
***** IMPORTANT NOTES ABOUT INSTALLING PATCHES *****

- In most cases, you need only replace the applicable DLL file, unless otherwise noted in the release notes. By overwriting any other files, you run the risk of overwriting any custom settings, especially Node Wizard Scripts.
- All patches are cumulative, but you may want to read through the notes about older patches to see if any other additional steps were required (other than replacing the DLL file).
- If you have any questions or concerns about an upgrade, please contact us. We don't want you to lose any data!

FRAMESLT 3.44 PATCH RELEASE NOTES – July 01, 2024

This release adds official builds for FM 2022 (v17), including support for FM 2022 update 4.

FRAMESLT 3.43 PATCH RELEASE NOTES – June 01, 2021

This release fixes a bug where the trial version could not initialize on certain versions of Windows. The fix is applied to the FrameMaker 2019 and 2020 version only.

FRAMESLT 3.41 PATCH RELEASE NOTES – January 22, 2021

This release includes:

- Official FrameMaker 2020 support.

FRAMESLT 3.40 PATCH RELEASE NOTES – January 24, 2019

This release includes:

- Official 64-bit FrameMaker 2019 support.
- Minor bug fixes

FRAMESLT 3.30 PATCH RELEASE NOTES – November 30, 2015

This release includes:

- Official FrameMaker 2015 support.
- A new Node Wizard Script option to treat whitespace as an attribute value delimiter (tokenized value recognition).
- New script attribute actions to sort values, such as numerically or alphabetically, in ascending or descending order.
- A fix to a bug where script autorunning did not work correctly upon document opening.

FRAMESLT 3.20 PATCH RELEASE NOTES – November 20, 2014

This release includes:

- In a Node Wizard Script, a new option to revert to the “classic” processing model when running on a book. In v3.00, with the introduction of book-based XPath axes, the processing model for books was changed to require XPath as the sole means of navigating book chapters. Previously, each file in the book was processed independently by the script, running all top-level XPath expressions at the chapter level. With this release, you can now add a <BookNavigationMode> element under a script <GeneralSettings>, in order to specify the use of classic mode processing. When specified and run on a book, the script will operate identically to the pre-v3.00 model.
- A bug where the plugin failed to read the proper XPath favorites setting file has been fixed.

FRAMESLT 3.20 PATCH RELEASE NOTES – May 12, 2014

This release includes:

- Official FM12 support.
- A fix to a Node Wizard bug where “Perform Actions and Match Next” did not work correctly with the attribute “Search and Replace String” action.

FRAMESLT 3.17 PATCH RELEASE NOTES – August 23, 2013

This release includes the following enhancements and fixes:

- A new way to enable licensing. Previously, a separate DLL file was required to enable the licensed version. Now, a simple license key in the preferences file enables the licensed version. If you are licensed and you need a key, please contact West Street.

- A new setting in the preferences file that allows the FrameSLT menu to appear in a parent WestStreet menu, or not at all (API calls enabled only). By default, the FrameSLT menu will still appear in the main book and document menus.

- A new setting in the preferences file that controls the maximum allowable length of an XPath expression. The valid range is 255 to 20,000 characters, with a default of 2,000 if the setting is not present.

- A new setting in the preferences file that controls the maximum number of characters in an element node that will be tested for an XPath expression with a textual node test. For example, consider the following expression:

```
//p[contains(., "some text")]
```

If this setting is set to 255 characters, FrameSLT will only search within the first 255 characters of each <p> element. This setting is provided to improve performance if the element axis also matches elements with large amounts of text, such as section-level elements. The valid range is 255 to 20,000 characters, with a default of 255 if the setting is not present.

- Node Wizard Script dialog box improvements – This dialog box now allows you to select any scripts document, whereas previously it was locked to the single scripts document in your settings area. It also allows you to more easily manage which file a script runs on when invoked. Along with these improvements, a new API call was added to set the current scripts document location, functioning similar to a selection within the new dialog box.

- Workspace compatibility – FrameSLT dialog boxes may now be incorporated with workspaces. With this enhancement, any dialog boxes that are open when you close FrameMaker will reappear when you restart it.

- Within a Node Wizard Script, you now have the option to specify an element tag for insertion, wrapping, etc. with a parameter value or an XPath query, like string values in other areas of a script. To enable this capability, you need to import the EDD from the scripts document in the installation package into your current scripts document (or just use the installation package file). The plugin is compatible with previous versions of the scripts document EDD, so previous scripts should still function as before without an EDD update.

