FANDOM

GAMES

ANIME

MOVIES

TV

VIDEO

WIKIS

START A
WIKI

ADVERTISEMENT

Hotline Wiki

**20
PAGES**

EXPLORE   **WIKI CONTENT**   **COMMUNITY**

in: Development

Protocol   EDIT

watch 01:25
Jurassic World: Dominion Dominates Fandom Wikis - The Loop

Do you like this video?

Play Sound

The following is based on the official Word document from Hotsprings that describes the Hotline Network Protocol 1.9.

You might also want to take a look at Virtual1's Hotline Server Protocol Guide, a reverse-engineered documentation of the protocol that was used for development of HotSocket, a REALbasic socket class.

You can find some aspects of the protocol explained in further detail in other development pages.

Contents

1. Protocol Overview
2. Session Initialization
3. Transactions
4. Transaction Types (with Type ID)
   4.1. User Login and Management
   4.2. Chat Transactions

Protocol Overview

Hotline client is an application executing on the user's computer, and providing user interface for end-user services (chat, messaging, file services and other). Hotline server provides services and facilitates communication between all clients that are currently connected to it. Tracker application stores the list of servers that register with it, and provides that list to clients that request it. All these applications use TCP/IP for communication.

To be able to connect to the specific server, IP address and port number must be provided to the client application. If client receives server's address from a tracker, the tracker will provide the client with complete address. Otherwise, the user of Hotline client software must manually set this address. IP port number, set in the Hotline client for a specific server, is called *base port number*. Additional port numbers utilized by the network protocol are determined by using this base port number. Namely, the base port number itself is used for regular transactions, while base port + 1 is used when upload/download is requested. HTTP tunneling uses base port + 2 for the regular transactions, and base + 3 for uploads/downloads.

Numeric data transmitted over the wire is always in the network byte order.

Session Initialization

After establishing TCP connection, both client and server start the handshake process in order to confirm that each of them comply with requirements of the other. The information provided in this initial data exchange identifies protocols, and their versions, used in the communication. In the case where, after inspection, the capabilities of one of the subjects do not comply with the

requirements of the other, the connection is dropped. The following information is sent to the server:

Description	Size	Data	Note
Protocol ID	4	'TRTP'	0x54525450
Sub-protocol ID	4		User defined
Version	2	1	Currently 1
Sub-version	2		User defined

The server replies with the following:

Description	Size	Data	Note
Protocol ID	4	'TRTP'	
Error code	4		Error code returned by the server (0 = no error)

In the case of an error, client and server close the connection.

Transactions

After the initial handshake, client and server communicate over the connection by sending and receiving *transactions*. Every transaction contains description (request) and/or status (reply) of the operation that is performed, and parameters used for that specific operation. A transaction begins with the following header:

Description	Size	Data	Note
Flags	1	0	Reserved (should be 0)
Is reply	1	0 or 1	Request (0) or reply (1)
Type	2		Requested operation (user defined). 0 for reply
ID	4	Not 0	Unique transaction ID (must not be 0)
Error code	4		Used in the reply (user defined, 0 = no error)
Total size	4		Total data size for the transaction (all parts)
Data size	4		Size of data in this transaction part This allows splitting large transactions into smaller parts

Immediately following the header is optional transaction data. Data part contains *transaction parameters*. When these parameters are

used, data part starts with the field containing the number of parameters in the parameter list:

Description	Size	Note
Number of parameters	2	Number of the parameters for this transaction
Parameter list...		

Parameter list contains multiple records with the following structure:

Description	Size	Note
Field ID	2	
Field size	2	Size of the data part
Field data...	size	Actual field content

Every field data format is based on the field type. Currently, there are only 3 predefined field data types: integer, string and binary.

Transaction Types (with Type ID)

This is the list of all transactions in the current version of Hotline software:

ID	Type	Initiator	Constant	
100	Error	?		myTran_Error
101	Get messages	Client	myTran_GetMsgs	
102	New message		Server	myTran_NewMsg
103	Old post news	Client		myTran_OldPostNews
104	Server message		Server	myTran_ServerMsg
105	Send chat	Client		myTran_ChatSend
106	Chat message		Server	myTran_ChatMsg
107	Login	Client		myTran_Login
	Send			

108	instant message	Client		myTran_SendInstantMsg
109	Show agreement		Server	myTran_ShowAgreement
110	Disconnect user	Client		myTran_DisconnectUser
111	Disconnect message		Server	myTran_DisconnectMsg
112	Invite to a new chat	Client		myTran_InviteNewChat
113	Invite to chat	Client	Server	myTran_InviteToChat
114	Reject chat invite	Client		myTran_RejectChatInvite
115	Join chat	Client		myTran_JoinChat
116	Leave chat	Client		myTran_LeaveChat
117	Notify chat of a user change		Server	myTran_NotifyChatChangeUser
118	Notify chat of a delete user		Server	myTran_NotifyChatDeleteUser
119	Notify of a chat subject		Server	myTran_NotifyChatSubject
120	Set chat subject	Client		myTran_SetChatSubject
121	Agreed	Client		myTran_Agreed
122	Server banner		Server	myTran_ServerBanner
	Get file			

200	name list	Client		myTran_GetFileNameList
202	Download file	Client		myTran_DownloadFile
203	Upload file	Client		myTran_UploadFile
204	Delete file	Client		myTran_DeleteFile
205	New folder	Client		myTran_NewFolder
206	Get file info	Client		myTran_GetFileInfo
207	Set file info	Client		myTran_SetFileInfo
208	Move file	Client		myTran_MoveFile
209	Make file alias	Client		myTran_MakeFileAlias
210	Download folder	Client		myTran_DownloadFldr
211	Download info		Server	myTran_DownloadInfo
212	Download banner	Client		myTran_DownloadBanner
213	Upload folder	Client		myTran_UploadFldr
300	Get user name list	Client		myTran_GetUserNameList
301	Notify of a user change		Server	myTran_NotifyChangeUser
302	Notify of a delete user		Server	myTran_NotifyDeleteUser
303	Get client info text	Client		myTran_GetClientInfoText

304	Set client user info	Client		myTran_SetClientUserInfo
350	New user	Client		myTran_NewUser
351	Delete user	Client		myTran_DeleteUser
352	Get user	Client		myTran_GetUser
353	Set user	Client		myTran_SetUser
354	User access		Server	myTran_UserAccess
355	User broadcast	Client	Server	myTran_UserBroadcast
370	Get news category name list	Client		myTran_GetNewsCatNameList
371	Get news article name list	Client		myTran_GetNewsArtNameList
380	Delete news item	Client		myTran_DelNewsItem
381	New news folder	Client		myTran_NewNewsFldr
382	New news category	Client		myTran_NewNewsCat
400	Get news article data	Client		myTran_GetNewsArtData
410	Post news article	Client		myTran_PostNewsArt
411	Delete news article	Client		myTran_DelNewsArt
500	Connection			myTran_KeepConnectionAlive

keep alive

The following are the lists of related transactions that are implemented in the new version of Hotline software:

User Login and Management

ID	Type	Initiator	Note
107	Login	Client	
109	Show agreement	Server	
121	Agreed	Client	
304	Set client user info	Client	
301	Notify of a user change	Server	
300	Get user name list	Client	
302	Notify of a delete user	Server	

Chat Transactions

ID	Type	Initiator	Note
115	Join chat	Client	
112	Invite to a new chat	Client	
113	Invite to chat	Client/Server	
114	Reject chat invite	Client	
117	Notify chat of a user change	Server	
116	Leave chat	Client	
118	Notify chat of a delete user	Server	
120	Set chat subject	Client	
119	Notify of a chat subject	Server	
105	Send chat	Client	
106	Chat message	Server	

Messaging Transactions			
ID	Type	Initiator	Note
104	Server message	Server	
108	Send instant message	Client	

Transaction Description

Transaction types are described using the following format:

Constant: Constant identifier used in the old version of the application.

