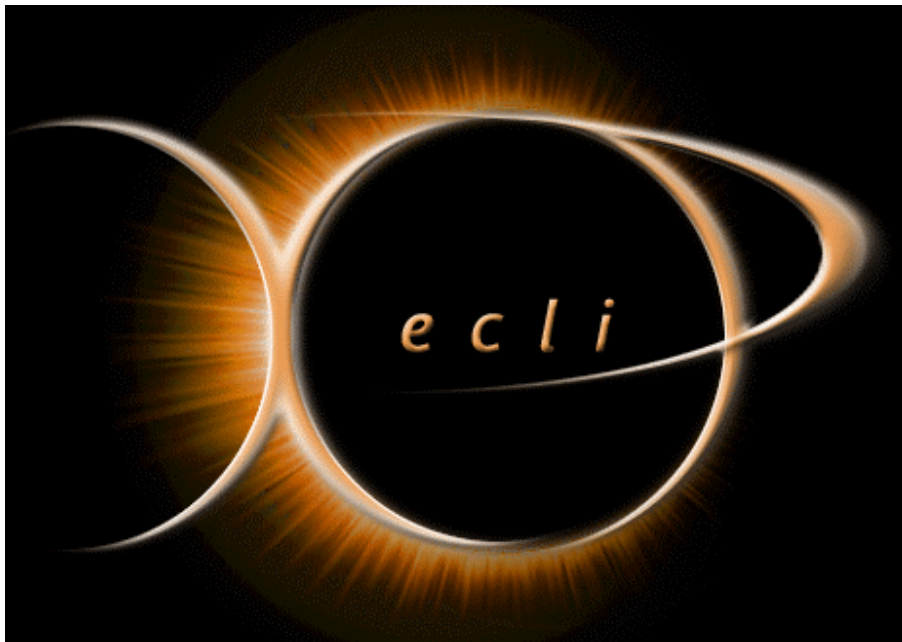


X e c l i P



XecliP Protocol

1 Index of content

1	Index of content.....	2
2	General protocol regulation.....	5
3	Message Types	6
3.1	User LOGIN / LOGOUT	6
3.1.1	LOGIN (Client to Server)	6
3.1.2	LOGIN_SUCCESS (Server to Client)	6
3.1.3	LOGIN_ERROR (Server to Client)	6
3.1.4	LOGOUT (Client to server)	7
3.2	USER_UPDATE (Server to Client).....	7
3.3	USER_STATUS (CLIENT to Server to Client)	7
3.4	SESSION_UPDATE (Server to Client)	8
3.5	UserProfile	8
3.5.1	GET_USERPROFILE_CANCEL (Client to Server).....	8
3.5.2	GET_USERPROFILE (Client to Server).....	8
3.5.3	GET_USERPROFILE_RESPONSE (Server to Client).....	9
3.5.4	PUT_USERPROFILE (Client to Server).....	9
3.5.5	QUERY_USERPROFILE (Client to Server).....	9
3.5.6	QUERY_USERPROFILE_RESPONSE (Server to Client).....	10
3.6	Create a New Session	11
3.6.1	INVITE (Client to Server).....	11
3.6.2	INVITE_SUCCESS (Server to Client)	11
3.6.3	INVITE_ERROR (Server to Client)	11
3.6.4	INVITE_CANCEL (Client to Server to Client).....	12
3.6.5	INVITATION (Server to Client).....	12
3.6.6	INVITATION_ACCEPT (Client to Server to Client)	13
3.6.7	INVITATION_DENY (Client to Server to Client).....	13
3.6.8	SESSION_CREATE (Client to Server to Client or Server to Client for visitor) ...	14
3.6.9	SESSION_READY (Client to Server to Client or Client to Server for visitor)	14
3.7	Messages while Session is running	15
3.7.1	Editor messages	15
3.7.1.1	EDITOR_OPEN (Client to Server to Client)	15
3.7.1.2	EDITOR_OPEN_RESPOND (Client to Server to Client).....	15
3.7.1.3	EDITOR_TEXT_CHANGED (Client to Server to Client).....	16
3.7.1.4	EDITOR_CURSOR_CHANGED (Client to Server to Client).....	16
3.7.1.5	EDITOR_SELECTION_CHANGED (Client to Server to Client)	17
3.7.1.6	EDITOR_ANNOTATION_CHANGED (Client to Server to Client)	17
3.7.1.7	EDITOR_FOCUS (Client to Server to Client)	18
3.7.1.8	EDITOR_VIEWPORT_CHANGED (Client to Server to Client)	18
3.7.1.9	EDITOR_SAVE (Client to Server to Client)	19
3.7.1.10	EDITOR_CLOSE (Client to Server to Client).....	19