- New API calls to retrieve the plugin major and minor version numbers.

FRAMESLT 3.12 PATCH RELEASE NOTES – October 1, 2012

This patch:

- Provides official FrameMaker 11 support. Earlier versions are mostly functional on FM11 but some anomalies are present.

- Fixes a bug that prevented the match history “browse” button in the Node Wizard from functioning.

- A new option for the FindNextNode API call to return an element unique ID (FP_Unique property), rather than an FDK F_ObjHandleT object ID. An FP_Unique ID is much easier to handle when FindNextNode is called from ExtendScript. See the *User Guide* for more information.

FRAMESLT 3.11 PATCH RELEASE NOTES - April 20, 2012

This patch fixes a problem with truncation when reading XPath expressions from Node Wizard Script elements. The official character limit was increased to 2,000 in v3.10, but some 255-character truncation routines still operated when reading expressions from scripts, effectively overriding the new increased character limit.

FRAMESLT 3.10 RELEASE NOTES – April 3, 2012

FrameSLT 3.10, originally planned as a limited bug fix/minor enhancement release, eventually grew to a significant advancement of the product. The following items briefly describe its contents... see the *User Guide* for more information.

*** IMPORTANT NOTES FOR EXISTING USERS: ***

This release is not considered a patch. If you are currently using a previous 3.x version of FrameSLT, you should note the following:

- When installing, you should overwrite all current files with files from the ZIP. **However, you may consider making a backup of your scripts file first.** See below.
- Upon initialization, this version will make a backup of your existing Node Wizard Scripts file and report the location. If you have custom scripts, you'll need to copy them back into the new scripts file afterwards. The EDD and template of the scripts file has changed... so you may need to make some minor adjustments to your scripts. Naturally, we apologize for any inconvenience... any changes should be minor. **You should consider making a backup of your current scripts file yourself before installing this new version just to be sure that your custom scripts do not get overwritten.**
- Your local preferences file will be overwritten. After the first startup, you should consider reviewing those settings (FrameSLT > Local Preferences) to be sure they reflect your desired settings.
- The "strings" language file has changed. If you are using customized language files, you should review the current default English strings file for any changes, then update your customized file. **The use of an outdated file could cause FrameSLT to crash FrameMaker.** In the installation package, any changes from the previous version are indicated in red.

INCREASED XPATH EXPRESSION LENGTH

The Node Wizard and Node Wizard Scripts now support XPath expressions up to 2000 characters, versus the previous limit of 255 characters.

FLOW-BASED XPATH AXES

To direct a query towards a specific flow, the XPath syntax now supports four new flow-related axes:

- flow-any:: - Query any flow
- flow-body – Query body page flow(s)
- flow-master – Query master page flow (s)
- flow-ref – Query reference page flow(s)

For example, the following expression will match all <p> elements in a flow named “MyFlow” on the body pages:

```
flow-body::MyFlow//p
```

The following expression will match the first text frame(s) of all flows in the document named “MyFlow”:

```
flow-any::MyFlow
```

With the ability to match a flow, two existing element actions have been extended to work on flows:

- Wrap_contents_in – Wraps all contents of the flow in the specified element. This works on currently structured and unstructured flows
- Delete_contents – Deletes the entire contents of the flow.

PAGE NUMBER FEATURES FOR XPATH AND SCRIPTS

An XPath expression can now retrieve and evaluate the page number of an element (or text frame using flow axes), with additions to the existing “fmprop::” axis property set. For example, the following expression will match all <p> elements on any page after page 3, where the first page of the document is always page 1:

```
//p[fmprop::PageNumInt > 3]
```

A second property, PageNumStr, allows you to retrieve the actual page number assigned by the document numbering properties. As an example, the following expression will match all <p> elements on page “iv”, assuming the use of Roman numerals for page numbering:

```
//p[fmprop::PageNumStr = “iv”]
```

Additionally, you can retrieve a page number within a Node Wizard Script with new GetPageNumInt and GetPageNumStr actions under Get_object_property. Combining these actions with new XPath capabilities, FrameSLT can now perform intricate operations between flows with processing based on page numbering.

