Domain Scripting Quick Start Guide

This document provides an overview of the web scriptability functionality added to JAWS 17.

Script Naming and Loading

Scripts are loaded when a .jss file exists with a name that matches the domain name and extension plus the .jss file extension.

Example: *www.FreedomScientific.com.jss*

In order to create a user friendly name for a web script file, add an entry to confignames.ini.

Example: *www.FreedomScientific.com=FSWebPage*

Settings Center and other managers have been updated to allow domain settings to be loaded. This includes the Dictionary Manager which may now have domain specific entries. For example, a medical web portal may have many medical terms only relevant to that portal. These may now be added to a domain specific dictionary without cluttering your application or default dictionary.

Web Scriptability Functions

Added and enhanced functions make web scripting more powerful.

GetDomainName

GetDomainName returns the domain name or file base name for a locally stored web page.

GetActiveConfiguration

GetActiveConfiguration has been extended via an optional parameter to be able to obtain either the application script base name, the domain script base name (or null if no domain), or either the domain or application name as appropriate.

GetDocumentXML

GetDocumentXML has been added to obtain a well-formed XML document representing the current virtual document. Note this XML reflects but is not identical to the underlying HTML of the document. For example, it contains control type and state information and other data assigned by our DOM server. This XML may be processed by instantiating an MSXML DOM document object via the JAWS scripting language, and then by calling methods on this object to process and extract data from the XML. The below two functions are used to find and interact with elements from the XML from this function.

PerformActionOnElementWithTagAndAttribute

PerformActionOnElementWithTagAndAttribute has been added to find an element based upon tag and attribute data and perform an action on it such as setFocus, makeVisible, leftMouseClick, rightMouseClick, doDefaultAction.

> Note: *Tag does not refer to the XML tag, but the FSTag attribute of an XML node which reflects the underlying HTML tag. The attribute may be any other attribute from the underlying HTML which is automatically added to the XML node such as id, class, value, etc.*

PerformActionOnElementWithID

PerformActionOnElementWithID has been added to find an element based upon its FSDom assigned unique ID and perform an action on it such as setFocus, makeVisible, leftMouseClick, rightMouseClick, doDefaultAction.

> Note: *The FSDom assigned unique ID is not the id attribute which may be assigned in the underlying HTML, but a unique identifier assigned to every element by our FSDom server and which is reflected in the fsID attribute of an XML node.*

Use of Existing Script Functions

You do not necessarily need to use the new functions or process the XML; dozens of existing script functions may also be used in a domain specific script file such as MoveToControlType, MoveToControlledTarget, MoveToDocumentOffset, MoveToElementOfDifferentType, MoveToElementOfSameType, MoveToHeading, GetCurrentHeading, MoveToFormField, MoveToTable, MoveToTableCell, GetRowText, GetColumnText, GetColumnHeader, GetRowHeader, MoveToLandmark, MoveToHTMLFrame, MoveToRegion, saveCurrentLocation, SelectFromSavedLocationToCurrent, ClearSavedLocation, etc.

Web Scriptability Usage

The following are a few examples of how web scriptability can be used:

- Adding functionality to an online spreadsheet to calculate and read row or column totals, or monitor cells without moving from the focused cell
- Quickly invoking commonly used links on a webpage such as the “Next” or “Prior” links in a Google or other search results screen
- Rapidly gathering and speaking pertinent information from a dynamically updated web portal
- Setting focus to an important control via a keystroke in a web portal without having to navigate or tab to find the control
- Present a list of the most important actionable controls in a web portal to the user to make their work more efficient, etc.

Web Scriptability Examples

A sample script set has been added to the JAWS installation which demonstrates how to use web scriptability functions to build a toolbar script for a web application. This sample set is called SharepointWeb. To see this in action, modify the confignames.ini file to map your SharePoint web URL to the SharepointWeb script set. This sample obtains XML from the active SharePoint portal, instantiates the MSXML document parser, builds a collection of toolbar controls from the SharePoint ribbon, and presents these controls in a list to the JAWS user. The JAWS user may invoke this toolbar script with **CTRL+INSERT+F8**, select a toolbar control, and invoke its action.

Smart Navigation is a new mode in JAWS to make web application navigation more intuitive by presenting the web application in a similar way to a regular Windows application. Two examples follow:

- When focused on a toolbar of buttons, the buttons are laid out on one line in the virtual buffer and **LEFT** and **RIGHT ARROW** will navigate from button to button rather than character to character. One can drill down and read the buttons by character by first pressing sayCharacter twice and then using **LEFT** and **RIGHT ARROW**. Navigation returns to control navigation when the user arrows up or down away from the toolbar.
- When in a table, the entire row is shown in Braille and arrowing left and right or up and down navigates by cell and reads the new cell content. One can read the entire row by pressing sayLine twice, and then proceed to read by row using **UP** and **DOWN ARROW** until navigation reaches the table boundary at which point it returns to control or cell navigation.