

# The iconv String Converter

For Linux and Unix platforms, Ice provides an `IconvStringConverter` template class that uses the `iconv` conversion facility to convert between the native encoding and UTF-8. The only member function of interest is the constructor:

**C++**

```
template<typename charT>
class IconvStringConverter : public Ice::BasicStringConverter<charT>
{
public:
    IconvStringConverter(const char* = nl_langinfo(CODESET));

    // ...
};
```

To use this string converter, you specify whether the conversion you want is for narrow or wide characters via the template argument, and you specify the corresponding native encoding with the constructor argument. For example, to create a converter that converts between ISO Latin-1 and UTF-8, you can instantiate the converter as follows:

**C++**

```
InitializationData id;
id.stringConverter = new IconvStringConverter<char>("ISO-8859-1");
```

Similarly, to convert between the internal wide character encoding and UTF-8, you can instantiate a converter as follows:

**C++**

```
InitializationData id;
id.stringConverter = new IconvStringConverter<wchar_t>("WCHAR_T");
```

The string you pass to the constructor must be one of the values returned by `iconv -l`, which lists all the available character encodings for your machine.

Using the `IconvStringConverter` template makes it easy to install code converters for any available encoding without having to explicitly write (or call) conversion routines, whose implementation is typically non-trivial.

## See Also

- [Installing String Converters](#)