X2Web version 2.1

http://www.sentman.com/x2web
html preprocessor adding tags to directly access data from XTension
http://www.shed.com/

New in 2.1
Improved Plug and Play. The instantx code will now only share lists from XTension that end in ".web" so any lists you want to be displayed in the instantx pages you must rename, but you dont have to share all your lists.

Browse and search the log file. The instantx pages add the ability to configure log files for display. Run X2Web and open the "Log Files" window from the file menu. You can select any number of log files you want to be available in instantx here. There is also the option of using regex to search through the files. This is the same engine for regex that BBEdit uses, so if you're familiar with it's "use grep" option the same code will work in X2Web. For example: if you want to just to 3pm to see what was happening then click the "use regex" button and enter:

3:/d/d:/d/d PM

This will match the first 3pm and then you can use the 'find next' button to search on from there.

Improved Security. The list pages will now only allow you to control units that are actually on the list being displayed. So even if someone discovers your other unit names from the log they will not be able to control them illegally.

NEW in 2.0
Plug and Play. Included with this version is the instantx folder which will get you up and running almost instantly. You still need to do the web server setup, but then just place that folder inside your web root and hit the index.x10 page in that folder. You'll be presented with a list of lists from XTension, select one and you'll get a list of units which you can control.

Database! You can log data points, anything really but right now the only thing I have a template to build a report from is motion sensor data. See the database section below for details on setting this up. You'll also need the motion report plugin file that is included with the package. Unlike the half

release I did of this version it now contains the ability to remove data from the database as well as add it. I've also included a template to use the plugin to give you really easy access to this data, but you will need to edit the html code yourself to include the proper unit names.

First, You'll want to turn logging on and off without too much hassle so create a pseudo unit in XTension called "log to database" no X10 code, it's just going to be used as a flag to enable talking to the database. Don't skip this, it's important otherwise you're not going to be able to quit X2Web as every time there is an event the script will try to re-launch it.

Second, add this handler to your attachments script:

```
on logdata(unitname, unitvalue)
	if (value of "log to database") is 100 then
		ignoring application responses
			tell application "x2web"
				«event x2wblogd» unitname given «class valu»:unitvalue
			end tell
		end ignoring
	end if
end logdata
```

now you have a new applescript command that you can call from the ON script of any motion sensor to log that data.
Third, add some code like this to the ON scripts of your motion sensors:

```
logdata("living room motion", 100)
```

Of course replacing the name with your unit name. For making a report, see the motion report template. It offered 2 kinds of reports. A simple yes/no report and a more complex one that give a color scale of green for no hits, yellow for some and red for a lot. See them both in the templates.

My database has a lot of units in it and has been up solidly for almost 2 months and the reports are starting to slow down, taking 3 or 4 seconds each to come up and a page with 4 or 5 reports on it is slow. So after a certain amount of time you'll want to delete data from the database. You can do this with a nightly, or weekly script in XTension like this for each of your units that is logging:

```
tell application "x2web"
	delete data before ((current date) - 30 * days) unit "living room motion"
end tell
```

This will delete any data older than 30 days from the database. At some point I will do some archiving scripts so that we can keep the data for a year

or longer, but there is no point to have the data being generated live at that point.

Looping. for people doing more detailed lists and such from apple scripts or plugins you can now define a loop like this:

```
<XLoop name="list of people">
   <li>[1] is [2] years old
</XLoop>
```

And then access it in applescript. You pass a list of lists with 2 fields (or more if you have more tags, the tags in this case can be [1] or <1> which ever makes more sense for your particular application and you can mix them) and the program will replace them into the loop. This is much easier and faster than doing a lot of concatenation within applescript as it only involves a single event and less data.

```
<AppleScript>
  set MyList to {{"james", "32"}, {"Jessica", "1.5"}}
  doList( "list of people", MyList)
</AppleScript>
```

would create the output

```
 <li>James is 32 years old
 <li>Jessica is 1.5 years old.
```

There are also special list types for getting a list of lists from XTension and a list of units in a list. You can see these in action in the plug and play pages.

Little bit faster still: This version can now get the time delta and timestamp data out of the preload event so loading pages with date data from sensors and such will be much faster.

NEW in 1.3:

Takes advantage of the new extract verbs in XTension. In order to greatly speed up your page displays, especially in complex pages do this: Make a new list in XTension and place the units that you are going access from the page into this list. Then at the beginning of your page place this tag:

<XTension param="preload" list="the name of your list">

This will load all of the units information from that list in a single apple event and the more units accessed in your page the more speed increase you will see with this, but it is worth doing even for just a few units.

You can include other files into your page. They will be processed as if they were a separate .x10 file and then inserted. Very useful for repeating menu's, headers, footers or other such things.