Access: Specifies access privilege required to perform the transaction.

Initiator: Specifies transaction initiator (client or server).

Fields used in the request: List of fields sent by the transaction initiator.

Fields used in the reply: List of fields sent back to the transaction initiator.

Reply is not sent: Receiver of transaction is not sending reply. **Reply is not expected**: Sender of transaction is not expecting reply.

Error (100)

Constant: myTran_Error

Initiator: None (?)

Get Messages (101)

Constant: myTran_GetMsgs

Initiator: Client

Fields used in the request: None

Fields used in the reply:

ID	Field Name	Note
101	Data	Message text

New Message (102)

Constant: myTran_NewMsg

Initiator: Server

Fields used in the request:

ID	Field Name	Note
101	Data	News text

Reply is not sent.

Old Post News (103)

Constant: myTran_OldPostNews

Access: News Post Article (21)

Initiator: Client

Fields used in the request:

ID	Field Name	Note
101	Data	

Fields used in the reply: None

Server Message (104)

Constant: myTran_ServerMsg

Initiator: Server

Receive a message from the user on the current server, server's administrator, or server software itself.

Fields used in the request:

ID	Field Name	Note
103	User ID	
102	User name	
		Bitmap created by combining the following values:

113	Options	- Automatic response (4)
		- Refuse private chat (2)
		- Refuse private message (1)
101	Data	Message to display
214	Quoting message	Message to quote

If User ID (103) field is not sent, receiver assumes that sender uses the following fields:

ID	Field Name	Note
101	Data	
109	Chat options	Server message (1) or admin message (any other value)

Reply is not sent.

Send Chat (105)

Constant: myTran_ChatSend

Access: Send Chat (10)

Initiator: Client

Send a chat message to the chat.

Fields used in the request:

ID	Field Name	Note
109	Chat options	Optional Normal (0) or alternate (1) chat message
114	Chat ID	Optional
101	Data	Chat message string

Reply is not expected.

Note: While the chat message data can have a theoretical length of 32KB, many servers will clip the text to some length (server dependent) far before this point. Best practice is to not allow the client to send more than 4KB in one chat message.

Chat Message (106)

Constant: myTran_ChatMsg

Initiator: Server

Receive a chat message from the chat.

Fields used in the request:

ID	Field Name	Note
114	Chat ID	
101	Data	Chat text

If Chat ID is not available, the Data field contains:

ID	Field Name	Note
101	Data	Special chat message

Reply is not sent.

Login (107)

Constant: myTran_Login

Initiator: Client

Start login sequence with the server (see *Transaction Sequences*).

Fields used in the request:

ID	Field Name	Note
105	User login	
106	User password	
160	Version	Currently 151

Fields used in the reply:

ID	Field Name	Note
160	Version	

If Version is >= 151, additional fields are included:

ID	Field Name	Note
161	Banner ID	Used for making HTTP request to get banner
162	Server name	Server name string

If server version is < 151, client sends Set Client User Info (304) transaction with only User Name (102) and User Icon ID (104) fields used, and does not expect a reply. It does not expect agreement to be received (109). Subsequently, it sends Get User Name List (300) request, followed by Get File Name List (200) or Get News Category Name List (370), depending on user preferences. After that, a banner is requested from HTTP server.

Send Instant Message (108)

Constant: myTran_SendInstantMsg

Initiator: Client

Send instant message to the user on the current server.

Fields used in the request:

ID	Field Name	Note
103	User ID	
113	Options	One of the following values: - User message (myOpt_UserMessage = 1) - Refuse message (myOpt_RefuseMessage = 2) - Refuse chat (myOpt_RefuseChat = 3) - Automatic response (myOpt_AutomaticResponse = 4)
101	Data	Optional
214	Quoting message	Optional

Fields used in the reply: None

Show Agreement (109)

Constant: myTran_ShowAgreement

Initiator: Server

Receive agreement that will be presented to the user of the client application. This transaction is part of the login sequence (see *Transaction Sequences***).**

Fields used in the request:

ID	Field Name	Note
101	Data	Agreement string
154	No server agreement	Optional No agreement available (1)
152	Server banner type	
153	Server banner URL	Optional If banner type is URL (1)
151	Server banner	Optional If banner type is not URL (1)

Reply is not sent.

Disconnect User (110)

Constant: myTran_DisconnectUser

Access: Disconnect User (22)

Initiator: Client

Disconnect user from the current server.

Fields used in the request:

ID	Field Name	Note
103	User ID	
113	Options	Optional Ban options

101	Data	Optional Name?

Fields used in the reply: None

Disconnect Message (111)

Constant: myTran_DisconnectMsg

Initiator: Server

Receive disconnect message from the server. Upon receiving this transaction, client should close the connection with server.

Fields used in the request:

ID	Field Name	Note
101	Data	Message to display on disconnect (mandatory)

Reply is not sent.

Invite New Chat (112)

Constant: myTran_InviteNewChat

Initiator: Client

Invite users to the new chat.

Fields used in the request:

ID	Field Name	Note
103	User ID	Optional
103	User ID …	Optional More user IDs…

Fields used in the reply:

ID	Field Name	Note
103	User ID	
104	User icon ID	

112	User flags	
102	User name	
114	Chat ID	

Invite To Chat (113)

Constant: myTran_InviteToChat

Initiator: Client

Invite user to the existing chat.

Fields used in the request:

ID	Field Name	Note
103	User ID	User to invite
114	Chat ID	

Reply is not expected.

The server can also be an initiator of this transaction.

Initiator: Server

Fields used in the request:

ID	Field Name	Note
114	Chat ID	
103	User ID	User to invite
102	User name	

Reply is not sent.

When client receives this message from the sever with version < 151, and client has automatic response or reject chat flag set, Reject Chat Invite (114) transaction is sent back to the server.

Reject Chat Invite (114)

Constant: myTran_RejectChatInvite

Initiator: Client

Reject invitation to join the chat.