GENERAL SCRIPTING ENHANCEMENTS

Node Wizard Scripts now:

- Provide a “Disable” attribute on event elements and action containers, which if set to “Yes”, effectively “comments out” anything beneath it.
- Report the elapsed time following the completion of a script.

NEW SCRIPT ACTIONS

Node Wizard Scripts now support a variety of new actions:

- String operations, such as adding and truncating strings, locating and replacing substrings, and length evaluation
- Basic numeric (integer) operations, including addition, subtraction, multiplication, and division
- Set_elem_text and Set_attr_value - Actions to directly set these items from a static string, parameter, or XPath query. Previously, you could set element text and attribute values from the clipboard only, which required the interim step of setting the clipboard first.
- Copy_elem_text_to_parameter and Copy_first_value_to_parameter – Streamlined actions for setting a parameter directly from element text or an attribute value. This capability existed previously, but may have required additional interim steps.
- Under the “Do_messaging” group, the following new interactive prompt actions, all of which set a specified parameter with the respective result:
 - Do_string_entry_prompt – Requests a free-form string
 - Do_path_browser_prompt – Allows the selection of a folder in the file system
 - Do_file_browser_prompt – Allows the selection of a file in the file system
- Under OtherActions, new Disable_screen_updates and Enable_screen_updates actions which turn screen updates on and off. With screen updates disabled, many scripts will run considerably faster.
- Under OtherActions, a new Exit_script action, which safely terminates the script immediately.
- Under OtherActions, new Close_file and Save_file actions. BE VERY CAREFUL WITH THESE! CHANGES CANNOT BE UNDONE!

ENHANCED SCRIPT ACTIONS

- Get_object_property actions now support “Copy_to_clipboard” and “Apply_to_parameter” options. Previously, these actions always copied the retrieved value to the clipboard, which is still the default if neither option is specified. The ability to apply the retrieved value directly to a parameter is an important step for streamlining and expediting certain scripts.
- Existing Do_messaging prompts (Do_OK_Cancel_prompt and Do_Yes_No_prompt) now allow you to capture the button click by setting a parameter, where OK/Yes = 1 and Cancel/No = 0. Previously, OK/Yes simply meant “continue processing” and Cancel/No always meant “terminate script.” Now, if a parameter is specified, the script always continues processing after setting that parameter. If no parameter is specified, previous behavior is unchanged.
- The existing Do_messaging prompts: Do_OK_Cancel_prompt and Do_Yes_No_prompt now support minor variations: Do_Cancel_OK_prompt and Do_No_Yes_prompt, where the only difference is the button that has the default focus, Cancel and No, respectively.

NEW SCRIPT PROLOG SECTION

The scripts file now includes an optional “prolog” section where you may place any miscellaneous text for any general purpose.

NEW API-BASED EXTERNAL CALLS

- SetScriptParm – Allows you to set a parameter for use by a Node Wizard Script, before running the script.
- GetScriptParmByte – Allows you to iteratively retrieve the value of a parameter that was set by a script, byte-by-byte.
- ClearScriptParms – Clears all defined parameters in memory.

NEW EXTENDSCRIPT “EXTERNAL OBJECT” METHODS

Using a new capability known as “external objects” (EOs), FrameSLT can now be invoked as an EO within an ExtendScript script. Preliminary method support includes SetScriptParm and GetScriptParm which set and retrieve parameter values, respectively.

FRAMESLT 3.01 PATCH RELEASE NOTES – January 4, 2011

IMPORTANT! If you are using version 3.00, you should install this patch, as the memory leak issue was potentially serious.

This patch addresses the following issues.

- **Memory leaks during Node Wizard and Node Wizard Script processing.** A serious memory leak issue associated with these features has been fixed.
- **Node Wizard Scripts and the new fmbook:: axis.** In some cases, a script may have reported that an expression beginning with this axis lacked the proper context for evaluation. This axis never requires a starting context (other than some active file). The problem has been fixed.
- **New Select_match action for Node Wizard Scripts.** A new action was added that allows elements to become selected as they are matched. This action may be useful for debugging and/or messaging purposes. To use this action, you must import the EDD from the scripts document in the installation package into your scripts document, or simply overwrite your scripts document. **If you overwrite, you will lose any custom scripts that you built.**

FRAMESLT 3.00 RELEASE NOTES – July 26, 2011

Thank you for checking out FrameSLT version 3! If you are already familiar with earlier versions, here's a summary what's new. **IMPORTANT - IF YOU ARE A CURRENT USER OF VERSION 2.x, YOU REALLY SHOULD READ THIS.** There have been some fundamental changes to how the plugin works. We expect that you may encounter some issues while migrating and we ask you to please feel free to contact us for help.

