

## **New Xpress Program Editor Debugger**

By Kevin Jepson

Second Beta Release

August 5 2011

### Enhanced Editor Features

- Verb and Property lists are ordered alphabetically
- Syntax Highlighting of verb code and property values
- Line numbers are no longer part of the text and they adjust automatically for insertions and deletions
- Search and Search/Replace function inside editor
- Goto line number inside editor
- Expand to full size of frame, font adjustment to aid in reading longer verbs
- Optional Word Wrap of long lines
- Toggle to enable or disable editor functions
- Property editor now shows actual value of property and DOES NOT MUNGE LISTS!
- Work around for "Buy 1 Get 1 Free" new verb issue

Most functionality is from the Edit Area javascript library by Christophe Dolivet

<http://www.cdolivet.com>

### Integrated Debugger Features

- Integrated debugger accessible from Verb Editor
- Analyses the verb code and generates a selectable list of variables
- Allows insertion of checkpoints for selected variable values and arbitrary text
- Allows insertion of breakpoints for selected variable values and arbitrary text that will halt execution and display the text and variable values. Task execution can be resumed or killed.
- Displays and automatically updates a debug log on the programmer

Thanks to Paul Rayner at Muddle for allowing me to test this out on his live system!

Enjoy

Ciao

KJ

**SCREENSHOTS of the Program Editor with Edit\_Area Javascript code in action!**  
**(Note debugger not shown see next section)**

# Xpress Program Editor

[View](#)[My Objects](#)[Programmer's Manual](#)[Close](#)

## Editing verb "lw" on Kevin

Name 

Owner: Kevin (#179)

Permissions

 Read  Execute  Debug

Arguments

```
1 "Usage: lookw object or lw object";
2 "Creates a URL for the object.";
3 "It will be displayed in a web browser window if the
4 client is using MOOTCAN.";
5 "Currently will not display objects inside containers.";
6 "player:tell(dobjstr)";
7 LOC = player.location;
8 dobj = LOC:match_object(dobjstr);
9 data = dobj;
10 if (dobjstr == "")
11     data = LOC;
12 endif
13 if (!valid(data))
14     player:tell("There is nothing with that name visible
15 here.");
16 return;
17 endif
```

Position: Ln 1, Ch 1

Total: Ln 27, Ch 744

 Toggle editor[Compile Verb](#)[Delete This Verb](#)

## Verbs

[lw](#)  
[object\\_list](#)  
[output\\_data](#)  
[recurse](#)  
[set-code](#)  
[tell](#)  
[test](#)  
[webcon](#)

## Properties

[\\_mail\\_task](#)  
[accessed](#)  
[aliases](#)  
[all\\_connections](#)  
[\\*\\*](#)  
[allow\\_override](#)  
[at\\_number](#)  
[at\\_room](#)  
[audio\\_url](#)  
[banner](#)  
[brief](#)  
[build\\_options](#)  
[current\\_file](#)  
[current\\_r](#)

Easy on the eyes!

Note Verb and Property lists are sorted alphabetically.

