The C++11 Ice String Converter Plug-in

Previous Next

The Ice run time includes a plug-in that supports conversion between UTF-8 and native encodings on Unix and Windows platforms. You can use this plug-in to install converters for narrow and wide strings into the communicator of an existing program. This feature is primarily intended for use in scripting language extensions such as Ice for Python; if you need to use string converters in your C++ application, we recommend using the technique described in Installing String Converters with C++11 instead.

Ice for Python uses the C++98 mapping, so in practice Ice for Python could only use the C++ 98 Ice String Converter Plug-in.

Note that an application must be designed to operate correctly in the presence of a string converter. A string converter assumes that it converts strings in the native encoding into the UTF-8 encoding, and vice versa. An application that performs its own conversions on strings that cross a Slice interface boundary can cause encoding errors when those strings are processed by a converter.

Configuring the Ice String Converter Plug-in

You can install the plug-in using a configuration property like the one shown below:

```
Ice.Plugin.IceStringConverter=Ice:createStringConverter iconv=encoding[,encoding] windows=code-page
```

The first component of the property value represents the plug-in's entry point, which includes the abbreviated name of the shared library or DLL (`Ice`) and the name of a factory function (`createStringConverter`).

The plug-in accepts the following arguments:

- `iconv=encoding[,encoding]`
  This argument is optional on Unix platforms and ignored on Windows platforms. If specified, it defines the `iconv` names of the narrow string encoding and the optional wide-string encoding. If this argument is not specified, the plug-in installs a narrow string converter that uses the default locale-dependent encoding.

- `windows=code-page`
  This argument is required on Windows platforms and ignored on Unix platforms. The *`code-page`* value represents a code page number, such as `1252`.

The plug-in's argument semantics are designed so that the same configuration property can be used on both Windows and Unix platforms, as shown in the following example:

```
Ice.Plugin.IceStringConverter=Ice:createStringConverter iconv=ISO8859-1 windows=1252
```

If the configuration file containing this property is shared by programs in multiple implementation languages, you can use an alternate syntax that is loaded only by the Ice for C++ run time:

```
Ice.Plugin.IceStringConverter.cpp=Ice:createStringConverter iconv=ISO8859-1 windows=1252
```

If using static libraries, you must also call the `Ice::registerIceStringConverter` function to ensure the plugin is linked with your application.

Back to Top ^

See Also

- UTF-8 Conversion with C++11
- Installing String Converters with C++11
- Plug-in Configuration
- Ice.InitPlugins
- Ice.Plugin.*
- Ice.PluginLoadOrder

Previous Next