

Getting Started or eEMU Tutorial

Almost in every book on a programming language, the phrase "Hello World" finds its way to the first examples. How about starting with it in our tutorial on "eEMU messaging language".

Assumption is made that whoever got as far as this article successfully installed eEMU, got it up and running and perhaps transformed xeb screen from boring grey to nice rainbow-like window by sending a few messages...

If you just installed eEMU, do not worry. You will see what can be done to your xeb screen by reading through a few simple examples. Let's get started...

Assumptions are that the eEMU server is installed on host localhost, listens on port 2345 and the password is icecream.

Example 1 (normal message):

Let us send a "normal" message with a low priority of 4 (rendered by the browser in grey background) to class EVERYBODY. Furthermore, let us make the message "sit" in the database (and xeb) for 30 minutes. The message text is "Hello World !"

```
emsg1 -n localhost -p 2345 -w icecream -t 30m -s 4 -c EVERYBODY -m 'Hello World !'
```

This is what will appear in your xeb:



Example 2 (time-to-live):

Here we go, the first message is there. But you may think that a message should stay in eEMU generally longer than 30 minutes, for this would be just a one-off message (though not of high importance). Anyway, let's make it sit in eEMU for 2 hours - long enough for everyone possibly interested to notice ...

```
emsg1 -n anna -p 2345 -w icecream -t 1h -s 4 -c EVERYBODY -m 'Hello World !'
```

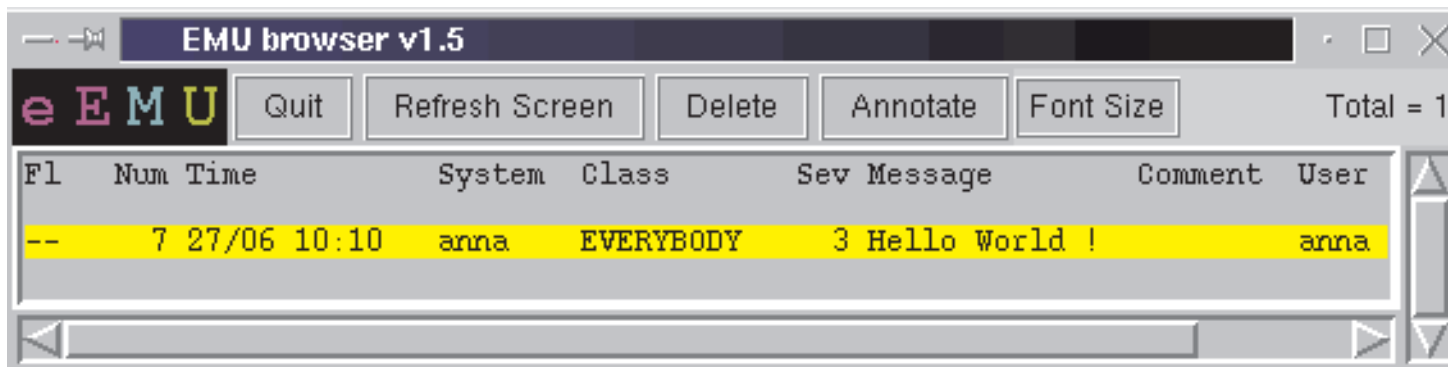
And this is your xeb now:



Example 3 (severity):

Quite good. "Hello World !" should sit in eEMU (xeb) for the next 2 hours. But will actually someone take notice of the message since the background in xeb is grey and even the bold font is gone (the message is displayed in bold only when it arrives the first time and every refresh is rendered in a "normal" font). To make it more "visible" let's raise the priority (severity) of the message from 4 to 3 . (xeb renders the background of the message based on its severity: 3 - yellow, 2 - orange, 1 - red).

```
emsg1 -n localhost -p 2345 -w iccream -t 1h -s 3 -c EVERYBODY -m 'Hello World !'
```



Note: As the message was just refreshed, its lifetime has also been "reset" to 1 hour from now on.

Example 4 (ObjectID):

OK. But how about sending a "Hello" message to another addressee - more specific than "World" - e.g. to Jarra. Can we do that?