Fields used in the request:

ID	Field Name	Note
114	Chat ID	

Reply is not expected.

Join Chat (115)

Constant: myTran_JoinChat

Initiator: Client

Join the chat.

Fields used in the request:

ID	Field Name	Note
114	Chat ID	

Fields used in the reply:

ID	Field Name	Note
115	Chat subject	
300	User name with info	Optional
300 ...	User name with info ...	Optional More user names with info

Leave Chat (116)

Constant: myTran_LeaveChat

Initiator: Client

Leave the chat.

Fields used in the request:

ID	Field Name	Note

114	Chat ID	

Reply is not expected.

Notify Chat Change User (117)

Constant: myTran_NotifyChatChangeUser

Initiator: Server

Notify the user of the chat that the information for some another user changed, or that a new user just joined the chat. This information should be added to (user joined the chat), or updated (user changed its info) in the chat user list.

Fields used in the request:

ID	Field Name	Note
114	Chat ID	
103	User ID	
104	User icon ID	
112	User flags	
102	User name	

Reply is not sent.

In the Hotline implementation v1.8x, this transaction is in fact used only when the user joins the chat. The user information update done by Notify Change User (301) transaction is also applied to any chat rooms on the clients receiving the update.

Notify Chat Delete User (118)

Constant: myTran_NotifyChatDeleteUser

Initiator: Server

Notify the user of the chat that a user left that chat. The client should update the chat user list.

Fields used in the request:

ID	Field Name	Note

114	Chat ID	
103	User ID	

Reply is not sent.

Notify Chat Subject (119)

Constant: myTran_NotifyChatSubject

Initiator: Server

Notify the user of the chat of the chat subject.

Fields used in the request:

ID	Field Name	Note
114	Chat ID	
115	Chat subject	Chat subject string

Reply is not sent.

Set Chat Subject (120)

Constant: myTran_SetChatSubject

Initiator: Client

Set chat subject for the chat.

Fields used in the request:

ID	Field Name	Note
114	Chat ID	
115	Chat subject	Chat subject string

Reply is not expected.

Agreed (121)

Constant: myTran_Agreed

Initiator: Client

Notify the server that the user accepted the server agreement.

Fields used in the request:

ID	Field Name	Note
102	User name	
104	User icon ID	
113	Options	Bitmap created by combining the following values: - Automatic response (4) - Refuse private chat (2) - Refuse private message (1)
215	Automatic response	Optional Automatic response string used only if the options field indicates this feature

Fields used in the reply: None

After receiving server's acknowledgement, the client sends Get User Name List (300) request, followed by Get File Name List (200) or Get News Category Name List (370), depending on user preferences.

Server Banner (122)

Constant: myTran_ServerBanner

Initiator: Server

Notify the client that a new banner should be displayed.

Fields used in the request:

ID	Field Name	Note
152	Server banner type	Uses only literal values
153	Server banner URL	Optional

Reply is not sent.

If banner type is URL, it is requested from that URL. Otherwise, the banner is requested from the server by Download Banner (212) request.

This transaction uses only literal value constants in the banner type field (etc. 'URL ', 'JPEG' or other).

Get File Name List (200)

Constant: myTran_GetFileNameList

Initiator: Client

Get the list of file names from the specified folder.

Fields used in the request:

ID	Field Name	Note
202	File path	Optional If not specified, root folder assumed

Fields used in the reply:

ID	Field Name	Note
200	File name with info	Optional
200 ...	File name with info ...	Optional More file names with info

Download File (202)

Constant: myTran_DownloadFile

Access: Download File (2)

Initiator: Client

Download the file from the specified path on the server.

Fields used in the request:

ID	Field Name	Note

201	File name	
202	File path	
203	File resume data	Optional
204	File transfer options	Optional Currently set to 2 Used only for TEXT, JPEG, GIFF, BMP or PICT files

Fields used in the reply:

ID	Field Name	Note
108	Transfer size	Size of data to be downloaded
207	File size	
107	Reference number	Used later for transfer
116	Waiting count	

After receiving reply from the server, the client opens TCP (or HTTP) connection to base port + 1 (HTTP uses base port + 3). On successful establishment, client sends the following record using the new connection:

Description	Size	Data	Note
Protocol	4	'HTXF'	0x48545846
Reference number	4		Use reference number received from the server
Data size	4	0	
RSVD	4	0	?

After this, server sends the flattened file object (see *Flattened File Object*) using this new TCP connection.

Upload File (203)

Constant: myTran_UploadFile

Access: Upload File (1)

Initiator: Client

Upload a file to the specified path on the server.

Fields used in the request:

ID	Field Name	Note
201	File name	
202	File path	
204	File transfer options	Optional Used only to resume download, currently has value 2
108	File transfer size	Optional Used if download is not resumed

Fields used in the reply:

ID	Field Name	Note
203	File resume data	Optional Used only to resume download
107	Reference number	

After receiving reply from the server, the client opens TCP (or HTTP) connection to base port + 1 (HTTP uses base port + 3). On successful establishment, client sends the following record using the new connection:

Description	Size	Data	Note
Protocol	4	'HTXF'	0x48545846
Reference number	4		Use reference number received from the server
Data size	4		File size
RSVD	4	0	?

After this, client sends the flattened file object (see *Flattened File Object*) using this new TCP connection.

Delete File (204)

Constant: myTran_DeleteFile

Access: Delete File (0) or Delete Folder (6)

Initiator: Client

Delete the specific file from the server.

Fields used in the request:

ID	Field Name	Note
201	File name	
202	File path	

Fields used in the reply: None

New Folder (205)

Constant: myTran_NewFolder

Access: Create Folder (5)

Initiator: Client

Create a new folder on the server.

Fields used in the request:

ID	Field Name	Note
201	File name	
202	File path	

Fields used in the reply: None

Get File Info (206)

Constant: myTran_GetFileInfo

Initiator: Client

Request file information from the server.

Fields used in the request:

ID	Field Name	Note

201	File name	
202	File path	Optional

Fields used in the reply:

ID	Field Name	Note
201	File name	
205	File type string	
206	File creator string	
210	File comment	Comment string
213	File type	
208	File create date	
209	File modify date	
207	File size	

Set File Info (207)

Constant: myTran_SetFileInfo

Access: Set File Comment (28) or Set Folder Comment (29)

Initiator: Client

Set information for the specified file on the server.

Fields used in the request:

ID	Field Name	Note
201	File name	
202	File path	Optional
211	File new name	Optional
210	File comment	Optional

Fields used in the reply: None

Move File (208)

Constant: myTran_MoveFile

Initiator: Client

Move the file from one folder to another on the same server.

Fields used in the request:

ID	Field Name	Note
201	File name	
202	File path	
212	File new path	

Fields used in the reply: None

Make File Alias (209)

Constant: myTran_MakeFileAlias

Access: Make Alias (31)

Initiator: Client

Make the file alias using the specified path.