FULL UNICODE (UTF-8) SUPPORT

All FrameSLT features, including XPath parsing and navigating, are believed to fully support the characters within the Unicode range.

EXPANDED XPATH SUPPORT – FILE-RELATED AXES

FrameSLT XPath now provides four new special axes that allow you to step from a document structure tree to a book, from a book to a chapter document, and directly from any file to another. The cumulative effect is that you can step from any file to another using XPath logic, which significantly enhances the power and flexibility of Node Wizard Scripts. For example, now you can start in any file, walk to another and copy data, then step back to the original file and insert that data anywhere. More on this later.

Here is a quick example... the following expression will match all <p> elements in an entire book, regardless of whether the book or a chapter document is currently active:

```
fmbook::*//*/fmchap::*//p
```

The updated User Guide has more explanation and several examples. Please read this information carefully, as these are non-standard axes that were designed specifically for FrameSLT. They have very specific and specialized functionality that may not be intuitive from a cursory glance.

Note: This enhancement inherently includes the ability to navigate a book structure tree, which never existed before.

EXPANDED XPATH SUPPORT – OBJECT PROPERTIES

XPath can now query on new object properties such as cross-reference properties (target file, idref, format), marker properties (text, format), and table properties (format). For example, you could match all <xref> elements whose underlying cross-references target "somefile.fm" with the following expression:

```
//<xref>[contains(fmprop::XRefSrcFile, "somefile.fm")]
```

See your User Guide for more information and examples.

FUNDAMENTAL CHANGE WITH NODE WIZARD (AND SCRIPTS) PROCESSING MODEL

With the introduction of book-oriented XPath axes and the ability to navigate book structure trees, the original processing model of the Node Wizard and Node Wizard Scripts with respect to books has become obsolete. Previously, when you ran either on a book, the Node Wizard setup or the script iterated through chapters and ran independently on each one. The Node Wizard included a "Book" or "Document" processing option that added further control over this, operating similar to the comparable option in the FrameMaker Find dialog box.

Now, the "chapter iteration" model is removed. All navigation through chapters (and any other file) must be controlled directly with XPath. For example, if you are in the Node Wizard and you want to match every <p> element in some book, you must use the proper axes to direct that query, such as:

```
fmbook::*//*/fmchap::*//p
```

Previously, you could have simply made the book active and specified:

```
//p
```


...after which the Node Wizard would iterate through each chapter automatically. THIS IS NO LONGER TRUE. If you did the same thing with FrameSLT 3.0, it would look for <p> elements in the book structure tree only. So, in summary:

- The XPath specified in the Node Wizard or a script will operate directly on the respective tree of the active file, whether it is a book or a document. If you have a book active and you want to step into document structure trees, you must use the new axes to do it.
- If you have previous scripts that ran on books based on the old model, THEY PROBABLY WILL NOT WORK ANYMORE without some adjustment to the XPath expressions. We apologize for this inconvenience and will be happy to assist with any migration efforts. We truly believe that this is a necessary step, as the use of XPath for inter-file navigation is exceedingly precise and must replace the original model.

ENHANCED NODE WIZARD SCRIPT ACTIONS

A host of new actions have been added to Node Wizard scripts, mostly related to the manipulation of non-markup items, such as the configuration of cross-references and markers. With the addition of file-related XPath axes combined with the existing action set, the scripts have evolved into a very powerful content manipulation tool with something of an XSLT flavor. The following is a brief summary:

- Use of scripts with file-related axes – Because you can now walk seamlessly between different files with XPath, you can use the traditional clipboard features to move content from any file to another, including between elements, attributes, and more. You can effectively build any kind of new document based on content anywhere else.
- Unlimited actions per script event – Previously, you were limited to a single element and/or attribute action per event. Now, there is no limit. Any kind of action can be configured in any kind of order, noting that you should take care not to destroy the original context of the XPath match during the process, otherwise subsequent actions could fail.
- Cross-reference configuration – A script can now insert and precisely configure a cross-reference to virtually any FrameMaker document, anywhere. With this functionality, you can create custom TOCs, indexes, etc.
- Marker configuration – A script can now insert and configure marker text.
- Referenced graphic configuration – A script can now configure the path of a referenced graphic.
- Table manipulation – A script can now do a variety of table manipulation tasks, such as adding/deleting rows and columns, setting column widths, and setting the format.
- Interactive prompts – You can now add interactive prompts that have the ability to display any kind of text, with the optional capability of aborting a script in progress.
- Miscellaneous amenities – Scripts can now do general activities such as refreshing EDD definitions and updating cross-references.

