

WEBWORDS

INTRODUCTION

The purpose of WebWords is to allow words or small passages to be sent to particular web sites for lookups, searches, etc. This could be both a convenience for the user and an opportunity to make money by selling sites the privilege of performing the function for the user.

Functions range from the obvious, like web searches or dictionary lookups, to the less obvious, like searching for an item on eBay or an online shopping service.

WEBWORD DESCRIPTORS






We do not want to hardcode WebWords into the client. Instead, we wish to come up with a framework for describing them that will allow flexibility in what WebWords are available and how they are used. Descriptors may be built into the client or downloaded from the PlayList server or perhaps installed in other ways. The elements of WebWord descriptors would be as follows:

- | | |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| Name | This is the name we'd use in menus and the like. "Search Web", "Define", "Search Amazon.com", etc. |
| Src | This is a URL to an image to be used for buttons. |
| Help | This is a short description of what the WebWord does; "Search the Internet for the selected terms, using Northern Light." |
| Command | This is the command to add to the jump URL so that the jump redirector knows where to send the word. |
| Enabled | If this is set to 0, the WebWord will not be shown to the user. We might use this to turn off a particular WebWord that is built into the client. |
| Toolbar | If this is set to 1, the WebWord will be added to the user's toolbar (the user may subsequently delete it, and that will be respected). |
| Id | This is a unique ID for the WebWord, so that it can be referred to by the playlist server (for replacement, for example). |

WEBWORD BEHAVIORS

- ☒ Any WebWord that is installed but not Enabled does not appear in the user interface. This fact will not be mentioned again; it is to be understood that all references to WebWords in the user interface mean "Enabled WebWords".
- ☒ The name of each WebWord appears in the Edit:WebWords hierarchical submenu.
- ☒ The name of each WebWord appears in context menus made on selections that are of moderate¹ size.

¹The exact meaning of "Moderate" will be left up to the Democratic Party.

-  WebWords should be available to be placed by the user on the toolbar.
-  WebWords placed on the toolbar by the playlist server must be removable by the user.
-  WebWords will always appear in the menus; the user cannot remove them.
-  WebWords should automatically appear on the toolbar if the playlist server requests it.
-  When a user chooses a WebWord, the WebWord should be sent to the adserver via the user's browser.² This will be discussed in depth later.

EXECUTING A WEBWORD

When the user chooses a WebWord, the client should:

- Form a jump URL using the “webword” action and the parameters described in the URL Chart.
- The WebWord's “command” attribute should be added to the URL as the value of the “command” query argument.
- The selection should be added to the URL as the value of the “selection” query argument. Don't forget to encode the selection properly.
- If there is no selection, or possibly at other times, we will bring up a dialog and allow the user to type in the word(s) they want to WebWord for.

WEBWORDS VS SPONSORED TOOLBAR LINKS

WebWords overlap Sponsored Toolbar Links in interesting ways.

- Both may show up on the toolbar.
- Both have names and icons.
- Both can be served from the Playlist server.

However, there are some differences, too.

- WebWords may be installed in the client from the start, and NOT served from the client.
- WebWords may appear in Light or Paid modes.
- WebWords send the user's selection to the adserver. Some users may view this as extremely sensitive information, we need to be *very* clear that this is going to happen.
- WebWords appear in menus, STL's (currently) do not.

