

Client-to-Client Protocol

Changelog

Additions from the previous version are ***bolded and italicized***

Deletions from the previous version are ~~struck-through~~

Overview

The client-to-client protocol is a secondary and independent protocol from the main server protocol. The client-to-client protocol is used to communicate special ***client-side*** commands that don't need ***additional*** interference from the server ~~between clients~~. ***That is, all client messages will be sent through the server protocol, but some client-side commands will be sent through the server as specially encoded messages rather than as server-side commands. This allows for development of client-side commands without requiring an update to the server or server protocol.***

Different implementations of the AceChat chat client may choose to implement this protocol differently, but the input and output ~~should~~ ***must*** be the same to allow for maximum cross-compatibility ***between clients***.

Encoded messages take the form:

`\protocolxxx<message>\xxx`

where xxx is the three digit hexadecimal protocol number.

Currently Supported Encodings

/me

User input in the form

`/me <message>`

is encoded as

`\protocol000<message>\000`

and is displayed in the channel in the form

`*<user> <message>*`

/party

User input in the form

`/party <message>`

is encoded as

`\protocol001<message>\001`

and is displayed in the channel in the form of a party. That is, each letter is a random color.

Unencoded Messages

Unencoded messages are displayed ~~as expected~~ *in the format that was chosen to display messages from the server, without any additional formatting. This format may vary between different implementations of the client.* The protocol only applies to messages that are wrapped in a protocol layer. Messages wrapped in an invalid or unsupported protocol layer ~~should~~ *will* either throw an error or be displayed with the protocol wrapper, at the client's discretion.