Reorganization of PPS attributes for the QNX CAR WebKit Browser#

For reference, the previous PPS attributes for the browser are documented here: http://graphics.ott.qnx.com/wiki/index.php/Kaleidoscope_Browser#Flash_HMI_API

The two main changes in the new browser PPS object handling are:

1. There is a single PPS "control" object that is used for global (i.e. window independent) attributes. Then there is a PPS object for each browser window.
2. Each PPS attribute name has a single letter prefix which puts it into one of 4 categories: Initial, State, Command, Response

When webkit_kd starts, it will use the QNX_COMMAND_OBJ environment variable to determine the location in the filesystem of its PPS control object. For example, the QNX CAR webkit startup script currently sets this environment variable to /fs/pps/qnxcar/external/webkit. In the reorganized system the PPS object named "webkit" would be the control object, and the containing directory "external" would be used for the per-window pps objects. To reduce confusion with other external apps, in the future we may wish to change the location of the PPS control object to something like: /fs/pps/qnxcar/external/webkit/control

The following attributes will be used in the control object:#

```
i_defaultClass
s_window*
c_quit
c_ping
r_pingResponse
```

The following attributes will be used in the per-window objects:#

```
i_class
i_id
i_posSize
i_visibility
i_uri
```

c_loadUri
c_go
c_zoomIn
c_zoomOut
c_zoomPercent
c_ping
c_modalDialog

s_posSize
s_visibility
s_requestedUri
s_uri
s_loadStatus
s_currentZoom

r_title
r_info
r_virtualKeyboard
r_pingResponse
r_dialogResponse
}

Details about each attribute follow:#

`i_defaultClass`
params: <class_name>
category: Initial
object: control
previous attribute: n/a
The default window manager class to use for any newly created windows.

`s_window<suffix>`
params: <object_name>
category: State
object: control
previous attribute: create_window, param 1
The name of the pps object associated with a WebKit window. This can either be a full path name (starting with '/') or be relative to the directory containing the control object. Note that there may be multiple attributes that start with 's_window' - each refers to a different window. There is an issue of how to come up with unique window names. I recommend to simply base the attribute name suffix on the pps object name, which already needs to be unique. **Important:** To avoid race conditions, you should wait until webkit_kd has performed the window's creation before deleting the window's PPS object. For example, if you are using the i_uri attribute, you can wait for the creation of the s_loadStatus attribute.

`c_quit`
params: none
category: Command
object: control
previous attribute: command::quit of last WebKit window
Clean up and exit the webkit_kd process.

`i_class`
params: <class_name>
category: Initial
object: per-window
previous attribute: create_window, param 6
The window class associated with the window. If not specified (normal case), the default class from the control object would be used. If neither attribute is set, WebKit would not set the class property of the window (leaving this up to OpenKode).

`i_id`

params: <id_string>
category: Initial
object: per-window
previous attribute: create_window, param 6
The window id used to set the io-winmgr id string property. If not specified, WebKit would not set this property (leaving this up to OpenKode).

`i_posSize`
params: <x> <y> <width> <height>
category: Initial
object: per-window
previous attribute: create_window, params 2-5
Initial size and position of the browser window. If not specified, information from the class in winmgr.conf will be used by Composition Manager, or Composition Manager will just use full screen. To leave just the position or just the size unspecified, use a value of "-1 -1".

`i_visibility`
params: <0|1>
category: Initial
object: per-window
previous attribute: n/a
Initial visibility of the window. If not specified, information from the class in winmgr.conf is used (by cm).

`i_uri`
params: <uri>
category: Initial
object: per-window
previous attribute: n/a
Initial uri to load in the window. Will not be used if s_uri is present and we are restoring a previous state.

`c_loadUri`
params: <uri>
category: Command
object: per-window
previous attribute: open_path
This command tells the browser to load the given URI. For the final URI of the page after it has finished loading, see the s_uri attribute.

`c_go`
params: reload|back|forward|stop|focusNextTypein|focusPrevTypein|close
category: Command
object: per-window
previous attribute: command

A series of simple browser commands that need no parameter. This attribute will only be used once on a delta basis, i.e. it is not a state attribute.

- reload: Reload the current web page. For the currently loaded URI see the s_uri attribute.
- back: Load the previous page in WebKit's page history.
- forward: Load the next page in WebKit's page history.
- stop: Stop loading the current web page.
- focusNextTypein: Change focus to the next field on the web page that accepts keyboard text input.
- focusPrevTypein: Change focus to the previous field on the web page that accepts keyboard text input.
- close: Close the browser window. Can also be achieved by deleting the PPS object for the window, or removing the window's s_window* attribute from the control object. Note that when using the first two methods of window closing, webkit_kd will automatically remove the s_window* control attribute for you. **Important:** to avoid a race condition you **must** wait until the s_window* control attribute has been removed before recreating the same window again.

