

Protocol and Encoding Versions

On this page:

- [Version Flexibility](#)
- [Version Ground Rules](#)

Version Flexibility

As we saw in the preceding sections, both the Ice protocol and encoding have separate major and minor version numbers. Separate versioning of protocol and encoding has the advantage that neither depends on the other: any version of the Ice protocol can be used with any version of the encoding, so they can evolve independently. For example, you could send a request with Ice protocol version 1.1, and encode the parameters with encoding version 2.3.

The Ice versioning mechanism provides the maximum possible amount of interoperability between clients and servers that use different versions of the Ice run time. In particular, older deployed clients can communicate with newer deployed servers and vice versa, provided that the message contents use types that are understandable to both sides. Intermediary servers, such as a Glacier2 router, do not and should not check the contents of the requests and responses they forward; this way, an older intermediary server can forward messages encoded with a newer encoding that it does not understand.



Due to a bug, a Glacier2 router prior to version 3.5 can only forward messages encoded using the 1.0 encoding.

Ice 3.5 introduced a new encoding, encoding version 1.1, in support of several new features such as the ability to define optional parameters in operations. In order to send a request with an optional parameter, you need to use encoding 1.1 or greater.

Ice 3.5 is capable of marshaling and unmarshaling messages with both the 1.0 and 1.1 encodings (it uses the 1.1 encoding by default). This allows interoperability to the maximum extent possible:

Client Ice Version	Server Ice Version	Encoding Version	Operation without Optional Parameter	Operation with Optional Parameter
3.4 or earlier	3.4 or earlier	1.0	Yes	No
3.4 or earlier	3.5	1.0	Yes	No
3.5	3.4 or earlier	1.0	Yes	No
3.5	3.5	1.1	Yes	Yes

When an Ice 3.5 client sends a request to an Ice 3.4 server, it must naturally use the 1.0 encoding. This is achieved by setting correctly (and often automatically) the encoding version on the proxy used. If this proxy is received by the client through a response using the 1.0 encoding, the decoded proxy will automatically have the desired 1.0 encoding (all proxies in a 1.0-encoded message implicitly have 1.0 as their encoding version and 1.0 as their protocol version). If this proxy is created with `stringToProxy` or `propertyToProxy`, the client will need to set this proxy's encoding version to 1.0 through one of the following methods:

- with `-e 1.0` in the source stringified proxy
- by creating a new proxy with the desired encoding version, by calling `ice_encodingVersion("1.0")` on the proxy
- by setting `Ice.Default.EncodingVersion` to 1.0.

Version Ground Rules

For versioning of the protocol and encoding to be possible, all versions (present and future) of the Ice run time adhere to a few ground rules:

1. [Encapsulations](#) always have a six-byte header; the first four bytes are the size of the encapsulation (including the size of the header), followed by two bytes that indicate the major and minor version. How to interpret the remainder of the encapsulation depends on the major and minor version.
2. The first eight bytes of a [message header](#) always contain the magic number 'I', 'c', 'e', 'P', followed by four bytes of version information (two bytes for the protocol major and minor number, and two bytes for the encoding major and minor number). How to interpret the remainder of the header and the message body depends on the major and minor version.

These ground rules ensure that all current and future versions of the Ice run time can at least identify the version and size of an encapsulation and a message. This is particularly important for message switches such as [IceStorm](#); by keeping the version and size information in a fixed format, it is possible to forward messages that are, for example, at version 2.0, even though the message switch itself may still be at version 1.0.

See Also

- [Basic Data Encoding](#)

- [Protocol Messages](#)
- [IceStorm](#)
- [IceGrid](#)