Fields used in the request:

ID	Field Name	Note
201	File name	
202	File path	
212	File new path	Destination path

Fields used in the reply: None

Download Folder (210)

Constant: myTran_DownloadFldr

Access: Download File (2)

Initiator: Client

Download all files from the specified folder and its subfolders on the server.

Fields used in the request:

ID	Field Name	Note
201	File name	
202	File path	

Fields used in the reply:

ID	Field Name	Note
220	Folder item count	
107	Reference number	Used later for transfer
108	Transfer size	Size of data to be downloaded
116	Waiting count	

After receiving reply from the server, the client opens TCP (or HTTP) connection to base port + 1 (HTTP uses base port + 3). On successful establishment, client sends the following record using the new connection:

Description	Size	Data	Note
Protocol	4	'HTXF'	0x48545846
Reference number	4		Use reference number received from the server
Data size	4	0	
Type	2	1	
RSVD	2	0	?
Download folder action	2	3	Next file action (3) See *Download folder actions*

For every item in the folder, server replies with:

Description	Size	Data	Note
Header size	2		
Header data	size		

Header data contains the following:

Description	Size	Data	Note
Type	2		?
File path	rest		

After receiving this header client can reply in 3 ways.

(1) If type is an odd number (unknown type?), or file download for the current file is completed:

Description	Size	Data	Note
Download folder action	2	3	Next file action (3) See *Download folder actions*

This notifies the server to send next item header.

(2) If download of a file is to be resumed:

Description	Size	Data	Note
Download folder action	2	2	Resume file transfer (2) See *Download folder actions*
Resume data size	2		
File resume data	size		See content for field (203)

(3) Otherwise, file download is requested by:

Description	Size	Data	Note
Download folder			Send file action (1) starts file download

action	2	1	See *Download folder actions*

When download is requested (case 2 or 3), server replies with:

Description	Size	Data	Note
File size	4		
File content…	size		Actual flattened file object (see *Flattened File Object*)

After every file download client could request next file:

Description	Size	Data	Note
Download folder action	2	3	Next file action (3) See *Download folder actions*

This notifies the server to send next item header.

Download Info (211)

Constant: myTran_DownloadInfo

Initiator: Server

Notify the client that all download slots on the server are full.

Fields used in the request:

ID	Field Name	Note
107	Reference number	Download reference number
116	Waiting count	Position in the server's queue

Reply is not sent.

Download Banner (212)

Constant: myTran_DownloadBanner

Initiator: Client

Request a new banner from the server.

Fields used in the request: None

Fields used in the reply:

ID	Field Name	Note
107	Reference number	Used later for transfer
108	Transfer size	Size of data to be downloaded

After receiving reply from the server, the client opens TCP (or HTTP) connection to base port + 1 (HTTP uses base port + 3). On successful establishment, client sends the following record using the new connection:

Description	Size	Data	Note
Protocol	4	'HTXF'	0x48545846
Reference number	4		Use reference number received from the server
Data size	4	0	
Type	2	2	
RSVD	2	0	?

After this, server sends the file content using this new TCP connection.

Upload Folder (213)

Constant: myTran_UploadFldr

Access: Upload File (1)

Initiator: Client

Upload all files from the local folder and its subfolders, to the specified path on the server.

Fields used in the request:

ID	Field Name	Note
201	File name	
202	File path	

108	Transfer size	Total size of all items in the folder
220	Folder item count	
204	File transfer options	Optional Currently set to 1

Fields used in the reply:

ID	Field Name	Note
107	Reference number	Used later for transfer

After receiving reply from the server, the client opens TCP (or HTTP) connection to base port + 1 (HTTP uses base port + 3). On successful establishment, client sends the following record using the new connection:

Description	Size	Data	Note
Protocol	4	'HTXF'	0x48545846
Reference number	4		Use reference number received from the server
Data size	4	0	
Type	2	1	
RSVD	2	0	?

Server can reply with:

Description	Size	Data	Note
Download folder action	2	3	Next file action (3) See *Download folder actions*

After which client sends:

Description	Size	Data	Note
Data size	2		Size of this structure (not including data size element itself)
Is folder	2	0 or 1	Is the following file path a folder
Path item			

count	2		Number of items in the path
File name path...			

File name path contains:

Description	Size	Data	Note
	2	0	Currently 0
Name size	1		
File/folder name	size		

After every file, server can send one of 3 requests.

(1) Request next file:

Description	Size	Data	Note
Download folder action	2	3	Next file action (3) See *Download folder actions*

This notifies the client to send next item.

(2) Resume a file download procedure:

Description	Size	Data	Note
Download folder action	2	2	Resume file transfer (2) See *Download folder actions*
Resume data size	2		
File resume data	size		See content for field (203)

After receiving this request, client starts sending file content from the requested location in the file.

(3) Request a file download:

Description	Size	Data	Note

Download folder action	2	1	Send file action (1) starts file download See *Download folder actions*

Client replies to download requests with:

Description	Size	Data	Note
File size	4		Current file size

After this client sends the flattened file object (see *Flattened File Object*).

Get User Name List (300)

Constant: myTran_GetUserNameList

Initiator: Client

Request the list of all users connected to the current server.

Fields used in the request: None

Fields used in the reply:

ID	Field Name	Note
300	User name with info	Optional
300 ...	User name with info ...	Optional More user names with info

Notify Change User (301)

Constant: myTran_NotifyChangeUser

Initiator: Server

Notify the user that the information for some another user changed, or that a new user just connected to the server. This information is to be added to (user joined), or updated (user changed its info) in the existing user list.

Fields used in the request:

ID	Field Name	Note

103	User ID	
104	User icon ID	
112	User flags	
102	User name	

Reply is not sent.

In the Hotline implementation v1.8x, this transaction is also applied to any chat rooms on the clients receiving the update.

Notify Delete User (302)

Constant: myTran_NotifyDeleteUser

Initiator: Server

Notify the user that some another user disconnected from the server. The client should update the existing user list.

Fields used in the request:

ID	Field Name	Note
103	User ID	

Reply is not sent.

Get Client Info Text (303)

Constant: myTran_GetClientInfoText

Access: Get Client Info (24)

Initiator: Client Request user information for the specific user.

Fields used in the request:

ID	Field Name	Note
103	User ID	

Fields used in the reply:

ID	Field Name	Note
102	User name	

101	Data	User info text string

Set Client User Info (304)

Constant: myTran_SetClientUserInfo

Initiator: Client

Set user preferences on the server.

Fields used in the request:

ID	Field Name	Note
102	User name	
104	User icon ID	
113	Options	Bitmap created by combining the following values: - Automatic response (4) - Refuse private chat (2) - Refuse private message (1)
215	Automatic response	Optional Automatic response string used only if the options field indicates this feature

Reply is not expected.

New User (350)

Constant: myTran_NewUser

Initiator: Client

Add a new user to the server's list of allowed users.