## Editing verb "lw" on Kevin

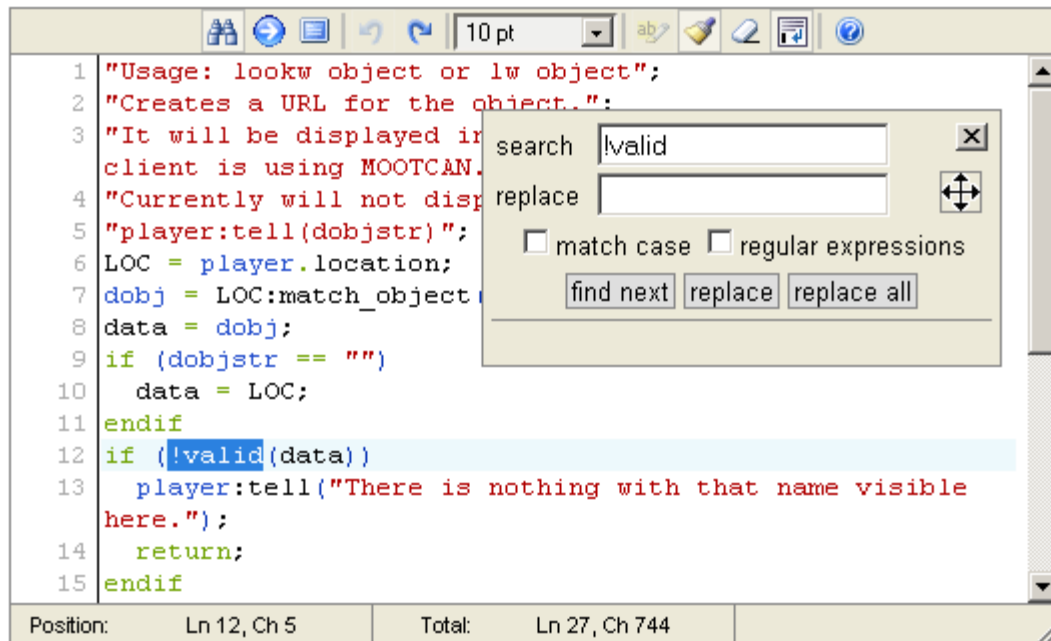
Name

Owner: Kevin (#179)

Permissions

Arguments

Read  Execute  Debug



```
1 "Usage: lookw object or lw object";
2 "Creates a URL for the object.";
3 "It will be displayed in
client is using MOOTCAN.
4 "Currently will not disp
5 "player:tell(dobjstr)";
6 LOC = player.location;
7 dobj = LOC:match_object;
8 data = dobj;
9 if (dobjstr == "")
10     data = LOC;
11 endif
12 if (!valid(data))
13     player:tell("There is nothing with that name visible
here.");
14     return;
15 endif
```

Position: Ln 12, Ch 5      Total: Ln 27, Ch 744

Toggle editor

**Compile Verb**

**Delete This Verb**

Search button brings up a search replace dialog

# Xpress Program Editor

The screenshot shows the Xpress Program Editor interface. At the top, there is a title bar with the text "Xpress Program Editor". Below the title bar, there is a toolbar with a search icon, a "me" text box, and buttons for "View", "My Objects", "Programmer's Manual", and "Close".

The main area is a code editor with a toolbar at the top containing icons for undo, redo, and font size (9 pt). The code editor contains the following code:

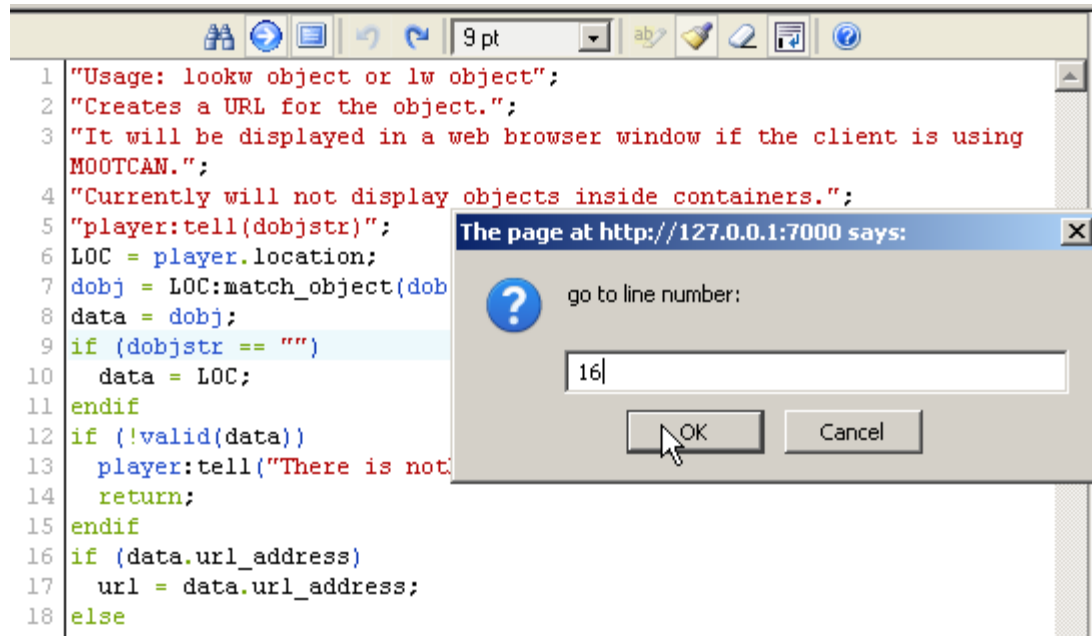
```
1 "Usage: lookw object or lw object";
2 "Creates a URL for the object.";
3 "It will be displayed in a web browser window if the client is using
MOOTCAN.";
4 "Currently will not display objects inside containers.";
5 "player:tell(dobjstr)";
6 LOC = player.location;
7 dobj = LOC:match_object(dobjstr);
8 data = dobj;
9 if (dobjstr == "")
10     data = LOC;
11 endif
12 if (!valid(data))
13     player:tell("There is nothing with that name visible here.");
14     return;
15 endif
16 if (data.url_address)
17     url = data.url_address;
18 else
19     base_url = $encore_web_utils:baseurl();
20     object = toint(data);
21     url = base_url + tostr(object);
22 endif
23 player:tell(" ");
24 player:tell("You view ", data.name, "...");
25 player:tell(" <", url, ">.");
26 "Last modified Tue Mar 22 19:49:14 2011 Mountain Daylight Time by
Kevin (#179).";
27
```