Of course we can, but since the first string of the message (also called objectID) is significant for eEMU (to distinguish between messages), we have to either:

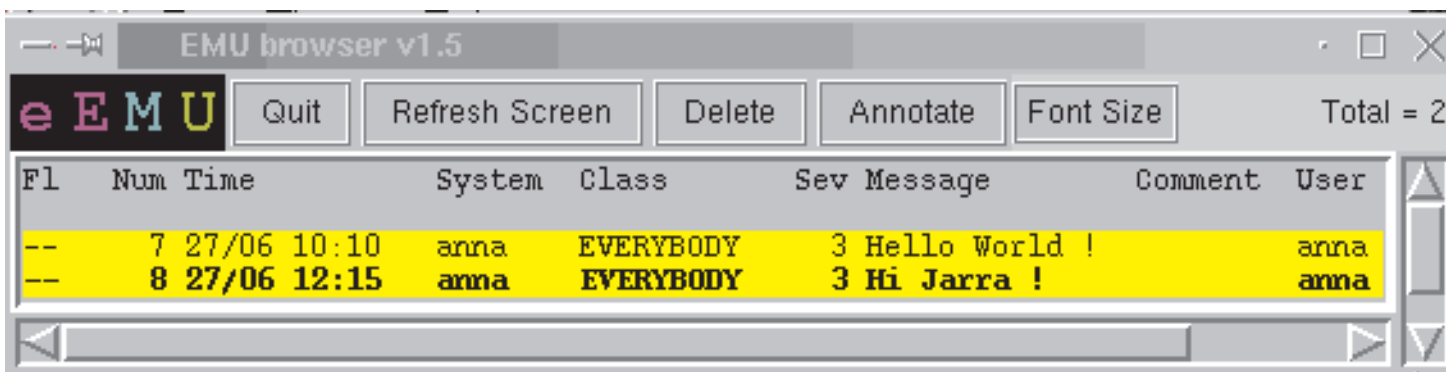
- change order of the greeting e.g. "Jarra Hello"
- change case of the first word e.g. "HELLO Jarra"
- change the greeting e.g. "Hi Jarra"

```
msg1 -n localhost -p 2345 -w icecream -t 1h -s 3 -c EVERYBODY -m "Jarra Hello"
```

```
msg1 -n localhost -p 2345 -w icecream -t 1h -s 3 -c EVERYBODY -m "HELLO Jarra"
```

```
msg1 -n localhost -p 2345 -w icecream -t 1h -s 3 -c EVERYBODY -m 'Hi Jarra !'
```

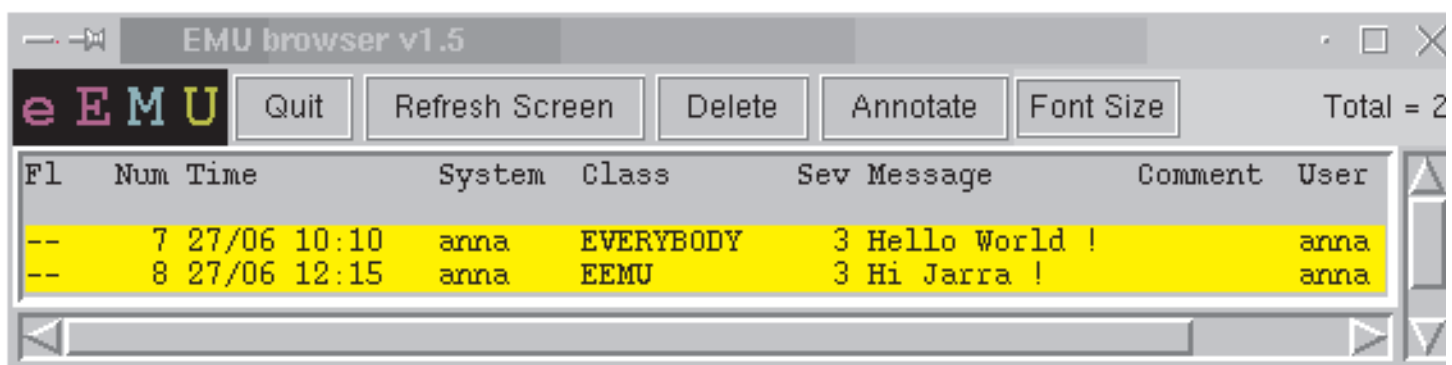
Let's try the third option:



Example 5 (class):

There are 2 messages in the eEMU database (and xeb screen). Both messages are addressed to class "EVERYBODY". Let's try to change this as well. If the class of the second message is changed to "EEMU" (for eEMU fans), we can filter messages on Jarra's browser to display messages addressed to EEMU class only.

```
msg1 -n localhost -p 2345 -w icecream -t 1h -s 3 -c EEMU -m 'Hi Jarra !'
```