`c_zoomIn`

params: [<number>]
category: Command
object: per-window
previous attribute: zoom_in
Increase the zoom factor for HTML layout by the given percentage. The default is 25%. The actual resulting zoom percentage will be written into the s_zoomPercent attribute by WebKit.

`c_zoomOut`

params: [<number>]
category: Command
object: per-window
previous attribute: zoom_out
Decrease the zoom factor for HTML layout by the given percentage. The default is 25%. The actual resulting zoom percentage will be written into the s_zoomPercent attribute by WebKit.

`c_zoomPercent`

params: <number>
category: Command
object: per-window
previous attribute: zoom_percent
Command to set the zoom factor of the browser window. The normal zoom is 100.

`c_ping`

params: none
category: Command
object: per-window
previous attribute: ping
Check if WebKit is processing attribute changes for the PPS object. This can be used on the control object or the per-window PPS objects. See also the r_pingResponse attribute.

c_modalDialog
params: <type_number> <dialog_text>
category: Command
object: per-window
previous attribute: n/a
This is currently the only command attribute created by WebKit for the HMI. It gives the HMI information on creating alert, confirm and prompt dialogs. The type numbers for these are 0, 1 and 2 respectively.

s_posSize
params: <x> <y> <width> <height>
category: State
object: per-window
previous attribute: scrn_prop
The current position and size of the browser window (i.e. destination viewport). WebKit will update this attribute as well as respond to changes.

s_visibility
params: <0|1>
category: State
object: per-window
previous attribute: scrn_enable
The current visibility of the browser window. WebKit will update this attribute as well as respond to changes.

s_requestedUri
params: <uri>
category: State
object: per-window
previous attribute: n/a
The URI that the browser was asked to load. For the final URI of the page after it has successfully finished loading, see the s_uri attribute. WebKit will update this attribute whenever it starts to load the main frame of a web page. For example, this attribute will contain the URI that WebKit was asked to load whenever the user selects a link, presses the "back" button, etc.

s_uri
params: <uri>
category: State
object: per-window
previous attribute: url
The uri of the last web page to be loaded (and should still be displayed) in the browser window. WebKit will update this attribute after a page loads. WebKit may use this attribute to restore a previous state. WebKit will not respond to changes in this attribute - see the c_loadUri attribute.

s_loadStatus

params: started|progress|finished|error|finished_with_error|cancelled <percent>|<error_msg>
category: State
object: per-window
previous attributes: start, progress, complete, info 0
This attribute is used for a series of responses to the HMI to tell it about the status of web page loading. The 3 possible values are:

- started: A web page or one of its subframes has started loading. The HMI should start the spinny thingy (progress indicator) and set its progress percentage to 0.
- progress: Page (or subframe) loading is still in progress. A number indicating the % completion is given.
- finished: The web page or one of its subframes has finished loading. Note that it may have been stopped by the user, by a page load error, or by the pending start loading of another page, etc, so this value does not indicate a complete or successful load.
- error: The web page loading encountered an error. The WebKit error message is given.
- finished_with_error: This value will be set after the "finished" value gets set if there was already an error on the page. This value is only used to indicate the current status more accurately. The HMI can ignore this value and just listen for "finished" and "error" separately. The WebKit error message is given.
- cancelled: This value will be set after the "finished" value gets set if the page load was cancelled (under the covers this is error -999). This value is only used to indicate the current status more accurately. The HMI can ignore this value and just listen for "finished". This state will occur when the user does a "stop" while a page is loading. This state will also occur (but only briefly) when the user does a "back", "forward", navigates to a new page, etc. before the current page has finished loading.

`s_currentZoom`
params: <number>
category: State
object: per-window
previous attribute: n/a
The current zoom factor of the browser window in percent. The normal zoom is 100.

`r_title`
params: <title_string>
category: Response
object: per-window
previous attribute: title
This is a response to the HMI containing the title of the web page currently loading or just loaded.

`r_info`
params: <info_type> <info_message>
category: Response
object: per-window
previous attribute: info 1,2
This is a response to the HMI containing status messages and hover-over-link text. The type numbers for these are 1 and 2 respectively.

r_virtualKeyboard
params: <action_code>
category: Response
object: per-window
previous attribute: virtual_keyboard
This is a response to the HMI that the virtual keyboard should be opened or closed. It also contains information on whether this is a password field being opened . The codes for these 3 actions are 1, 0 and 2 respectively.

r_pingResponse
params: [modal]
category: Response
object: per-window
previous attribute: ping_response
This is a response to the HMI that a c_ping attribute change was received. If webkit event loop processing is in a modal state, the word "modal" will be set as the attribute's value. Otherwise the value will be blank.

r_dialogResponse
params: <result_number> <prompt_response>
category: Response
object: per-window
previous attribute: n/a
This is a response from the HMI to WebKit that the user has completed a modal dialog such as the javascript alert, confirm or prompt. It will contain information on whether the user pressed OK or CANCEL (result numbers 1 and 0 respectively), and the prompt string they entered (if any).