At the bottom of the code editor, there is a status bar with the following information:

Position:	Ln 9, Ch 7	Total:	Ln 27, Ch 744
-----------	------------	--------	---------------

On the right side of the editor, there is a sidebar with two sections: "Verbs" and "Properties". The "Verbs" section contains a list of verbs: [lw](#), [object\\_list](#), [output\\_dir](#), [recurse](#), [set-code](#), [tell](#), [test](#), and [webcon](#). The "Properties" section contains a list of properties: [\\_mail\\_task](#), [accessed](#), [aliases](#), [all\\_connections](#), [\\*\\*](#), [allow\\_override](#), [at\\_number](#), [at\\_room](#), [audio\\_url](#), [banner](#), [brief](#), [build\\_options](#), [current\\_file](#), and [current\\_name](#).

Full Screen button expands the text box to fill the whole frame.



Goto Line Number

## Editing property "test" on Kevin

Name

Owner: Kevin (#179)

Data Type: List

Size: 96 bytes

Permissions:  Read  Change

Property Value:

```
1 | {34.5, 34, #34, "34", {"open", 1}}
```

Toggle editor

**Save Property**

**Delete This Property**

## Verbs o

[lw](#)  
[object\\_list](#)  
[output\\_data](#)  
[recurse](#)  
[set-code](#)  
[tell](#)  
[test](#)  
[webcon](#)

[sound\\_ve](#)  
[subtitle\\_fr](#)  
[table\\_bg](#)  
[table\\_bor](#)  
[table\\_lay](#)  
[table\\_ver](#)  
[table\\_wid](#)  
[test](#)  
[Text\\_Align](#)  
[thing\\_arri](#)  
[thing\\_por](#)  
[title\\_font](#)  
[title\\_font](#)  
[title\\_table](#)  
[title\\_table](#)  
[ts\\_client](#)

Property editor opens with insert cursor at start of the line and does not add a spurious linefeed. Displays nested lists properly and does not munge them into strings on saving.

## SCREENSHOTS of the Program Editor Debugger in action!

The debugger is accessed by clicking the Debugger button below the verb code display.

```
Position: Ln 1, Ch 1 Total: Ln 9, Ch 297
```