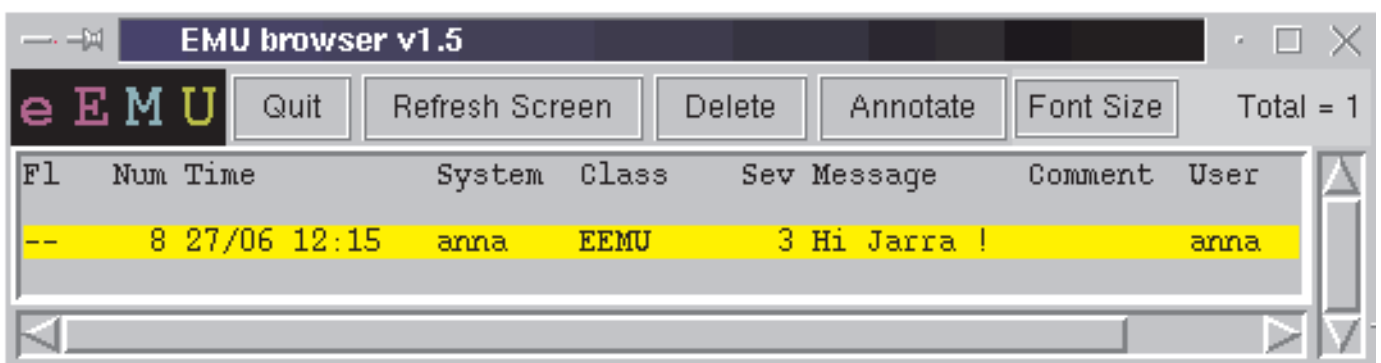


Example 6 (EMUSELECT):

Now, we can filter messages based on the class. To make it simple, let's assume that Jarra runs his browser (xeb) on the same host that runs the eEMU server. To filter messages, the EMUSELECT env. variable must be set.

```
export EMUSELECT=" grep EEMU "
```

Subsequently, Jarra's browser is restarted:



Example 7 (Delete):

Jarra saw our nice greeting and has decided to delete the message;

How did he do it ? There are 2 possible ways:

1) through the browser - Delete button:



2) using emsg1:

```
.emsg1 -o delete -n localhost -p 2345 -w icecream -m "anna:Hi"
```

Note: notice the syntax of the delete message. ObjectID following the "m" switch consists of the host sending the message (anna) and the first word of the message itself (Hi) delimited by a colon (:). **anna:Hi**

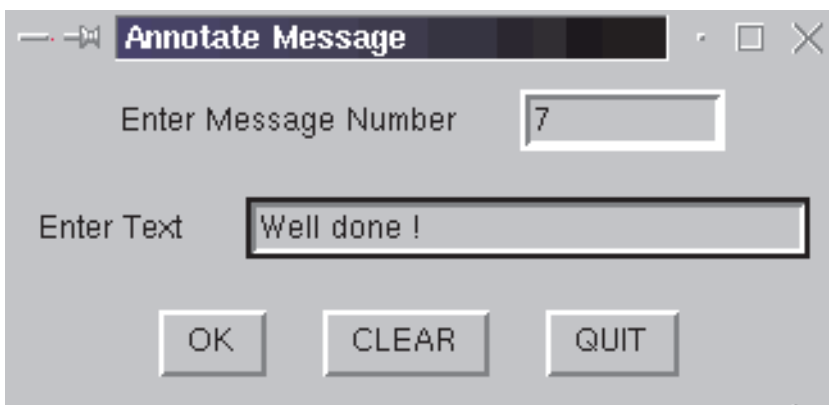
Example 8 (Comment):

Back to our very first message. Anna decided to send us an annotation to the message. The annotation text read "Well done !". She had two options to achieve it:

