Message Format and Protocol for the Graphics Server V2

General format of message

A message is composed of three different fields, a sync character, the message length, and the payload, as shown in Figure 1 below.

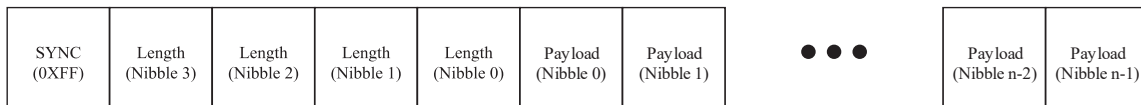| SYNC (0XFF)	Length (Nibble 3)	Length (Nibble 2)	Length (Nibble 1)	Length (Nibble 0)	Payload (Nibble 0)	Payload (Nibble 1)	● ● ●	Payload (Nibble n-2)	Payload (Nibble n-1)

Figure 1: Overall message format.

Every message must begin with a SYNC byte of 0xFF. If at any time, even in the middle of a message, if a SYNC byte is transmitted, the GraphicsServer will recognize this as the beginning of a new message and discard anything that was partially received.

All other data in the message is transmitted as bytes, but only the 4 least significant bits are relevant. If any byte contains data with the upper 4 bits set (unsigned characters with values greater than 15) then the message is aborted and the graphics server begins looking for a SYNC character.

The four bytes after the SYNC character is the total length of the payload (n in Figure 1). Since each Length character only uses the lower four bits, the maximum length of the Payload is 16,535 characters. For example, the message that begins with 0xFF, 0x00, 0x00, 0x05, 0x06 would have a Payload length of 0x0056 = 86 bytes.

The first byte of the payload defines the different graphics commands that you can send. These are listed below.

Char value	Mnemonic	Description
1	CLEAR	Clear the display and set every pixel to the background color
2	SET_BACKGROUND_COLOR	Set the background color to the specified color
3	SET_PIXEL	Set the specified pixel location to the specified color
4	UNUSED	
5	DRAW_STRING	Draw the specified sting at a specified location
6	SET_DRAWING_COLOR	Set the drawing color for text, shapes, and pixels
7	DRAW_RECTANGLE	Draw the specified rectangle at the specified location
8	FILL_RECTANGLE	Draw and fill the specified rectangle at the specified location
9	CLEAR_RECTANGLE	Clear the specified rectangle at the specified location (set the region to the background color)
10	DRAW_OVAL	Draw the specified oval at the specified location
11	FILL_OVAL	Draw and fill the specified oval at the specified location

12	REPAINT	Cause the graphics server to update the display
13	DRAW_LINE	Draw the specified line
14	FILE_REQUEST	Request a file from the graphics server
15	UNUSED	

The detailed format for each of these messages is given below. Note that the message does not include the SYNC character and the message length. It only gives the Payload part of the message. For example, to draw a line from pixel at (0, 0) (top left of the display) to the pixel at (45, 100) the following message (in char) would be sent to the server.

0xFF, 0x00, 0x00, 0x01, 0x01, 0x0D, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x02, 0x0D, 0x00, 0x00, 0x06, 0x04

CLEAR:

CLEAR (0x01)

SET_BACKGROUND:

SBC (0x02)	Red (Nibble 1)	Red (Nibble 0)	Green (Nibble 1)	Green (Nibble 0)	Blue (Nibble 1)	Blue (Nibble 0)

SET_PIXEL:

SBC (0x02)	x coord (Nibble 3)	x coord (Nibble 2)	x coord (Nibble 1)	x coord (Nibble 0)	y coord (Nibble 3)	y coord (Nibble 2)	y coord (Nibble 1)	y coord (Nibble 0)	Red (Nibble 1)	Red (Nibble 0)	Green (Nibble 1)	Green (Nibble 0)	Blue (Nibble 1)	Blue (Nibble 0)

DRAW_STRING:

DS (0x05)	x coord (Nibble 3)	x coord (Nibble 2)	x coord (Nibble 1)	x coord (Nibble 0)	y coord (Nibble 3)	y coord (Nibble 2)	y coord (Nibble 1)	y coord (Nibble 0)	Character (Nibble 1)	Character (Nibble 0)	●●●	Character (Nibble 1)	Character (Nibble 0)

SET_DRAWING_COLOR:

SDC (0x06)	Red (Nibble 1)	Red (Nibble 0)	Green (Nibble 1)	Green (Nibble 0)	Blue (Nibble 1)	Blue (Nibble 0)

DRAW_RECTANGLE:

