Plug-in Configuration

Plug-ins are installed using a configuration property of the following form:

```
Ice.Plugin.Name=entry_point [arg ...]
```

In most cases you can assign an arbitrary name to a plug-in. In the case of IceSSL, however, the plug-in requires that its name be `IceSSL`.

The value of *`entry_point`* is a language-specific representation of the plug-in's factory. In C++, it consists of the path name of the shared library or DLL containing the factory function, along with the name of the factory function. In Java and .NET, the entry point specifies the factory class.

The language-specific nature of plug-in properties can present a problem when applications that are written in multiple implementation languages attempt to share a configuration file. Ice supports an alternate syntax for plug-in properties that alleviates this issue:

```
Ice.Plugin.name.cpp=...    # C++ plug-in
Ice.Plugin.name.java=...   # Java plug-in
Ice.Plugin.name.clr=...    # .NET (Common Language Runtime) plug-in
```

Plug-in properties having a suffix of `.cpp`, `.java`, or `.clr` are loaded only by the appropriate Ice run time and ignored by others.

After extracting the plug-in's entry point from the property value, any remaining text is parsed using semantics similar to that of command-line arguments. Whitespace separates the arguments, and any arguments that contain whitespace must be enclosed in quotes:

```
Ice.Plugin.MyPlugin=entry_point --load "C:\Data Files\config.dat"
```

Ice passes these arguments to the plug-in's entry point during loading.

See Also

- Ice Plug-In Properties
- IceSSL