3.7.2	Switch role of navigator / driver	20
3.7.2.1	SWITCH_REQUEST (Client to Server to Client).....	20
3.7.2.2	SWITCH_CANCEL (Client to Server to Client).....	20
3.7.2.3	SWITCH_ACCEPT (Client to Server to Client)	21
3.7.3	Speech Request and Speech Answer.....	22
3.7.3.1	Speech Request (Client to Server to Client).....	22
3.7.3.2	Speech Answer (Client to Server to Client)	22
3.7.4	Visitors and Session	23
3.7.4.1	SESSION_JOIN (Client to Server to Client)	23
3.7.4.2	SESSION_JOIN_ACCEPT (Client to Server to Client).....	23
3.7.4.3	SESSION_JOIN_DENY (Client to Server to Client).....	24
3.7.4.4	SESSION_READY (see Create a session)	24
3.7.4.5	SESSION_JOIN_DATA (Server to Client).....	24
3.7.4.6	SESSION_JOIN_CANCEL (Client to Server to Client)	25
3.7.4.7	SESSION_JOIN_LEAVE (Client to Server to Client)	25
3.7.5	Chat messages	26
3.7.5.1	CHAT_MESSAGE (Client to Server)	26
3.7.5.2	CHAT_UPDATE (Server to Client)	26
3.7.5.3	CHAT_NOTIFY (Client to Server to Client)	26
3.8	The End of a Session	27
3.8.1	SESSION_CLOSE (Client to Server).....	27
3.8.2	SESSION_FINISHED (Server to Client)	27
3.9	Playback a Session	28
3.9.1	SESSIONLOG_QUERY (Client to Server).....	28
3.9.2	SESSIONLOG_QUERY_RESPOND (Server to Client)	28
3.9.3	SESSIONLOG_GET (Client to Server).....	29
3.9.4	SESSIONLOG_RESPOND (Server to Client)	29
3.10	ERROR.....	29

<p>Published by</p>	<p>XecliP Team</p> <p>Fachpraktikum CSCW term 2006/2007</p> <p>at Fernuniversität in Hagen (Germany) faculty of Informatik Juniorprofessur für Verteilte Systeme für kooperative Arbeits-/Lernumgebungen</p>
<p>Authors and developers of the XecliP plug-in</p>	<p>Angela Leberfinger, Karin Paland, Franka Schnarrenberger, Rainer Gloede, Tobias Harmes, Dirk Jablonski, Dominik Kröger</p>

2 General protocol regulation

1. The general structure of the message:

message-type (message-parameter, message-parameter, ...) \r\n

message-type:

LOGIN | LOGOUT | INVITE | INVITE_ACCEPT | INVITE_DENY | INVITATION ...

(For message type see next paragraph)

message-parameter (enclosed by round brackets):

(parameter-name ':' parameter-value, ...)

Example for message of message type LOGIN:

LOGIN(username::cmFpbmVy,password::cnVkZXJib290)\r\n

2. Each message must have a message type.
3. A message can but must not contain pairs of parameter name and values.
4. For messages without parameters the round brackets can be omitted.
5. The sequence of those parameters is not predetermined. But the parameter names in on message must be unique. So the values to a parameter name can be identified.
6. The Type of a Message should be build up from capitals.
7. The parameter name is case sensitive. In general a parameter name should be based on low case letters.
8. The message structure is used fore each data exchange between clients an server. e.g. error messages are built in the same way.
9. The parameter values are base64 coded.
10. Each row of data has to be finalized with CRLF. In case of java application the string "\r\n" is used usually.