PARAMETERS IN NODE WIZARD SCRIPTS

Node Wizard Scripts now support the concept of "parameters", with some similarity to the usage of parameters in XSLT stylesheets. You can set any number of parameters based on static text, clipboard content, XPath queries, etc., then use them:

- In XPath expressions
- To configure object properties
- To insert text into a document

For example, if you set the parameter "MyParm" to "p", you could subsequently use the following expression to match all <p> elements in a file:

```
// $MyParm
```

The introduction of parameters has significantly expanded the flexibility of scripting; however, their use can add a higher level of complexity. If you need assistance, please see the User Guide and/or contact us.

NODE WIZARD ENHANCEMENTS

The Node Wizard dialog box includes the following:

- A new counter in the Match History area that allows you to see where you are in the history and jump directly to any particular match.
- The ability to automatically load XPath favorites upon startup, based on a new preferences option.
- Better control of the attribute action area, including the automatic population of drop-down boxes when adding action attributes and values.
- A new option to allow the Node Wizard to open files, used in conjunction with the new file-related axes.

TRANSLATABLE INTERFACE

Most menu commands, dialog box controls, and messages are now customizable, with the primary intent to allow translation to another language. You will see a new Language menu that controls these features, including a sample "Bogusian" language that is intended to demonstrate the general framework. See your User Guide for more information. Note that FrameSLT employs a very large number of strings for dialog boxes and prompts... translation would be no trivial task. If you have interest in translation and especially if you would like to offer another language suitable for distribution with the product, please contact us.

TRANSFORMATION FEATURE DEPRECATION

The seldom-used transformation feature set is scheduled for deprecation with the next major release. All capabilities have been surpassed by the enhanced Node Wizard Script capabilities, using a much more elegant methodology. All transformation features are believed to be still functional; however, they were not tested and we do not intend to support them.

FRAMESLT 2.36 PATCH RELEASE NOTES – February 4, 2010

- Node Wizard and Node Wizard Script actions now support some degree of Unicode, especially as related to content movement and copy/paste actions. You should now be able to reliably copy and paste

characters in the unicode range between attributes and elements. Note that the XPath engine does not yet support Unicode for expressions.

- The Node Wizard and Node Wizard Scripts now support a new attribute action, "Set unique ID". This action will auto-detect the unique ID on the matched element and set it to an 8-character alphabetic ID. It will fail if the element has no unique ID attribute or its unique ID attribute is currently specified (that is, it will not overwrite existing IDs). If the XPath expression matches an attribute, that attribute will be ignored and the parent element will be used to auto-detect the proper attribute.

- Minor bugs with script-related copy/paste actions have been fixed, including 1) The failure of the clipboard to clear when attempting to copy whitespace, then pasting to an attribute (the most recent clipboard persisted and pasted instead) and 2) removal of a spurious message box that sometimes appeared during script actions with the text "Empty!", left by mistake during a debug session.

FRAMESLT 2.35 PATCH RELEASE NOTES – December 28, 2009

- A Node Wizard script will now accept the currently-selected element in the target document as a context node for the XPath expressions in the script. In other words, you can now run a script on a selected element, rather than the entire flow. With this, the previous restriction that all script XPath expressions had to begin with a forward slash (/) has been removed. In a script, all expressions that do not begin with a forward slash will use the currently-selected element (at the time of script launch) as its context. Naturally, any expressions that do begin with a forward slash do not need a context node and will ignore the selected element.

- The FrameSLT.dll and support files can now be installed anywhere on the local hard drive, provided that the path in maker.ini is correct. Support files (such as FrameSLT_Settings.fm) must be in the same folder as the DLL file. Existing installation instructions are still valid and have not changed.

- A crash associated with a "Hello" external call has been fixed.

