W3A/V – WWW Applets/Viewers API

May 30, 1995

W3A-V: Viewers

A viewer is used to add viewing capability for a new format. It is an interactive application that normally displays its output in a window provided by the browser. It is characterized by:

1. one or more MIME types.
2. a list of object files.
3. a suffix.

All viewers must export the following functions:

- `Bool initXXX(char ***mime_types, int *nrtypes, float **qual)`

  (`XXX` is replaced with the unique suffix.) Initializes private variables and returns the list of formats that the applet accepts. Each string in `mime_types` must be a valid MIME type, possibly including wildcards `*`, `?` and `[]` (just like Unix file name patterns). The `qual` array must be exactly as long as the `mime_types` array and contain numbers between 0.0 and 1.0. A return of `FALSE` indicates an error, in which case *errno* holds the error code (see...).

- `Bool openXXX(const W3ADocumentInfo info, W3AWindow window, long id)`

  Directs the viewer to start a new window to display a document of the indicated type. The passed `window` is a drawing area in which the viewer can draw or place sub-windows. In a browser that works with the X Toolkit, the `W3AWindow` type is just an alias for `Widget`, on other window systems it will be something else. How to negotiate window positions and sizes is platform dependent (see also `W3Atoplevel`). The return code is `FALSE` in case of an **error**.

  The ID is a unique number, generated by the browser. The routines below use the same ID to refer to the same viewer instance.

- `int writeXXX(long id, const char *buf, size_t nchars)`

  Sends characters to the viewer. After the last call, a call with `nchars = 0` must be made, to indicate to the viewer that the end of the data is reached. The viewer may display the data immediately, or it may wait for the end of the data. The return value is equal to *`nchars`* or -1 in case of an error. (See **error codes**).

- `Bool closeXXX(long id)`

  Requests the viewer to destroy the window(s) corresponding to *`id`* and invalidate the *`id`*.

- `void eventXXX(long id, long sourceid, long eventtype, void *params)`

  This function is called when an abstract event occurred in the browser or another applet. *id* is the accessory's own ID, *source˙id* is the ID of the applet in which the event originated, *eventtype* is the type of event, *params* depends on the event type. (See the explanation of `W3Aevent`.)

- `Bool infoXXX(long id, W3ADocumentInfo *info)`

  This function may be called only after all data has been written to the viewer (the last call to `writeXXX` had `nbytes = 0`). Calling `infoXXX` gives the viewer the opportunity to modify the `title` field of `info`. Other fields may also be modified, but there is usually little sense in doing so. Most viewer will return without modifying `info`.

  Like with all changes to fields of a `W3ADocumentInfo`, the previous contents of the title fields must be freed and the new contents must be allocated on the heap.

> **Warning:** *Note that the restriction that* `infoXXX` *may only be called after all the data has been written to the viewer is different from the restriction on* `infoXXX` *in agents, where the function may be called immediately after the* `openXXX`.