Fields used in the request:

ID	Field Name	Note
105	User login	
106	User password	

102	User name	
110	User access	User access privileges bitmap (see *Access Privileges*)

Fields used in the reply: None

Delete User (351)

Constant: myTran_DeleteUser

Initiator: Client

Delete the specific user from the server's list of allowed users.

Fields used in the request:

ID	Field Name	Note
105	User login	

Fields used in the reply: None

Get User (352)

Constant: myTran_GetUser

Initiator: Client

Request the information for the specific user from the server's list of allowed users.

Fields used in the request:

ID	Field Name	Note
105	User login	

Fields used in the reply:

ID	Field Name	Note
102	User name	
105	User login	Every character in this string is negated (login[i] = ~login[i])
106	User password	
		User access privileges bitmap (see *Access*

110	User access	*Privileges*)

Set User (353)

Constant: myTran_SetUser

Initiator: Client

Set the information for the specific user in the server's list of allowed users.

Fields used in the request:

ID	Field Name	Note
105	User login	
106	User password	
102	User name	
110	User access	User access privileges bitmap (see *Access Privileges*)

Fields used in the reply: None

User Access (354)

Constant: myTran_UserAccess

Initiator: Server

Set access privileges for the current user.

Fields used in the request:

ID	Field Name	Note
110	User access	User access privileges bitmap (see *Access Privileges*)

Reply is not sent.

User Broadcast (355)

Constant: myTran_UserBroadcast

Access: Broadcast (32)

Initiator: Client

Broadcast the message to all users on the server.

Fields used in the request:

ID	Field Name	Note
101	Data	

Fields used in the reply: None

The server can also be an initiator of this transaction.

Initiator: Server

Fields used in the request:

ID	Field Name	Note
101	Data	Administrator message

Reply is not sent.

Get News Category Name List (370)

Constant: myTran_GetNewsCatNameList

Initiator: Client

Get the list of category names at the specified news path.

Fields used in the request:

ID	Field Name	Note
325	News path	Optional

Fields used in the reply:

ID	Field Name	Note
323	News category list data	Optional
323	News category list data	Optional
...	...	More news categories

If version of client/server is 1.5 (prior to April 15, 1999?), instead of the previous reply, the following is sent:

ID	Field Name	Note
320	News category list data	Optional
320	News category list data	Optional
...	...	More news categories

Get News Article Name List (371)

Constant: myTran_GetNewsArtNameList

Initiator: Client Get the list of article names at the specified news path.

Fields used in the request:

ID	Field Name	Note
325	News path	Optional

Fields used in the reply:

ID	Field Name	Note
321	News article list data	Optional

Delete News Item (380)

Constant: myTran_DelNewsItem

Access: News Delete Folder (37) or News Delete Category (35)

Initiator: Client

Delete an existing news item from the server.

Fields used in the request:

ID	Field Name	Note
325	News path	

Fields used in the reply: None

New News Folder (381)

Constant: myTran_NewNewsFldr

Access: News Create Folder (36)

Initiator: Client

Create new news folder on the server

Fields used in the request:

ID	Field Name	Note
201	File name	
325	News path	

Fields used in the reply: None

New News Category (382)

Constant: myTran_NewNewsCat

Access: News Create Category (34)

Initiator: Client

Create new news category on the server.

Fields used in the request:

ID	Field Name	Note
322	News category name	
325	News path	

Fields used in the reply: None

Get News Article Data (400)

Constant: myTran_GetNewsArtData

Access: News Read Article (20)

Initiator: Client

Request information about the specific news article.

Fields used in the request:

ID	Field Name	Note
325	News path	
326	News article ID	
327	News article data flavor	

Fields used in the reply:

ID	Field Name	Note
328	News article title	
329	News article poster	
330	News article date	
331	Previous article ID	
332	Next article ID	
335	Parent article ID	
336	First child article ID	
327	News article data flavor	Should be "text/plain�? Other values are currently ignored
333	News article data	Optional (if data flavor is "text/plain�?)

Post News Article (410)

Constant: myTran_PostNewsArt

Access: News Post Article (21)

Initiator: Client

Post new news article on the server.

Fields used in the request:

ID	Field Name	Note
325	News path	
326	News article ID	ID of the parent article?
328	News article title	
334	News article flags	
327	News article data flavor	Currently "text/plain�?
333	News article data	

Fields used in the reply: None

Delete News Article (411)

Constant: myTran_DelNewsArt

Access: News Delete Article (33)

Initiator: Client

Delete the specific news article.

Fields used in the request:

ID	Field Name	Note
325	News path	
326	News article ID	
337	News article – recursive delete	Delete child articles (1) or not (0)

Fields used in the reply: None

Connection Keep Alive (500)

Constant: myTran_KeepConnectionAlive

Pings the connection (?)

Flattened File Object

Transactions 202 (Download File), 203 (Upload File), 210 (Download Folder) and 213 (Upload Folder) format the file object in the following way:

Flat file header:

Description	Size	Data	Note
Format	4	'FILP'	0x46494C50
Version	2	1	
RSVD	16		
Fork count	2	2	

Flat file information fork header:

Description	Size	Data	Note
Fork type	4	'INFO'	0x494E464F
Compression type	4	0	Currently no compression
RSVD	4		
Data size	4		Size of the flat file information fork

Flat file information fork:

Description	Size	Data	Note
Platform	4	'AMAC' or 'MWIN'	Operating system used
Type signature	4		File type signature
Creator signature	4		File creator signature
Flags	4		
Platform flags	4		

RSVD	32		
Create date	8		See description for the File Create Date field (208)
Modify date	8		See description for the File Modify Date field (209)
Name script	2		
Name size	2		
Name	size		Maximum 128 characters
Comment size	2		
Comment	comment size		When Comment size > 0

Flat file data fork header:

Description	Size	Data	Note
Fork type	4	'DATA'	0x44415441
Compression type	4	0	Currently no compression
RSVD	4		
Data size	4		Actual file content size

Transaction Fields

There are 3 predefined field data types: integer, string and binary. If field data does not fit in the first two categories, it is sent as binary data and interpreted by the receiving machine. Some of the binary fields are currently used as strings. All integer fields are treated as unsigned, and can be sent as 16 or 32-bit numbers. This is determined by evaluation of the number itself. Namely, if integer can be represented using only 2 bytes, it is sent as such. In the case when the number is greater than 2^{16}, it's sent as 32-bit number. String fields currently use 8-bit ASCII character set.

Error Text (100)

Constant: myField_ErrorText

Data (101)

Constant: myField_Data

Type: Binary

User Name (102)

Constant: myField_UserName

Type: String

User ID (103)

Constant: myField_UserID

Type: Integer

User Icon ID (104)

Constant: myField_UserIconID

Type: Integer

User Login (105)

Constant: myField_UserLogin

Type: String

User Password (106)

Constant: myField_UserPassword

Type: String

Reference Number (107)

Constant: myField_RefNum

Type: Integer

Transfer Size (108)

Constant: myField_TransferSize

Type: Integer

Chat Options (109)

Constant: myField_ChatOptions

Type: Integer

User Access (110)

Constant: myField_UserAccess

Type: Binary

This field is represented as 64-bit bitmap. The specific bit meaning is described in the *Access Privileges* section of this document.