Toggle editor

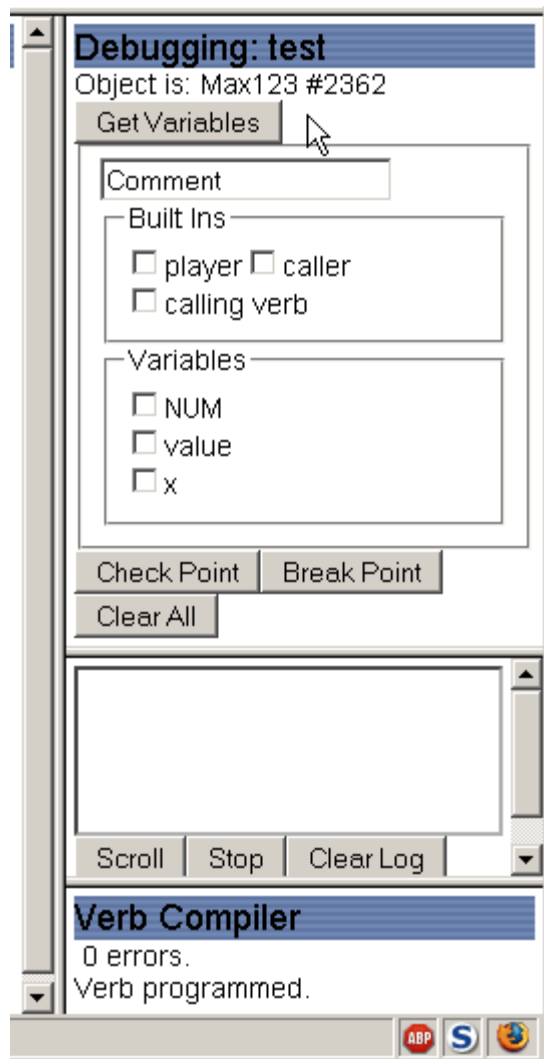
**Compile Verb**

**Debug This Verb**

**Delete This Verb**

Done

Once activated the debugger changes the right hand side to display three frames. The top one contains a control panel for the debugger. the middle one displays the current contents of the debug log and the third displays notifications and confirmations from the system.



Clicking the "Get Variables" button will analyse the code in the editor and list the variables found. Selecting the variables to be watched and then selecting "Check Point" and/or "Break Point" will insert a code line in at the current cursor position that will output the values of the selected variables and comment text to the debug log when the verb is executed. Arbitrary text can be added by entering it in the "Comment" box.

Here is an example showing the code with the addition of a checkpoint at line 5 and a breakpoint at line 8.

**Xpress Program Editor**

me View My Objects Programmer's Manual Close

---

**Editing verb "test" on Max123**

Name: test Owner: Max123 (#2362)

Permissions:  Read  Execute  Debug Arguments: this none none

```

1  "A general test verb for checking code stuff";
2  "Place code to be tested below :-) ";
3  x = $string_utils:explode("hello 07:51:28 goodbye");
4  value = x[2][1..2];
5  $bp:cp(#2362,#2362,"test","Testing for example",
6  {"player",player},{"value",value});
7  player:tell("value is ", value);
8  NUM = tonum(value);
9  $bp:bp(#2362,#2362,"test",task_id(),"Testing for
10 example 2",{"calling verb",verb});suspend();
11 player:tell("num is ", NUM);
12 "Last modified Sun Nov 27 15:58:23 2011 EST by Max123
13 (#2362).";

```

Position: Ln 1, Ch 1 Total: Ln 11, Ch 477

Toggle editor

Compile Verb Debug This Verb Delete This Verb

Done

**Debug**

Object is:

Get Var

Testing

Built

pl

ca

Varia

NU

va

X

Check F

Clear Al

Scroll

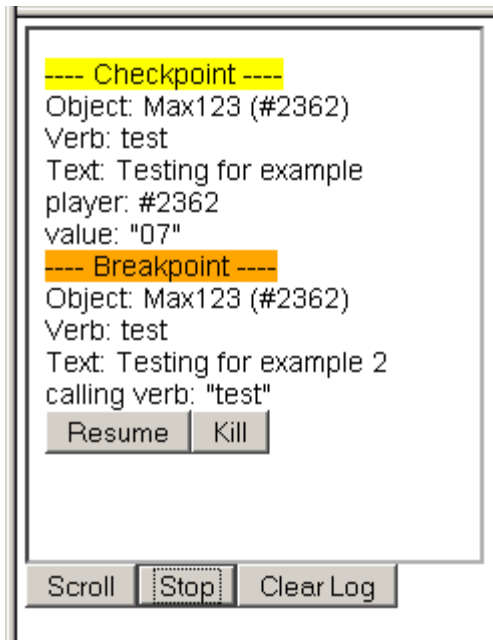
**Verb C**

0 errors.

Verb pro

When this code is executed the debug log display would look like this.





Checkpoints do not interrupt the execution of the verb whereas a breakpoint will suspend the current execution task.

Clicking on "Resume" will allow the verb to continue execution while clicking "Kill" will end the task.

"Clear Log" will delete the contents of the user's debug log.

The debug log automatically scrolls to the latest entry, clicking "Stop" will allow the user to scroll back in the log to see previous entries.