FRAMESLT 2.32 PATCH RELEASE NOTES

This release fixes a bug that prevented a Node Wizard script from properly copying the content of text() nodes to the clipboard. It is also the build for the initial FM9 release.

While the core features of the plugin appear to function well with FM9, note that there are some anomalies with the appearance of dialog boxes and reports that did not occur in previous versions. We believe that there are bugs in version 9 of the FDK which prevent us from fixing them at this time. We will monitor the evolution of FrameMaker 9 and update our software as appropriate.

FRAMESLT 2.3 RELEASE NOTES

This is an important release with a number of improvements, including:

GENERAL

- Faster XPath parsing
- Addition of the "fmprop" axis to FrameSLT XPath
- New "Perform Action(s) and Match Next" button in Node Wizard

TRANSFORMATIONS

- Support for relative file paths for transformation elements
- Support for parameters, similar to XSLT (xsl:param)
- New "maximum matches" attribute for FSLT_for-each and FSLT_copy-of elements
- Optional "double-click editing" for transformation elements

EXTERNAL CALLS

- Support for setting parameters (FSLT_param) via external calls before transforming files

See your User Guide for more details.

FRAMESLT 2.21 PATCH RELEASE NOTES

Made an important fix to speed up complex Node Wizard scripts with subevents. The original design sometimes caused them to run very slow.

Fixed the last() function, which was accidentally broken by features added in 2.2

FRAMESLT 2.2 RELEASE NOTES

Version 2.2:

- Adds the very important Node Wizard subevent feature. See the User Guide for more information.
- Combines the Full and Lite versions into a single, full-featured product.
- Adds the "Match All" button to the Node Wizard.
- Adds a new preference option to handle the XPath position() function more closely to standard.

FRAMESLT 2.1 RELEASE NOTES

Version 2.1 provides some improvements on how FrameSLT stores its settings, especially with respect to the installation process. If you already have 2.0 working, there is no need to install 2.1. If you do install 2.1, it will reset any custom settings you had, if any.

Version 2.11 added the ability for the Node Wizard to "Perform Actions on All" with a particular node selected in the Structure Tree. Previously, you had to process a whole structure tree with that button. Now, you can select a single element and process that branch only. Any XPath that does not begin with a to-root axis (/) will start from the context of the selected element.

Version 2.12 fixed a crash that a user encountered when closing the FrameSLT error report. It was an obscure and rare problem.

FRAMESLT 2.0 RELEASE NOTES

Thank you for checking out FrameSLT version 2! If you are already familiar with earlier versions, here's a summary what's new. There have been many small things done, these are just the highlights. Information on point/patch releases is included afterwards. Note that all fixes/additions are cumulative.

EXPANDED XPATH SUPPORT

XPath support is now much better, noting the following improvements:

- Support for the contains(), starts-with(), and not() functions
- Full node set comparison abilities in predicates and function terms, consistent with the W3C
- Better support for W3C syntax conventions
- Better parsing and error reporting

NODE WIZARD IMPROVEMENTS

The Node Wizard now has:

- Lots of new and very versatile element and attribute actions
- The ability to perform attribute actions on the attributes matched by the XPath expression
- The ability to transfer text from attributes to elements, and vice-versa
- The ability to search/replace text within attribute values
- The ability to search and perform actions in any structured flow
- "History" buttons to peruse the history of recent matches
- Easier entry of attributes and values for attribute actions
- Context-sensitive tips for element and attribute actions
- The ability to warn you when a book query comes around full circle
- An improved "quick finder" that is actually worth using
- The ability to perform "invasive" actions such as unwrapping and deleting elements as a "Perform All" function

NODE WIZARD SCRIPTING

You can now script any action or set of actions supported by the Node Wizard for full automation. Scripts can be run at any time, or set to autorun during key events. This is a very powerful feature and we recommend you check out the new Node Wizard Scripts tutorial.

CONDITIONAL TEXT AUTO-ASSIGNMENT

Actually introduced in version 1.2, this feature allows you to associate condition tags with structural markup, like that EDD context rule you always wished you had. A good place to start with this is the new tutorial on Condition Management.

TRANSFORMATION BUG FIXES

- FSLT_create-xref now works right
- The FSLT_if and FSLT_when elements can now work according to W3C standards, if designated as such in the Preferences.

EXTERNAL CALLS

You can now launch a Node Wizard script from another API client or FrameScript.