(The XecliP server runs by default on port 2631. The port seems not to be used by any other application. -> see: <http://www.auditmypc.com/port/tcp-port-2631.asp>)

3 Message Types

3.1 User LOGIN / LOGOUT

3.1.1 LOGIN (Client to Server)

<i>Message Type</i>	<i>Parameter</i>	<i>Comment</i>
LOGIN		first login at the server
	username	user name
	password	password of the user

3.1.2 LOGIN_SUCCESS (Server to Client)

<i>Message Type</i>	<i>Parameter</i>	<i>Comment</i>
LOGIN_SUCCESS		<p>Answer from server that the user login in was accepted.</p> <p>After login the server pushes all new data that the client need on first log in. e.g all users that are present at the server and all running sessions,....</p> <p>(see: USER_UPDATE, SESSON_UPDATE)</p>

3.1.3 LOGIN_ERROR (Server to Client)

Message Type	Parameter	Comment
LOGIN_ERROR	message	Cause of why login was deneyed. (e.g. wrong password)

3.1.4 LOGOUT (Client to server)

Message Type	Parameter	Comment
LOGOUT		user will be disconnected from the server Implicit a USER_UPDATE is send to all other clients which are logged in at the server (see: USER_UPDATE)

3.2 USER_UPDATE (Server to Client)

Message Type	Parameter	Comment
USER_UPDATE		This message is send by the server if the status of the user changes. Especially when a user loges in or logs out at the server.
	user	String representation of the user Object.

3.3 USER_STATUS (CLIENT to Server to Client)

Message Type	Parameter	Comment
USER_STATUS		This message is created by the client, when the user status is toggled between status "not available" and "online".
	user	Name of the user where the status is toggled.
	userstatus	New status of the user.

3.4 *SESSION_UPDATE (Server to Client)*

Message Type	Parameter	Comment
SESSION_UPDATE		This message is send to all clients registered at the server if the data of session is changed. Especially when a new session is created on the server, if the session participant's changes or a switch role is performed on a session.
	session	String representation of the session object.

3.5 *UserProfile*

3.5.1 *GET_USERPROFILE_CANCEL (Client to Server)*

Message Type	Parameter	Comment
GET_USERPROFIL E_CANCEL		This message is send by a client, if a respond from a GET_USERPROFILE message should be cancelled. That is the case if the user cancels the progress monitor, while the data are transfered form server to the client.

3.5.2 *GET_USERPROFILE (Client to Server)*

Message Type	Parameter	Comment
GET_USERPROFILE		New message is sent to server to ask for a UserProfile. If a user is looking for a special UserProfile, he will send a GetUserProfileMessage, which contains the name of the user from which he want the profile.
	userprofile name	name of the user

3.5.3 GET_USERPROFILE_RESPONSE (Server to Client)

Message Type	Parameter	Comment
GET_USERPROFILE_RESPONSE		This is the response message of GET_USERPROFILE message. The server will send the UserProfile to the requesting user.
	profiledata	userProfile object

3.5.4 PUT_USERPROFILE (Client to Server)

Message Type	Parameter	Comment
PUT_USERPROFILE		Message send to server to store the Userprofile. If a user creates a UserProfile, this message will send all made input to the server to store it.
	profiledata	userProfile of the sending user

3.5.5 QUERY_USERPROFILE (Client to Server)

Message Type	Parameter	Comment
QUERY_USERPROFILE		Message send to the server to get informations about which userprofiles matches the query.
	skills	skillsData chosen skills/abilities
	project	a projectname
	type	searchType searching for skills, projectname or both

3.5.6 QUERY_USERPROFILE_RESPONSE (Server to Client)

Message Type	Parameter	Comment
QUERY_ USERPROFILE_ RESPONSE		Message send back from server to requesting client, containing the profiles, which matches the query made before.
	profiledata	profileData containing all UserProfiles, which fit the search made at the SessionWizard

3.6 Create a New Session

3.6.1 INVITE (Client to Server)

Message Type	Parameter	Comment
INVITE		User (inviter) wants to invite an other user to a new session
	invitee	Name of the user which should be invited (invitee).
	title	Title of the session.
	desc	Description of the session.
	withoutaccept	String value which marks, if a visitor can join the session with ("1") or without ("2") acceptance by the driver.
	recording	String value which marks, if the session is recorded ("1") or not ("2").

