

String Parameters in Local Calls

In C++, and indirectly in Python, Ruby, and PHP, all Ice local APIs are narrow-string based, meaning you could not for example recompile `Properties.ice` to get property names and values as wide strings.

Installing a narrow-string converter could cause trouble for these local calls if UTF-8 conversion occurs in the underlying implementation. For example, the [stringToIdentity](#) operation creates an intermediary UTF-8 string. If this string contains characters that are not in your native codeset (as determined by the narrow-string converter), the `stringToIdentity` call will fail.

Likewise, when Ice reads [properties](#) from a configuration file, it converts the input (UTF-8 characters) into native strings. This conversion can also fail if the native encoding cannot convert some characters.

Most strings in local calls are never problematic because Ice does not perform any conversion, for example:

- adapter names in `createObjectAdapter`
- property names and values in `Properties`
- `ObjectAdapter::createProxy`, where the identity conversion occurs only when the proxy is marshaled

Finally, consider the Slice type [Ice::Context](#), which is mapped in C++ as a `map<string, string>`. The mapping for `Context` cannot be changed to `map<wstring, wstring>`, therefore you cannot send or receive any context entry that is not in your narrow-string native encoding when a narrow-string converter is installed.

See Also

- [Object Identity](#)
- [Properties and Configuration](#)
- [Request Contexts](#)