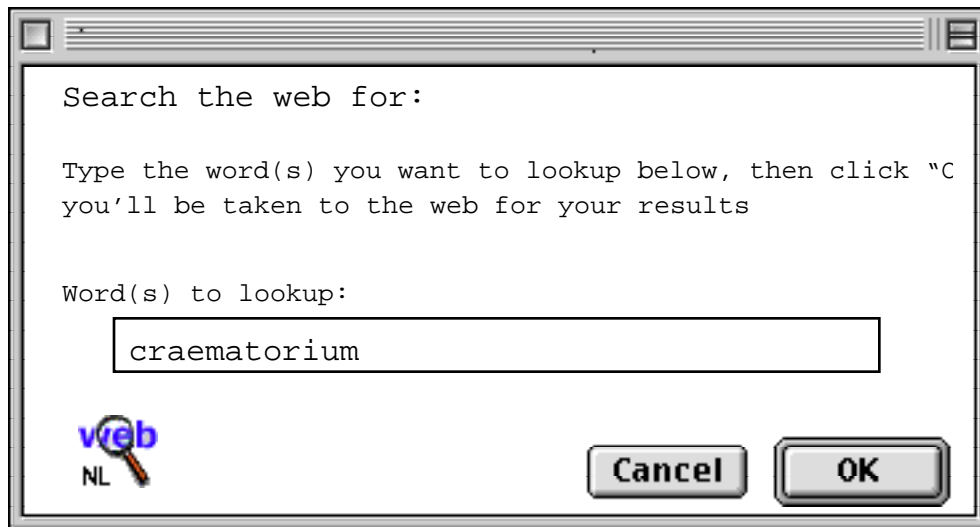
We may wish to consider merging the concepts. Perhaps STL's could get (for an extra price) WebWord behavior. Perhaps WebWords served from the Playlist server could be STL's with special attributes. This requires further thought.


CONTEXT-FREE WEBWORDS

WebWords may not always come from context; the user may wish instead to type them in. A

²It's possible that we may someday allow certain WebWords to return their data directly to the client; “Translate to French” or something. Or, they may return a list of replacement words (think Thesaurus). Later.

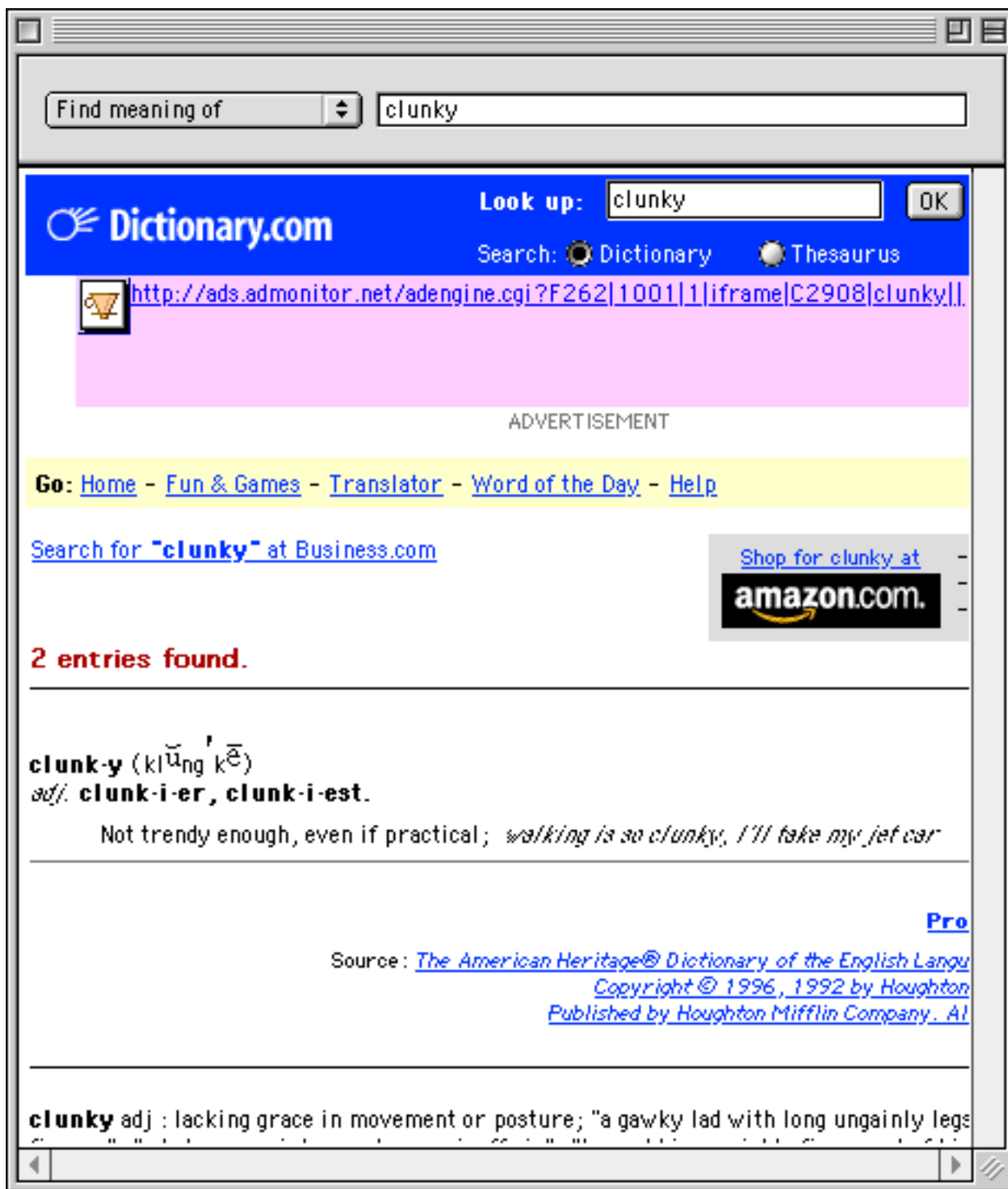
very simply solution is to have the user click a specific toolbar button or choose a specific menu item, and bring up a dialog before launching the URL:




