

FrameMaker 10 and ExtendScript Automation

This eSeminar will provide an introduction to FrameMaker 10's new ExtendScript scripting model. We will introduce the scripting environment and demonstrate solutions to real-world production challenges.

Objectives

After completing this unit, you should be able to:

- Understand what ExtendScript is and how it works with FrameMaker.
- Identify the benefits of automation.
- Understand what is involved in writing scripts.
- Find scripting resources.

Topics

In this unit, you will learn about the following topics:

- What is ExtendScript in FrameMaker 10?
- ExtendScript joins other FM automation methods
- Why would you use scripting?
- How to get started with scripting
- Best and worse of FrameMaker ExtendScript
- Resources

What is ExtendScript in FrameMaker 10?

- ExtendScript is a scripting environment built into FrameMaker 10. It gives you the ability to automate FrameMaker tasks.
- The concept is similar to Visual Basic for Applications (VBA) that is built into Microsoft Office products like Word and Excel.
- ExtendScript is Adobe's version of the JavaScript scripting language. It is extended to include features that go beyond the core JavaScript language.
- ExtendScript is used for automation in other Adobe products, such as RoboHelp, InDesign, Photoshop, Illustrator, and Acrobat.

ExtendScript joins other FM automation methods

Frame Developers Kit (FDK)

- Uses C/C++ libraries and requires a compiler.
- Code is compiled down to a dll, which is then installed.

FrameScript

- Third-party scripting environment for FrameMaker.
- Like ExtendScript, FrameScript scripts are compiled at runtime.

Why would you use scripting?

Scripting is all about automation and making things go faster. There are three major reasons to use automation:

- Automate tedious, repetitive tasks.
- Extend FrameMaker functionality.
- Make impractical tasks practical.

We will demonstrate scripts that illustrate all three reasons for automation. All examples are based on production scripts designed for actual clients.

Automate tedious, repetitive tasks

- Remove duplicate entries in an index.
- Change the alignment of anchored frames.
- Highlight specific table cells, based on content.

Extend FrameMaker functionality

- Unlock text insets.
- Save FrameMaker 10 documents to FrameMaker 8.
- Notification scripts—scripts that run automatically in response to a FrameMaker event.

Make impractical tasks practical

- Add static images to document pages.
- Create custom PDF bookmarks.

How to get started with scripting

There are three basic requirements for writing scripts

- Understand the language syntax, in this case, ExtendScript.
- Understand the target application's "object model," in this case, FrameMaker.
- Have the ability to develop "algorithms" for solving the problem.

Understand JavaScript syntax

- Since ExtendScript is JavaScript, there are many resources for learning the language, such as books, web sites, and videos.
- Most of the resources will deal with JavaScript in web browsers and won't apply directly to FrameMaker.
- There is some documentation available from the ExtendScript Toolkit's Help menu, but none of it pertains directly to FrameMaker.
- The best way to learn is to use and adapt sample scripts.

Understand the FrameMaker object model

- This may be more important than knowing JavaScript syntax.
- You need to know the FrameMaker objects that can be manipulated, for example, documents, paragraphs, formats, etc.
- Objects will have properties that can be retrieved or changed. For example, a paragraph format has a Name property.
- There are “methods” that are used to manipulate FrameMaker objects.
- At this point, the only available documentation is the Object Model Viewer (Help > Object Model Viewer).
- Currently, the best approach for learning the FrameMaker object model is to download the free FDK (Frame Developers Kit) and use its documentation.

Learn to develop algorithms for solving problems

- Start at the smallest level (for example, a table cell), and develop code for that level.
- Expand the code up to each of the next levels, for example, all of the cells in a table, all of the tables in a document, all of the documents in a book.
- Use a series of small functions instead of developing a long linear script. This promotes easier testing, troubleshooting, and reuse of common functions.

Best and worse of FrameMaker ExtendScript

Best:

- Great dialog box model. “Panels” and form objects resize to accommodate text.
- Easy to localize dialog boxes, menus, etc.
- Built-in functions for manipulating XML and working with files and folders.
- Ability to do cross-application scripting.

Worse:

- Very little FrameMaker-specific documentation and examples.
- Bugs.
- Difficult to integrate with non-Adobe applications, such as databases, and Excel spreadsheets.

Resources

Books

- *JavaScript: The Definitive Guide* by David Flanagan. O'Reilly.
- *JavaScript: The Good Parts* by Douglas Crockford. O'Reilly.

Videos

- Douglas Crockford on JavaScript (<http://yuiblog.com/crockford/>).

Online forums and mailing lists

- <http://forums.adobe.com/community/frame-maker/>
- <http://tech.groups.yahoo.com/group/framescript-users/>
- <http://lists.frameusers.com/mailman/listinfo/framers>
- <http://www.omsys.com/cgi-bin/dada/mail.cgi/list/framers>

Script vendors

- Carmen Publishing Inc. (rick@frameexpert.com)