrid.Node.WaitTime` property.

- **Allocatable**

Specifies whether the server can be allocated. A server is allocated implicitly when one of its allocatable objects is allocated. This checkbox is ignored if the server activation mode is session; a server with this activation mode is always allocatable. Default: false.

- **Depends on the application distribution**

A server that depends on the application distribution is stopped and disabled before any application distribution update, and re-enabled after such update. Default: true.

- **IcePatch2 Proxy**

A stringified (or well-known) proxy for the IcePatch2 server than contains this server's private distribution. Possible values:

Value	Description
"None selected" (default) or blank	no private distribution defined
<code>\${application}.IcePatch2/server</code>	the well-known proxy of an IcePatch2 server deployed in this application using the IcePatch2 default server-template (with a default instance-name parameter value).
<i>stringified proxy</i>	your own stringified or well-known proxy

- **Directories**

List of directories included in the application distribution. When blank, the entire IcePatch2 server repository is used as the distribution.

## Children

A plain server can have two types of children:

- [Adapter](#)
- [Database Environment](#)

## IceBox Server

### Properties

The Properties panel for an IceBox server is identical to the Properties panel for a [Plain Server](#).

When you create a new IceBox server, some properties are created automatically:

Property Name	Property Value
IceBox.InstanceName	\${server}
Ice.Admin.Endpoints	tcp -h 127.0.0.1

The `Ice.Admin.Endpoints` setting enables the Admin object in the main communicator of this IceBox server.

The Path to Executable is typically `icebox`, `iceboxd` (for a C++ IceBox), `java` (for a Java IceBox) or `iceboxcs` (for a .NET IceBox). In Java, the Command Arguments should include the IceBox container class name, `IceBox.Server`.

## Children

An IceBox server can have only one type of children, [Service](#).

## Server Instance

A server instance is a server created from a server template; it may be a plain server or an IceBox server.

## Properties

The Server Instance Properties panel offers the following fields:

- **Template**  
The name of the [Server Template](#).
- **Parameters**  
Use this table to assign values to the template parameters defined in the server template.
- **Property Sets**  
List of property-set IDs; you refer to a [property set](#) to "include" all its properties in this server-instance.
- **Properties**  
Ice properties private to this server. Overall, the properties of the server instance are a combination of properties defined in the server template (including its own property sets references) augmented and possibly overridden by properties defined in the server instance.

## Children

An instance of an IceBox server template can have one type of children, [Property Set](#).

An instance of a plain server template cannot have any child.