3.6.2 INVITE_SUCCESS (Server to Client)

Message Type	Parameter	Comment
INVITE_SUCCESS		The invitation was delivered successfully to the invitee.

3.6.3 INVITE_ERROR (Server to Client)

Message Type	Parameter	Comment
INVITE_ERROR	message	The invitation could not be delivered to the invitee. (possible reason: user is offline)

3.6.4 INVITE_CANCEL (Client to Server to Client)

Message Type	Parameter	Comment
INVITE_CANCEL		The inviter can cancel the invitation while the invitation is in process or waiting on the invitee side. This message informs the server about the cancel and forwards the cancel message to the invitee also. By that way the original invitation is cancelled on the invitee side too.
	requester	Name of the user, which initiated the see invitation.
	title	Title of the session.

3.6.5 INVITATION (Server to Client)

Message Type	Parameter	Comment
INVITATION		After receiving a INVITE message from the inviter, the server creates this message to notify the invitee that he is invited to a new session.
	requester	Name of the user which is the inviter.
	title	Title of the session.
	desc	Description of the session.
	withoutaccept	String value which marks, if a visitor can join the session with ("1") or without ("2") acceptance by the driver.
	recording	String value which marks, if the session is recorded ("1") or not ("2").

3.6.6 INVITATION_ACCEPT (Client to Server to Client)

Message Type	Parameter	Comment
INVITATION_ACCEPT		If the invitee has accepted the invitation. The message is created and is send to the server. The server forwards this message to the inviter client.
	comment	Additional information from the invitee for the inviter. Comment may be blank.
	session	Title of the session.
	requester	User name of the inviter.

Remark:

The Server now change the status of the two Users to "in session".
After INVITATION_ACCEPT the inviter client, answers with a SESSION_CREATE message.

3.6.7 INVITATION_DENY (Client to Server to Client)

Message Type	Parameter	Comment
INVITATION_DENY		If the invitee has not accepted the invitation. The message is created and is send to the server. The server forwards this message to the inviter client.
	comment	Additional information from the invitee for the inviter. Comment may be blank.
	session	Title of the session.
	requester	User name of the inviter.

3.6.8 SESSION_CREATE (Client to Server to Client or Server to Client for visitor)

Message Type	Parameter	Comment
SESSION_CREATE		If an invitation is accepted by the invitee. The inviter client identifies all project data and sends this message to the invitee, to synchronic the workspace.
	sessionname	Title of the session.
	sessiondata	Project data to synchronise the workspace.

Remark:

The server sends to all clients a SESSION_UPDATE message for the new created session.

3.6.9 SESSION_READY (Client to Server to Client or Client to Server for visitor)

Message Type	Parameter	Comment
SESSION_READY		Message for notify the driver or client, that the session was successfully initiated at navigator / visitor side.
	sessionname	Title of the session.
	userName	Can be the navigator name or the name of the visitor which will participate on the session.

3.7 Messages while Session is running

3.7.1 Editor messages

3.7.1.1 EDITOR_OPEN (Client to Server to Client)

Message Type	Parameter	Comment
EDITOR_OPEN		Message is created on opening of an editor on the driver side. Message is only created for files within the shared session project.
	sessionname	Title of the session.
	username	User which opened the editor. (driver)
	filename	Path name of the file.
	text	Initial text content of the editor.

3.7.1.2 EDITOR_OPEN_RESPOND (Client to Server to Client)

Message Type	Parameter	Comment
EDITOR_OPEN_RESPOND		Message is created when a navigator / visitor opens a new editor window. Message is only created for files within the shared session project.
	sessionname	Title of the session.
	username	User which send the respond.
	filename	Path name of the file.