This is easy to implement, and initiating the action is not modal; it avoids Cmd-F-word-return--cmd-period Syndrome. It's also true, however, that it does use an alert, and some people find this to be clunky³.

Another possibility would be a WebWord Wazoo, that would display results as well as let you type in a string:

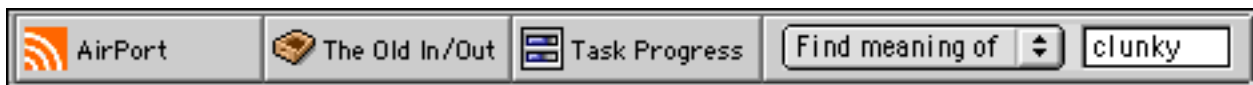
³clunky, n. 1. Clumsy, in the sense of "not the way I want to use it, to hell if you do". 2. Simple and elegant, but not fashionable. As in "using the phone to call you is clunky; let's use HearMe instead."



This has the advantage of displaying the results in Eudora, but it does indeed suffer from Cmd-F-word-return--cmd-period Syndrome; when the wazoo is invoked, the user will not be able to "predict" which action it will take. It would also require opening a window or switching wazoos to invoke a WebWord.

Another possibility is to use just the input part of the WebWord Wazoo, and stash it someplace appropriately small and ubiquitous, such as the toolbar (or even the ad window):

QUALCOMM Confidential. Dispose in Mountains of Ad Revenue



The results would then be displayed in a browser window or perhaps in Eudora itself.

It's possible we could mix and match qualities from these various approaches.

WEBWORDS & PAID MODE

One sticky issue with WebWords is that they really require us to have partners to handle them. There's very little point in having "Lookup in Dictionary..." as a WebWord if we don't have a site to which to send the queries.

For WebWords served from the Playlist server, this is not an issue; if we don't have a partner, we don't serve the WebWord. For Paid mode, however, we have a problem; Paid mode does not communicate with our ad server, and so we have no ability to add or remove WebWords from the Paid mode client.

We could simply declare WebWords to be a feature of Sponsored mode only. However, with the new thinking that Paid mode is actually of value to us, we don't want the Paid mode client to be less capable.

Possible solutions include:

- Having the Paid mode client make Playlist requests occasionally to get the proper WebWord list. Most users would be happy with this, a few would want to turn it off, and the lunatic fringe would consider us the AntiChrist for even trying it.
- Have the WebWords come in META tags in the update nag. This would require some machinery we don't have, but moreover would mean that the update nag actually Makes Things Happen, instead of just informing the user. Shrug.
- Have some sort of "installer" that the user would be notified of via the update nag, and could choose to download and run or not.

None of these alternatives make me very happy. Here is one that is worse:

- Have a fixed table for Paid mode. If we get new ones, too bad, the Paid mode user doesn't get them. If some in the table go away, the user chooses them and gets an error.

APPENDIX A - POSSIBLE WEBWORDS

Some possibilities for webwords are listed below. These are suggestive, not exhaustive:

- Dictionary Lookup meaning of word.
- Thesaurus Find synonyms of word.
- WebSearch Search web for word.
- PeopleSearch Search public directory for person.
- Buy Search for a product by title (amazon.com, etc)
- Map Get a map to an address.