<XTension param="include" file="/control/menu.inc">

the file link is relative to the web server root, you can only include files from inside the web server folder. It doesn't matter what the extension of the file is it will always be processed as if it was a .x10 file. These files are processed and cached the same way that a regular .x10 file is, so don't worry about much overhead or extra disk access from using this feature

You can conditionally include html and xtension tags with <if><else></if> tags. The logic is applescript and just needs to result in a true or false. for example, to send specific code to just one machine by IP address so something like:

```
<if eval="getFormVariable( \"kcip\") is \"10.0.1.196\"">
  Ah, you are connecting from the Audrey in the bedroom
<else>
  normal, non custom controls
</if>
```

Since the eval="" needs to be enclosed in quotes, any other double quotes you need to include need to be escaped with a slash. The getFormVariable("kcip") will return the IP address of the browser. This is the code that is set by the server, there are a bunch of other values you can get here, if you

need to know what they are before I get a chance to document it drop me an email.

New formatting options for the date displays. For TimeDelta like:

<XTension param="time delta" unit="furnace" format="short">

valid formats for time delta are short (03:12:21) and long (5 hours, 7 minutes and 15 seconds) without a format parameter it just returns raw seconds.

<XTension param="timestamp" unit="furnace" format="short time">
for timestamp the values are: long time, long date, short time, short date, abbreviated date and with nothing it returns a very long date.

NEW in 1.2:
Many new actions see action section below. Smarter data collection for the units should be a little bit faster.

NEW in 1.1:
Adds 10 second timeouts to all events and fixed the remote targeting. At least it works for me now both with the IP address or the AppleTalk name of the remote server. Scripts are now compiled when you run the program the first time and then stored compiled for faster loading and will run even if the program they are trying to contact isn't running at the time. Unlike the previous version that required any dependent programs to always be running when x2web launched. You can edit the plugins while the program is running, save them and then select 'reload plugins' from the file menu. The program will know you have altered the code and recompile it. Keep in mind that this invalidates the cache of pages stored, so they will have to be re-parsed the next time they are requested, but that generally doesn't take more than a fraction of a second anyway.

**INSTALLING WITH PERSONAL WEB SHARING:

in pws's control panel preferences screen switch to the 'Actions' tab.

Create a new action. Set it's popup to "Launch At Suffix".

Click the "select" button and locate the X2Web program. I don't know if this has to be inside the server root folder or not, mine is and it does work from there.

In the suffix field write ".x10" or whatever file type you want processed by program. If you put .html in there then all the files you serve can be processed by it.

**INSTALLING WITH QUID PRO QUO

this is the server that I personally use as it has much better performance and reliability and it's also free.

Open the "Server Settings" window from the "Control" menu and select the "actions" icon.

Click the "New Action" button.

Give the action a name, I used "x2web"

in the path field fill in the path to the x2web program. In this case I believe it does have to be inside of the server folder. If you put it just in the server root you should enter "/x2web"

Switch to the "Suffix Mappings" panel

Click the "New Mapping" button.

In the suffix field enter the file suffix that you want processed by the program. I used ".x10"

for the Type and Creator fields enter "*"

for the "MIME Type" enter "text/html"

and from the action popup menu you should select "x2web" or whatever you called it in the first step above.

Lastly you can select "Quid Pro Quo" from the preferences screen on the program and check the "connect to XTension on this machine" checkbox which should be the default but seems to not be on a new install.

Other web servers are more likely to be setup similarly to QuidProQuo than pws.

**THE XTENSION TAG

the program adds a tag to get information from XTension. You can put any number of these tags into a document that is setup to be processed with the program and it will replace the tag with the appropriate data from XTension.

To display the on/off status of a unit use a tag like:

<XTension param="status" unit="my light">

replace "my light" with any unit name in extension. There are other Param values to return other data:

value : returns 0 - 100 for the percentage on of the lamp or whatever the number is in the case of a pseudo unit.

timestamp : returns the date and time of the last activity of the unit

description : returns the description of the unit

for some units it doesn't make sense to return on or off so the "status" param has another parameter you can send to tell it to return different values. For example, If you have a unit "garage door" that is set by a switch on the door, it doesn't make sense to say that it is on or off so you would do something like:

<XTension param="status" unit="garage door" return="open:closed">

This tag will now return open or closed instead of on and off. you can put any 2 values separated by a colon. For the fancy display people you could use this to display a different icon depending on the status of a lamp by doing something like this:

<img src="<XTension param="status" unit="garage door" return="/open_icon.gif:/closed_icon.gif">">

The extension parameter would be replaced by the appropriate link text to the icon of your choice.

**CONTROLING XTENSION

Every page parsed by the program acts like it's own cgi program. You can send it form data in get or post parameters from a form, or just as extra data in a regular link. The program makes it easier to link back to the same page by providing you with a [thislink] tag. If you use this instead of hard coding the link you can re-use code or re-name pages without having to alter the html code. You need to pass 3 values for most commands. "unit" and "action" and "value" unit is the name of the unit action tells it what you want to do. For example:

Currently, valid actions are:
on, off, dim, xpress, toggle, removeunitevents and speak.
Version 1.2 adds: block, unblock, doScript, GlobalScript

to run doScript, store the script text in a form element named "script" to run a global script place the name of the global script in the "script" form element. For global scripts you can also specify the:
<input type="text" name="in" value="30">
<input type="text" name="intime" value="minutes">

a form that turns on a unit named "outside light"

<form action="[thislink]" method="post>
<input type="hidden" name="unit" value="outside light">
<input type="hidden" name="action" value="on">
<input type="submit" value="turn on the light">

</form>

replace the action form element with one that passes "toggle" to create a button that toggles the light on and off.