User Alias (111)

Constant: myField_UserAlias

User Flags (112)

Constant: myField_UserFlags

Type: Integer

User flags field is a bitmap with the following values:

Bit	Value	Description
0	1	User is away
1	2	User is admin (or disconnected?)
2	4	User refuses private messages
3	8	User refuses private chat

Options (113)

Constant: myField_Options

Type: Integer

Chat ID (114)

Constant: myField_ChatID

Type: Integer

Chat Subject (115)

Constant: myField_ChatSubject

Type: String

Waiting Count (116)

Constant: myField_WaitingCount

Type: Integer

Server Agreement (150)

Constant: myField_ServerAgreement

Server Banner (151)

Constant: myField_ServerBanner

Type: Binary

Server Banner Type (152)

Constant: myField_ServerBannerType

Type: Integer

This field can have one of the following values:

Value	Equivalent Value	Description
1	'URL '	URL link
3	'JPEG'	JPEG file
4	'GIFf'	GIF file

5	'BMP '	BMP file
6	'PICT	PICT file

Server Banner URL (153)

Constant: myField_ServerBannerUrl

Type: Binary

No Server Agreement (154)

Constant: myField_NoServerAgreement

Type: Integer

The value of this field is 1 if there is no agreement to be sent.

Version (160)

Constant: myField_Vers

Type: Integer

Community Banner ID (161)

Constant: myField_CommunityBannerID

Type: Integer

Server Name (162)

Constant: myField_ServerName

Type: Binary

File Name with Info (200)

Constant: myField_FileNameWithInfo

Type: Binary

File name with info field content is presented in this structure:

Description	Size	Data	Note
Type	4		Folder ('fldr') or other

Creator	4		
File size	4		
	4		Reserved?
Name script	2		
Name size	2		
Name data	size		

File Name (201)

Constant: myField_FileName

Type: String

File Path (202)

Constant: myField_FilePath

Type: Binary See Path Parameters for more info.

File Resume Data (203)

Constant: myField_FileResumeData

Type: Binary

File resume data field content is presented in this structure:

Description	Size	Data	Note
Format	4	'RFLT'	
Version	2	1	Currently 1
RSVD	34		?
Fork count	2	2	Currently 2
Fork info list...			

Fork info list contains one or more records with the following structure:

Description	Size	Data	Note

Fork	4	'DATA'	
Data size	4		Current file size
RSVD A	4		?
RSVD B	4		?

File Transfer Options (204)

Constant: myField_FileXferOptions

Type: Integer

File Type String (205)

Constant: myField_FileTypeString

Type: String

File Creator String (206)

Constant: myField_FileCreatorString

Type: String

File Size (207)

Constant: myField_FileSize

Type: Integer

File Create Date (208)

Constant: myField_FileCreateDate

Type: Binary

File create date field has this structure:

Description	Size	Data	Note
Year	2		
Milliseconds	2		
Seconds	4		

See Date Parameters for more info.

File Modify Date (209)

Constant: myField_FileModifyDate

Type: Binary

File modify date field has this structure:

Description	Size	Data	Note
Year	2		
Milliseconds	2		
Seconds	4		

See Date Parameters for more info.

File Comment (210)

Constant: myField_FileComment

Type: String

File New Name (211)

Constant: myField_FileNewName

Type: String

File New Path (212)

Constant: myField_FileNewPath

Type: Binary

File Type (213)

Constant: myField_FileType

Type: Binary

File type field contains only one value:

Description	Size	Data	Note
File type	4		File type code ('fldr' or other)

Quoting Message (214)

Constant: myField_QuotingMsg

Type: Binary

Automatic Response (215)

Constant: myField_AutomaticResponse

Type: String

Folder Item Count (220)

Constant: myField_FldrItemCount

Type: Integer

User Name with Info (300)

Constant: myField_UserNameWithInfo

Type: Binary

User name with info field contains this structure:

Description	Size	Data	Note
User ID	2		
Icon ID	2		
User flags	2		
User name size	2		
User name	size		User name string

News Category GUID (319)

Constant: myField_NewsCatGUID

News Category List Data (320)

Constant: myField_NewsCatListData

Type: Binary

News category list data field contains this structure:

Description	Size	Data	Note
Type	1	1, 10 or 255	Category folder (1), category (10) or other (255)
Category name	rest		

This field is used for client/server version 1.5 (prior to April 15, 1999?).

News Article List Data (321)

Constant: myField_NewsArtListData

Type: Binary

News article list data field contains this structure:

Description	Size	Data	Note
ID	4		
Article count	4		Number of articles
Name size	1		
Name	size		Name string
Description size	1		
Description	size		Description string
List of articles…			Optional (if article count > 0)

List of articles contains:

Description	Size	Data	Note
Article ID	4		
Time stamp	8		Year (2 bytes), milliseconds (2 bytes) and seconds (4 bytes)
Parent article ID	4		
Article flags	4		
Flavor count	2		
Title size	1		
Title	Size		Title string
Poster size	1		
Poster	Size		Poster string
Flavor list...			Optional (if flavor count > 0)

Flavor list has the following structure:

Description	Size	Data	Note
Flavor size	1		
Flavor text	size		MIME type string
Article size	2		

News Category Name (322)

Constant: myField_NewsCatName

Type: String

News Category List Data 1.5 (323)

Constant: myField_NewsCatListData15

Type: Binary

News category list data field contains this structure:

Description	Size	Data	Note

Type	2	2 or 3	Bundle (2) or category (3)

If type value indicates a bundle, what follows the type is:

Description	Size	Data	Note
Count	2		
Name size	1		
Name data	size		

In the case of a category type, type value is followed by:

Description	Size	Data	Note
Count	2		
GUID			
Add SN	4		
Delete SN	4		
Name size	1		
Name data	size		

News Path (325)

Constant: myField_NewsPath

Type: Binary

News Article ID (326)

Constant: myField_NewsArtID

Type: Integer

News Article Data Flavor (327)

Constant: myField_NewsArtDataFlav

Type: String

News Article Title (328)

Constant: myField_NewsArtTitle

Type: String

News Article Poster (329)

Constant: myField_NewsArtPoster

Type: String

News Article Date (330)

Constant: myField_NewsArtDate

Type: Binary

News article date field contains this structure:

Description	Size	Data	Note
Year	2		
Milliseconds	2		
Seconds	4		

See Date Parameters for more info.