Remark:

Cause for this message can be that the navigator / driver open an editor window by themselves. At first this messages is recognised by the server.

This message is also send as respond when an editor is opened by an EDITOR_OPEN message. When the server has received all respond messages from all XP-partners of a session, this message is forwarded to the driver. The driver has to wait for this message from the server, till all editor windows are open.

3.7.1.3 EDITOR_TEXT_CHANGED (Client to Server to Client)

Message Type	Parameter	Comment
EDITOR_TEXT_CHANGED		Message is created when the text of the editor content is changed. Message is only created for files within the shared session project.
	sessionname	Title of the session.
	username	User which made the text change. (driver)
	filename	Path name of the file.
	text	New inserted text.
	offset	Offset of the document where the text change occurred.
	length	Length of the text which is replaced.
	replace	Text of the document which is replaced by this operation.

3.7.1.4 EDITOR_CURSOR_CHANGED (Client to Server to Client)

Message Type	Parameter	Comment
EDITOR_CURSOR_CHANGED		Message is created when the position of the cursor is changed in the editor window. Message is only created for files within the shared session project.
	sessionname	Title of the session.
	username	User which made the cursor change. (driver)
	filename	Path name of the file.
	cursorposition	Position of the cursor related to the editor document.

3.7.1.5 EDITOR_SELECTION_CHANGED (Client to Server to Client)

Message Type	Parameter	Comment
EDITOR_SELECTION_CHANGED		Message is created when a part of the text is marked for selection. Message is only created for files within the shared session project.
	sessionname	Title of the session.
	username	User which performed the selection event. (may be driver, navigator, visitor)
	filename	Path name of the file.
	offset	Offset selected text.
	length	Length of selected text.

3.7.1.6 EDITOR_ANNOTATION_CHANGED (Client to Server to Client)

Message Type	Parameter	Comment
EDITOR_ANNOTATION_CHANGED		Message is created on folding event. The folding event can collapse or expand an editor region. Message is only created for files within the shared session project.
	sessionname	Title of the session.
	username	User which performed the folding event.
	filename	Path name of the file.
	offset	Offset of the region.
	length	Length of the region.
	collapsed	String value "0" for not collapsed region / "1" for collapsed region.

3.7.1.7 EDITOR_FOCUS (Client to Server to Client)

Message Type	Parameter	Comment
EDITOR_FOCUS		Message is created when a new editor window gets the focus on the driver side. Message is only created for files within the shared session project.
	sessionname	Title of the session.
	username	User where the focus event was initiated. (driver)
	filename	Path name of the file.

3.7.1.8 EDITOR_VIEWPORT_CHANGED (Client to Server to Client)

Message Type	Parameter	Comment
EDITOR_VIEWPORT_CHANGED		Message is created when the visible region of the editor changes. (e.g. scrolling up or down) Message is only created for files within the shared session project.
	sessionname	Title of the session.
	username	User which made the view port change.
	filename	Path name of the file.
	line	Line number of the first line (TopIndex) of the visible region.

3.7.1.9 EDITOR_SAVE (Client to Server to Client)

Message Type	Parameter	Comment
EDITOR_SAVE		Message is created when the editor content is saved. Message is only created for files within the shared session project.
	sessionname	Title of the session.
	username	User where the save event was performed. (driver)
	filename	Path name of the file.

3.7.1.10 EDITOR_CLOSE (Client to Server to Client)

Message Type	Parameter	Comment
EDITOR_CLOSE		Message is created when the editor window is closed. Message is only created for files within the shared session project.
	sessionname	Title of the session.
	username	User which closed the editor window.
	filename	Path name of the file.
	saveit	String value "1" marks that the editor content was saved on closing event. Value "0" marks that the editor window was closed without saving the content.

3.7.2 Switch role of navigator / driver

3.7.2.1 SWITCH_REQUEST (Client to Server to Client)

Message Type	Parameter	Comment
SWITCH_REQUEST		Request to change the role of the navigator to driver and to change the role of the driver to navigator. Both, the driver or the navigator can request a switch role.
	sessionname	Title of the session.
	username	User who has initiated the switch request.