You could do the same thing with a single link like:

Toggle the outside lights

It is important when doing this to properly escape the name of the unit to be sent across a link like that. for most uses it will be enough to replace any spaces in the name with a "+" for other non text characters you'll need to escape them too. You can look up more information on this in any book on CGI scripts.

if you replace the text with the tag from above that shows the proper icon for the unit you create a clicable image just like in an xtension view.

The program doesn't care how the information gets sent to it, you can replace the hidden "unit" form element with a popup list, or a selection box or whatever you like. A popup list could even be generated by an applescript as shown in the last section of this readme.

NEW in 1.0 is the ability to include the 'in' and 'for' form variables that will create events in extension instead of doing things immediately. You can also include a 'intime' and 'fortime' value set to either "seconds", "minutes", or "hours" to tell it how to calculate the timing. If you don't include this for either value then it will assume you mean seconds. To turn on the outside lights in 30 minutes for 6 hours you would use a form like:

```
<form action="[thislink]" method="post">
<input type="hidden" name="action" value="turnon">
<input type="text" name="unit" value="outside lights">
<input type="text" name="in" value="30">
<input type="text" name="intime" value="minutes">
<input type="text" name="for" value="6">
<input type="text" name="fortime" value="hours">
<input type="submit" value="do it">
</form>
```

again, it doesn't care where the data comes from as long as the name/value information is there. So you can use popups for the action, or the name or whatever you like. See the "advanced_dim.x10" demo window for a nice example of how to do this.

**SECURITY

Since the pages all can control any object for public pages a way to control what unit it will control is necessary. Otherwise anyone with a little info could fill in any unit name and action and control anything in your house. (well, they would have to know what the units name was first) Currently I'm handling security with <XSecurity> tags. And then you should use regular web page access controls to add passwords to pages that you don't want other people having access to.

The security tag looks like this:

<XSecurity unit="outside light" allow="on:off:dim:toggle:xpress">

You can place these tags anywhere in the page and they will be removed before sending the page to the browser. If you include no security tags the page will allow control of all units. If you include any security tags the page will only control those specific units and allow only those specific actions. You could allow dim, but not on or off, or just on if you wish.

**ADDING APPLESCRIPTS

to do some of the more complex things it is necessary to add some logic. You can embed applescripts into webpages and include the return value in the html sent from the page. You do this via the <AppleScript> tags like:

```
<AppleScript>
  tell application "Weatherman"
     set currenttemp to current outside temperature
  end tell

     if currenttemp < 70 then
       return "<font color=blue>" & currenttemp & "</font>"
     else
```

```
      return "<font color=red>" & currenttemp & "</font>"
    end if
</AppleScript>
```

This is a fictional example, but you can see how useful this might be! The script object will be replaced by the returned value. There is a bug currently where the program will fail if you don't return anything, so remember to at least do a return "" at the end even if you don't need it.

There are a few applescript commands that I've added that you might find useful.

ReplacePageTag("my name", "James")

this will replace any occurrance of <my name> or [my name] in the document with "James" this is useful if you have many places where you want a result to show up. Or if you want to control the background color or text color in several different places in the document with a single script.

If you want to build regular links you can use the URLEncode() command like:

text = URLEncode("this is the link")

will return "this+is+the+link". I talked about this above in the controlling section.

You can also get access to any of the form elements from the last that was submitted to get this page.

value = GetFormVariable("unit")

would return to you the name of the last unit that was changed by the page.

You can have as many applescripts in a page as you want, but the more you have the longer it will take to return the page. The scripts are compiled only the first time the page is asked for, after that they remain in the cache and further hits on the same page will be faster.

Check my website for interesting things to do with the apple script tag and send me links to public sites or snippets of code to do more fun stuff and I"ll

post it on my website.

**ADDING AND USING PLUGINS
New in 1.0 is the ability to wrap your applescripts into plugins! This makes them easier to use across different pages and easier to share with other people. If you have good applescripts and want to make them into a plugin please send them to me I will help! Check out the xtest.x10 page for how to call a plugin and look at the Example Plugin file in the plugins folder in the x2web data folder to see how they are put together. I have plans for LOTS of great plugins.

**BUGS AND KNOWN ISSUES
If you use an applescript it must return something or the program may quit. If your script doesn't return any data just add a return "" to the end of the script.

Some applescript errors seem to be able to quit the program. This should only be a problem while you're building your pages, once they are working it's no longer a problem.
The XPress tag may not work yet.
targeting a copy of XTension on a remote machine doesn't work for everyone. I can use it here. If you are able to use it please let me know.

**Latest Version and other info

Latest version always at:
http://www.sentman.com/software.htm

this software is released as shareware in the traditional sense. You may use it for free. If you get a lot of value out of this you might consider purchasing a copy of one of my other shareware products. Many of these are very inexpensive and could help with your HA setup. This is not necessary, you may use this for free, but the more money I don't have to make working during the day, the more energy I have to work on these things at night;)

Thanks,
James james@sentman.com