News Article – Previous Article (331)

Constant: myField_NewsArtPrevArt

Type: Integer

News Article – Next Article (332)

Constant: myField_NewsArtNextArt

Type: Integer

News Article Data (333)

Constant: myField_NewsArtData

Type: Binary

News Article Flags (334)

Constant: myField_NewsArtFlags

Type: Integer

News Article – Parent Article (335)

Constant: myField_NewsArtParentArt

Type: Integer

News Article – First Child Article (336)

Constant: myField_NewsArt1stChildArt

Type: Integer

News Article – Recursive Delete (337)

(Delete Children)

Constant: myField_NewsArtRecurseDel

Type: Integer

Access Privileges

The following is the list of access privileges currently employed by the application. There are 3 types of access privileges: general, folder and bundle. Folder privileges are set per folder. Bundle access is related to the logical grouping of the information. General access privileges are used to set privileges for a user.

Delete File (0)

Constant: myAcc_DeleteFile

Type: folder

Upload File (1)

Constant: myAcc_UploadFile

Type: folder, general

Download File (2)

Constant: myAcc_DownloadFile

Type: folder, general

Rename File (3)

Constant: myAcc_RenameFile

Move File (4)

Constant: myAcc_MoveFile

Create Folder (5)

Constant: myAcc_CreateFolder

Type: folder

Delete Folder (6)

Constant: myAcc_DeleteFolder

Type: folder

Rename Folder (7)

Constant: myAcc_RenameFolder

Move Folder (8)

Constant: myAcc_MoveFolder

Read Chat (9)

Constant: myAcc_ReadChat

Type: general

Send Chat (10)

Constant: myAcc_SendChat

Type: general

Open Chat (11)

Constant: myAcc_OpenChat

Close Chat (12)

Constant: myAcc_CloseChat

Show in List (13)

Constant: myAcc_ShowInList

Create User (14)

Constant: myAcc_CreateUser

Delete User (15)

Constant: myAcc_DeleteUser

Open User (16)

Constant: myAcc_OpenUser

Modify User (17)

Constant: myAcc_ModifyUser

Change Own Password (18)

Constant: myAcc_ChangeOwnPass

Send Private Message (19)

Constant: myAcc_SendPrivMsg

News Read Article (20)

Constant: myAcc_NewsReadArt

Type: bundle, general

News Post Article (21)

Constant: myAcc_NewsPostArt

Type: general, bundle

Disconnect User (22)

Constant: myAcc_DisconUser

Type: general

Cannot be Disconnected (23)

Constant: myAcc_CannotBeDiscon

Get Client Info (24)

Constant: myAcc_GetClientInfo

Type: general

Upload Anywhere (25)

Constant: myAcc_UploadAnywhere

Any Name (26)

Constant: myAcc_AnyName

Type: general

No Agreement (27)

Constant: myAcc_NoAgreement

Set File Comment (28)

Constant: myAcc_SetFileComment

Type: folder

Set Folder Comment (29)

Constant: myAcc_SetFolderComment

Type: folder

View Drop Boxes (30)

Constant: myAcc_ViewDropBoxes

Make Alias (31)

Constant: myAcc_MakeAlias

Type: folder

Broadcast (32)

Constant: myAcc_Broadcast

Type: general

News Delete Article (33)

Constant: myAcc_NewsDeleteArt

Type: bundle

News Create Category (34)

Constant: myAcc_NewsCreateCat

Type: bundle

News Delete Category (35)

Constant: myAcc_NewsDeleteCat

Type: bundle

News Create Folder (36)

Constant: myAcc_NewsCreateFldr

Type: bundle

News Delete Folder (37)

Constant: myAcc_NewsDeleteFldr

Type: bundle

Download Folder Actions

These values are used to control folder upload/download process. When an application receives folder upload request, it is presented with the first applicable file. In the reply, application will specify an action to be performed:

Send File (1)

Constant: dlFldrAction_SendFile

Send file action starts the download of the file specified in the request. An additional TCP connection is opened to transfer this file, according to the protocol described in Download Folder (210) and Upload Folder (213) transaction.

Resume File Download (2)

Constant: dlFldrAction_ResumeFile

This action is similar to the send file action. It also starts the download, while providing the starting position in the file. An additional TCP connection is opened to transfer this file, in the same manner as for send file action.

Next File (3)

Constant: dlFldrAction_NextFile

Next file action notifies the receiver to send the name of the next file in a folder. Download of the current file in not initiated.

Transaction Sequences

Hotline client contains few predefined transaction sequences in its current implementation. These sequences are described in this section.

Login

After sending Login transaction (107), server will reply with Show Agreement (109). If user accepts the agreement, Hotline client sends Agreed transaction (121), followed by Get User Name List (300). Next, a Get File Name List (200) or Get News Category Name List (370) transaction is sent, depending on user preferences.

If server version is < 151, server will not send Show Agreement reply. In this case, after Login (107) transaction is sent, client sends Set Client User Info (304) transaction with only User Name (102) and User Icon ID (104) fields used, and does not expect a reply. Subsequently, it sends Get User Name List (300) request, followed by Get File Name List (200) or Get News Category Name List (370), depending on user preferences. After that, a banner is requested from HTTP server.

See the Transactions page for more details.

Invite To Chat

When client receives Invite To Chat (113) transaction from the sever with version < 151, and client has automatic response or reject chat flag set, Reject Chat Invite (114) transaction is sent back to the server.

Tracker Interface

All string values use 8-bit ASCII character set encoding.

Client Interface with Tracker

After establishing a connection with tracker, the following information is sent:

Description	Size	Data	Note
Magic number	4	'HTRK'	

Version	2	1 or 2	Old protocol (1) or new (2)

When version number is 2, request also includes additional data:

Description	Size	Data	Note
Login size	1	>= 31	Login string size
Login	size		Login string (padded with 0)
Password size	1	>= 31	Password string size
Password	size		Password string (padded with 0)

Reply received from the tracker starts with a header:

Description	Size	Data	Note
Magic number	4	'HTRK'	0x4854524B
Version	2	1 or 2	Old protocol (1) or new (2)

Server information header follows, formatted as:

Description	Size	Data	Note
Message type	2	1	Sending list of servers
Message data size	2		Remaining size of this request
Number of servers	2		Number of servers in the server list
Number of servers	2		Same as previous field
Server list…			

A record in the server list has the following structure:

Description	Size	Data	Note
IP address	4		Server's IP address
IP port number	2		Server's IP port number
Number of users	2		Number of users connected to this particular server

	2	0	
Name size	1		Server's name string size
Name	size		Server's name
Description size	1		Server's description string size
Description	size		Description of the server

If the number of servers in the server list is less than number of servers specified in the server information header, client will expect an additional server information, starting with the new server information header. The field containing number of servers in the new header should have the same value as the previous one.

When a client is connected to the tracker over the HTTP tunneling protocol, the client does not send any request to the tracker, although it still expects a properly formatted reply. In this case establishing a connection to the tracker indicates a request for the server list.