3.7.2.2 SWITCH_CANCEL (Client to Server to Client)

Message Type	Parameter	Comment
SWITCH_CANCEL		The user (driver or navigator) can cancel his own request to switch the role by this message.
	sessionname	Title of the session.
	username	User who has initiated the switch request and now perform the cancel.

3.7.2.3 SWITCH_ACCEPT (Client to Server to Client)

Message Type	Parameter	Comment
SWITCH_ACCEPT		Driver or navigator can accepts the switch role request. By accepting the switch request the role of the navigator changes to driver and the role of the driver changes to navigator.
	sessionname	Title of the session.
	username	User who accepted the switch request.
	requester	User who has initiated the switch request.

Remark:

After SWITCH_ACCEPT a SESSION_UPDATE message is send by the server, to make the new role of the users in the session transparent for other users.

3.7.3 Speech Request and Speech Answer

3.7.3.1 Speech Request (Client to Server to Client)

Message Type	Parameter	Comment
SPEECH_REQUEST		Message is used to inform the driver about a speech request and its reset respectively by a visitor in the driver/navigator chat.
	username	name of the requester
	type	Message type: "request" - "cancel" - "reject"

3.7.3.2 Speech Answer (Client to Server to Client)

Message Type	Parameter	Comment
SPEECH_ANSWER		Answer to speech request message
	requester	The username who has initialized the request
	type	Message type: "accept" - "cancel" - "revoke"

3.7.4 Visitors and Session

General Remark:

A Session can be defined in that way that a visitor can join the session immediately or only after acceptance from driver.

3.7.4.1 SESSION_JOIN (Client to Server to Client)

Message Type	Parameter	Comment
SESSION_JOIN		If a visitor wants to join a session this message is send to the server.
	sessionname	Title of the session.
	visitorname	Name of the visitor who wants to join the session.

Remark:

If the session needs the acceptance from the driver, this message is send to client of the driver and also to the client of the navigator for information.

When the visitor can join the session immediately the server sends the session CREATE_MESSAGE to the visitor client to set up the workspace with the initial data.

3.7.4.2 SESSION_JOIN_ACCEPT (Client to Server to Client)

Message Type	Parameter	Comment
SESSION_JOIN_ACCEPT		The message is created when the driver of a session accepts the JOIN_REQUEST of the visitor.
	sessionname	Title of the session.
	requester	Name of the visitor who wants to join the session.
	comment	Info text from the driver to be shown on visitor side.

Remark:

After SESSION_JOIN_ACCEPT the server sends the session CREATE_MESSAGE to the visitor client to set up the workspace with the initial data.

3.7.4.3 SESSION_JOIN_DENY (Client to Server to Client)

Message Type	Parameter	Comment
SESSION_JOIN_DENY		The message is created when the driver of a session denies the JOIN_REQUEST of the visitor.
	sessionname	Title of the session.
	requester	Name of the visitor who wants to join the session.
	comment	Info text from the driver to be shown on visitor side.

3.7.4.4 SESSION_READY (see Create a session)

Remark:

When the visitor's workspace is set up with the initial data from the SESSION_CREATE message, a SESSION_READY message is send from the visitor's client to the server.

3.7.4.5 SESSION_JOIN_DATA (Server to Client)

Message Type	Parameter	Comment
SESSION_JOIN_DATA		This message contains the changed data to bring the visitors workspace up to date.
	sessionname	Title of the session.
	visitorname	Name of the visitor who wants to join the session.
	editorfinalmap	String representation of a HashMap for changed editor content.
	openeditormap	String representation of a HashMap for open editors.

Remark:

When the server receives the SESSION_READY message from the client, the server responds with this message. From that time when the SESSION_JOIN_DATA message is send to the client, the visitor is officially part of the session. The server creates and sends a SESSION_UPDATE message to all Users.