DRAWRECT (0x07)	x coord (Nibble 3)	x coord (Nibble 2)	x coord (Nibble 1)	x coord (Nibble 0)	y coord (Nibble 3)	y coord (Nibble 2)	y coord (Nibble 1)	y coord (Nibble 0)	width (Nibble 3)	width (Nibble 2)	width (Nibble 1)	width (Nibble 0)	height (Nibble 3)	height (Nibble 2)	height (Nibble 1)	height (Nibble 0)

FILL_RECTANGLE:

FILLRECT (0x08)	x coord (Nibble 3)	x coord (Nibble 2)	x coord (Nibble 1)	x coord (Nibble 0)	y coord (Nibble 3)	y coord (Nibble 2)	y coord (Nibble 1)	y coord (Nibble 0)	width (Nibble 3)	width (Nibble 2)	width (Nibble 1)	width (Nibble 0)	height (Nibble 3)	height (Nibble 2)	height (Nibble 1)	height (Nibble 0)

CLEAR_RECTANGLE:

CLRRECT (0x9)	x coord (Nibble 3)	x coord (Nibble 2)	x coord (Nibble 1)	x coord (Nibble 0)	y coord (Nibble 3)	y coord (Nibble 2)	y coord (Nibble 1)	y coord (Nibble 0)	width (Nibble 3)	width (Nibble 2)	width (Nibble 1)	width (Nibble 0)	height (Nibble 3)	height (Nibble 2)	height (Nibble 1)	height (Nibble 0)

DRAW_OVAL:

DRAWOVAL (0xA)	x coord (Nibble 3)	x coord (Nibble 2)	x coord (Nibble 1)	x coord (Nibble 0)	y coord (Nibble 3)	y coord (Nibble 2)	y coord (Nibble 1)	y coord (Nibble 0)	width (Nibble 3)	width (Nibble 2)	width (Nibble 1)	width (Nibble 0)	height (Nibble 3)	height (Nibble 2)	height (Nibble 1)	height (Nibble 0)

FILL_OVAL:

FILLOVAL (0xB)	x coord (Nibble 3)	x coord (Nibble 2)	x coord (Nibble 1)	x coord (Nibble 0)	y coord (Nibble 3)	y coord (Nibble 2)	y coord (Nibble 1)	y coord (Nibble 0)	width (Nibble 3)	width (Nibble 2)	width (Nibble 1)	width (Nibble 0)	height (Nibble 3)	height (Nibble 2)	height (Nibble 1)	height (Nibble 0)

REPAINT:

REPAINT (0x0C)

DRAW_LINE:

FILLOVAL (0xD)	x coord (Nibble 3)	x coord (Nibble 2)	x coord (Nibble 1)	x coord (Nibble 0)	y coord (Nibble 3)	y coord (Nibble 2)	y coord (Nibble 1)	y coord (Nibble 0)	x1 coord (Nibble 3)	x1 coord (Nibble 2)	x1 coord (Nibble 1)	x1 coord (Nibble 0)	y1 coord (Nibble 3)	y1 coord (Nibble 2)	y1 coord (Nibble 1)	y1 coord (Nibble 0)

REQUEST_FILE:

REQUEST FILE (0x0E)

Return Messages

Two types of messages may be sent back to the client: Mouse message and file selection message. These have the same general format as the message that may be sent to the graphics server, and are detailed below. In the mouse messages below, the BUTTON NUMBER is 1 for left button, 2 for middle button, and 3 for right button. The x and y coordinates are the x and y coordinates of where the mouse is when the button action occurs. A click is only sent if there is a press and release action within a certain amount of time. (operating system dependent)

MOUSE_MESSAGE BUTTON PRESS:

DOWN (TYPE 1)	BUTTON NUMBER	x coord (Nibble 3)	x coord (Nibble 2)	x coord (Nibble 1)	x coord (Nibble 0)	y coord (Nibble 3)	y coord (Nibble 2)	y coord (Nibble 1)	y coord (Nibble 0)

MOUSE_MESSAGE BUTTON RELESE:

UP (TYPE 2)	BUTTON NUMBER	x coord (Nibble 3)	x coord (Nibble 2)	x coord (Nibble 1)	x coord (Nibble 0)	y coord (Nibble 3)	y coord (Nibble 2)	y coord (Nibble 1)	y coord (Nibble 0)

MOUSE_MESSAGE BUTTON CLICK:

CLICK (TYPE 3)	BUTTON NUMBER	x coord (Nibble 3)	x coord (Nibble 2)	x coord (Nibble 1)	x coord (Nibble 0)	y coord (Nibble 3)	y coord (Nibble 2)	y coord (Nibble 1)	y coord (Nibble 0)

FILE_PATH_MESSAGE:

FILEPATH (0x0E)	Character (Nibble 1)	Character (Nibble 0)	● ● ●	Character (Nibble 1)	Character (Nibble 0)