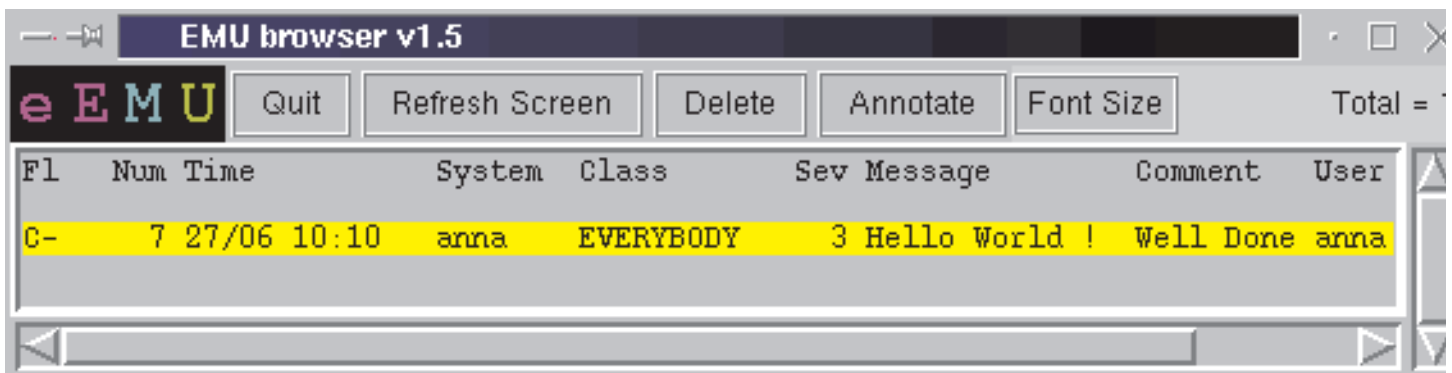
1) using emsg1:

```
./emsg1 -o comment -n localhost -p 2345 -w icecream -m "anna:Hello Well Done "
```

2) through the browser - the Annotate button:



And the comment appears in the "Comment" field:



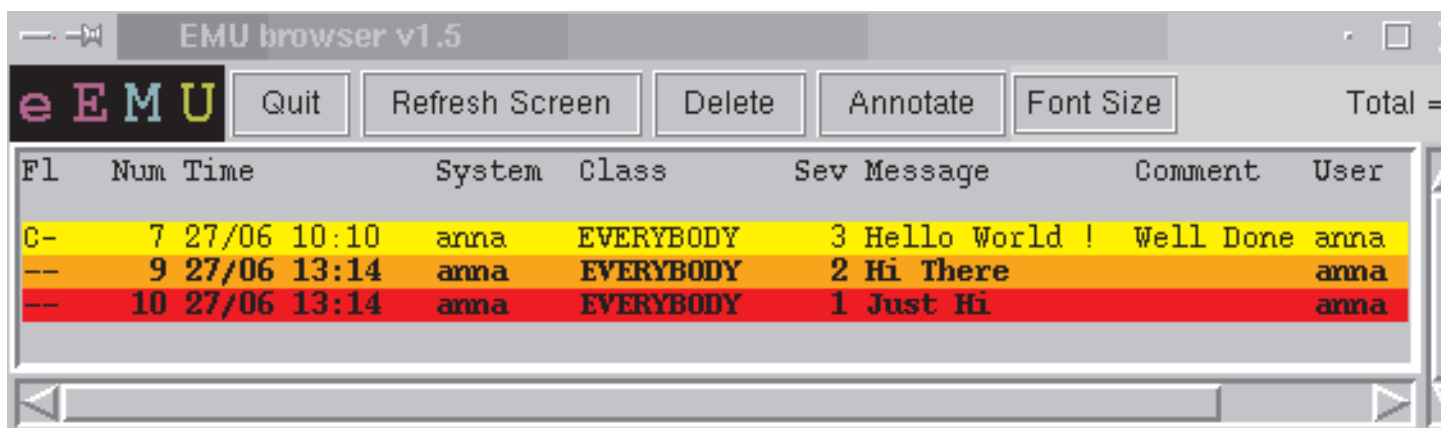
Note: notice the syntax of the "comment" message. ObjectID must be specified at the beginning of the "-m" switch followed by the actual text of the comment: **"anna:Hello Well Done "**

Example 9 :

This is just to leave a nice "rainbow" xeb screen behind

```
emsg1 -n localhost -p 2345 -w icecream -t 15m -s 2 -c EVERYBODY -m "Hi There"
```

```
emsg1 -n localhost -p 2345 -w icecream -t 5 -s 1 -c EVERYBODY -m "Just Hi"
```



The screenshot shows a window titled "EMU browser v1.5". The window has a menu bar with "e E M U" and several buttons: "Quit", "Refresh Screen", "Delete", "Annotate", "Font Size", and "Total =". Below the menu bar is a table with the following columns: "Fl", "Num", "Time", "System", "Class", "Sev", "Message", "Comment", and "User". The table contains three rows of data, each with a different background color: yellow, orange, and red.

Fl	Num	Time	System	Class	Sev	Message	Comment	User
C-	7	27/06 10:10	anna	EVERYBODY	3	Hello World !	Well Done	anna
--	9	27/06 13:14	anna	EVERYBODY	2	Hi There		anna
--	10	27/06 13:14	anna	EVERYBODY	1	Just Hi		anna