3.7.4.6 SESSION_JOIN_CANCEL (Client to Server to Client)

Message Type	Parameter	Comment
SESSION_JOIN_CANCEL		If anything goes wrong before the visitor could enter the session. This message is created to abort a session SESSION_JOIN_REQUEST.
	sessionname	Title of the session.
	visitorname	Name of the visitor who wants to join the session.

Remark:

This message is forwarded by the server to the driver and navigator, to delete a maybe pending JOIN_REQUEST.

3.7.4.7 SESSION_JOIN_LEAVE (Client to Server to Client)

Message Type	Parameter	Comment
SESSION_JOIN_LEAVE		Message is created if a visitor wants to leave the session.
	sessionname	Title of the session.
	visitorname	Name of the visitor who wants to leave the session.

Remark:

When a visitor leaves a session the server creates a SESSON_UPDATE message.

3.7.5 Chat messages

3.7.5.1 CHAT_MESSAGE (Client to Server)

Message Type	Parameter	Comment
CHAT_MESSAGE		Message from a client about a new chat message which was entered by the user while a session is running.
	message	message text
	type	String variable „v“ to identify messages of visitor-chat view or „dn“ to identify messages of driver/navigator chat view.

3.7.5.2 CHAT_UPDATE (Server to Client)

Message Type	Parameter	Comment
CHAT_UPDATE		After receiving a CAHT_MESSAGE the server forwards the CAHT-UPDATE message to all participants of a session.
	message	message text
	type	String variable „v“ to identify messages of visitor-chat view or „dn“ to identify messages of driver/navigator chat view.
	timestamp	timestamp (server)
	username	Name of the user which has send the original CAHT_MESSAGE.

3.7.5.3 CHAT_NOTIFY (Client to Server to Client)

Message Type	Parameter	Comment
CHAT_NOTIFY		Message is send by the client if a user enters something in the chat input field at the chat view.
	username	User name of the user who is writing.

3.8 The End of a Session

3.8.1 SESSION_CLOSE (Client to Server)

Message Type	Parameter	Comment
SESSION_CLOSE		Message is created if the driver or navigator of the session wants to close the session.
	sessionname	Title of the session.

Remark:

When the server receives a SESSION_CLOSE message, a SESSION_FINISHED message is send to all users. If the session is marked as to be recorded, the whole session log of this session is saved for later playback.

3.8.2 SESSION_FINISHED (Server to Client)

Message Type	Parameter	Comment
SESSION_FINISHED		Server notifies the clients about a session close.
	sessionname	Title of the session.
	drivername	Name of the driver.
	navigatorname	Name of the navigator.

3.9 Playback a Session

3.9.1 SESSIONLOG_QUERY (Client to Server)

Message Type	Parameter	Comment
SESSIONLOG_QUERY		Message is used to request all available session for playback, which are stored at the server. The parameter of the message can be used to search only for special user, sessions and to restrict the number of session which may be responded.
	sessionName	Title of the session.
	userName	Search for sessions where the user was driver or navigator.
	dateFrom	All searched session must be equal or younger than the data from.
	dateTo	All searched session must be equal or older than the data to.

3.9.2 SESSIONLOG_QUERY_RESPOND (Server to Client)

Message Type	Parameter	Comment
SESSIONLOG_QUERY_RESPOND		Messages respond to a SESSIONLOG_QUERY message, containing the result of the query.
	listofsessions	String representation of a HashMap containing timestamps and the corresponding sessions. (HashMap<GregorianCalendar, String>)

3.9.3 SESSIONLOG_GET (Client to Server)

Message Type	Parameter	Comment
SESSIONLOG_GET		Require session log form server for session playback.
	sessionname	Title of the session.
	timestamp	Timestamp of session recording.

3.9.4 SESSIONLOG_RESPOND (Server to Client)

Message Type	Parameter	Comment
SESSIONLOG_RESPOND		Messages respond to a SESSIONLOG_GET message, containing a list of all messages from the session log.
	messagelist	String representation of a list containing string representation of all message from the session log. (LinkedList<String>)

3.10 ERROR

Message Type	Parameter	Comment
ERROR		Message in case of any unpredictable errors. Message is used to notifying the client user.
	message	Error message text.