Server Interface with Tracker

Server sets up UDP port that is used to periodically send the following information to the trackers:

Description	Size	Data	Note
	2	1	
IP port number	2		Server's listening UDP port number
Number of users	2		Number of users connected to this particular server
	2	0	
Pass ID	4		Random number generated by the server
Name size	1		Server's name string size
Name	size		Server's name
Description size	1		Server's description string size

Description	size		Description of the server

In the case when old (?) tracker is used, the additional information is formatted as follows:

Description	Size	Data	Note
Password size	1		Server's tracker password string size
Password	size		Server's tracker password

For a new version of the tracker:

Description	Size	Data	Note
Login size	1		Server's tracker login string size
Login	size		Server's tracker login
Password size	1		Server's tracker password string size
Password	size		Server's tracker password

HTTP Tunneling

When client is unable to communicate with the server using plain TCP connection, HTTP tunneling can be utilized to access the Hotline server over an HTTP proxy. To accomplish this, the client creates two connections to the server. One would be used for sending, and other for receiving data. After these connections are open, the client begins transmitting standard HTTP requests. If HTTP proxy terminates connection while that connection is still in use, the client recreates them, and interrupted requests are resent.

HTTP Requests

HTTP POST request is sent over sending, while GET request is sent over receiving connection. The POST request is specified as follows:

POST <address> HTTP/1.0\r\n

Proxy-Connection: Keep-Alive\r\n

Pragma: no-cache\r\n

Host: <host>\r\n

Content-Length: 999999999\r\n

Content-Type: hotline/protocol\r\n

\r\n

The server replies to this request at the time when connection is
about to be closed, as:

HTTP/1.0 302 Found\r\n

Connection: close\r\n

Content-Length: 8\r\n

Content-Type: hotline/protocol\r\n

\r\n

Next 8 bytes are filled with 0 to indicate termination of a connection.

GET request is specified as:

GET<address> HTTP/1.0\r\n

Proxy-Connection: Keep-Alive\r\n

Pragma: no-cache\r\n

Host: <host>\r\n

Accept: hotline/protocol\r\n

\r\n

Server's immediate reply to GET is:

HTTP/1.0 200 OK\r\n

Proxy-Connection: Keep-Alive\r\n

Content-Length: 999999999\r\n

Content-Type: hotline/protocol\r\n

\r\n

After this reply, server uses this connection to send data to the client.

Address used in these requests is standard URL address followed by the session ID, specified as the file in the root directory. This is an example of such address:

http://tracker.com:5497/5555-5555-5555

Session ID is used in order to identify the client in the case of disconnect. Host name specified in the HTTP headers is in the form of standard domain name string, followed by the port number. For example:

tracker.com:5497

Data Header

Additional header precedes every transaction that is sent over these two connections. This header has the format:

Description	Size	Data	Note
Data code	4		Disconnect (0), data (1), padding (2)
Data size	4		
Data…	size		

Data content depends on the data code specified. If data code value is 1 (constant is http_Data), data content is transaction data as described in this specification (this includes tracker protocol). Code and size with value 0 (hard-coded constant) notifies the remote end of a pending disconnect.

After predetermined period of inactivity on an HTTP connection, the proxy server can close that link in order to preserve its resources. To prevent this, additional "padding�? data is transmitted, only to keep this connection "alive�?. Data code value 2 (http_Padding) indicates that this is the case. When remote end receives this packet type, its data content is simply discarded.

Global Server

1.1 Server Information

Hotline servers will be able to create an account on the global server by providing a unique *server name* **(relatively short in size) and an** *access password***. This information constitutes an account login information that will have to be provided in every subsequent access to the global server. At the time the account is created, the global server assigns the unique** *server ID* **to the server.**

Global server will provide servers with the ability to store a predefined set of data fields. In addition to the name, the server can also provide region specific *server alias***. Description field can be used to describe the server's content to users. Servers can also be optionally classified into one of the few predefined categories provided by the global server. This will allow users to determine server's content based on a common** *classification* **scheme. An optional** *public encryption key* **can be used to authenticate the server to the users that are connecting to it. Global server will also record server's** *original and current***(last used)** *IP address***.**

Server flags **enable or disable various operations that global server performs.** *Searchable flag* **signals if the server will be included as part of the results of the user's query.** *Rating* **field enables users to rate the server.**

Server status **describes the current availability of server's services.** *On-line status* **indicates that server is currently operational and ready to process requests.** *Active status* **shows that server's account is active, even if the server is not currently on-line. Removing active status indicates that the server can't be made operational in the short term. This can be used when the server is about to go through a non-trivial maintenance process. The server can also specify the** *number of users* **currently connected to it. Global server records** *date* **when account was** *created* **and** *accessed***.**

The following table summarizes the server information stored on the global server:

Data	Options	Note
Server ID		Assigned by the global server
Server name		Used as login
Access password		
Server alias		Region specific alias

Description		
Classification		
Icon		Graphical icon
Rating		
Public encryption key		
IP address	Original Current	Include the port number
Attributes	Searchable Allow rating	
Status flags	Active On-line	Active or not On-line or off-line
Number of users		
Date	Account created Last access	

1.2 Global Server Transactions

1.2.1 Server Login

Initiator: Server

This transaction is used every time the server logins to the global server. It must be the first transaction sent to the global server.

Fields used in the request:

ID	Field Name	Note
	Server name	
	Access password	
	New account indicator	Optional Indicates if this is a new account

Fields used in the reply: None

If server indicates that it creates a new account, and account with identical ID already exists in the database, or if a new account cannot be created for any other reason, the global server indicates these conditions with the proper error code.

1.2.2 Update Server Information

Initiator: Server

Update server information on the global server. All fields in this request are optional.

Fields used in the request:

ID	Field Name	Note
	Access password	
	Server name	
	Server alias	
	Description	
	Classification	
	Icon	
	Attributes	
	Status flags	
	IP port number	Hotline protocol port number
	Number of users	Current number of users

Fields used in the reply: None

1.2.3 Delete Server Account

Access: Administrator

Initiator: Client

Delete server account from the database.

Fields used in the request:

ID	Field Name	Note

	Server name	

Fields used in the reply: None

1.2.4 Rate Server

Initiator: Client

Fields used in the request:

ID	Field Name	Note
	Server name	
	Rating	

Fields used in the reply: None

1.2.5 Query Server Database

Initiator: Client

Create a query for the server database. All fields in this request are optional. If client does not specify the search string, the list of all servers is returned.

Fields used in the request:

ID	Field Name	Note
	Search string	Optional
	Classification	Optional

Fields used in the reply:

ID	Field Name	Note
	Server ID	
	Server ID …	Optional More server IDs

1.2.6 Get Server Information

Initiator: Client

Get information about the specific server.

Fields used in the request:

ID	Field Name	Note
	Server ID	

Fields used in the reply:

ID	Field Name	Note
	Server name	
	Server alias	
	Description	
	Current IP address	Including port number
	Classification	
	Icon	
	Status flags	
	Number of users	

Categories

Contact

FANDOM APPS

Take your favorite fandoms with you and never miss a beat.

Hotline Wiki is a FANDOM Lifestyle Community.

VIEW MOBILE SITE