



About the CTXML Disk Image

Copyright © 2008 by Apple, Inc. All Rights Reserved.

The disk image CTXML contains files that illustrate various methods for processing the XML generated by Cinema Tools, version 4.0 and later. This disk image contains several example projects. If you wish to edit the example source code from the disk image, we recommend copying the projects elsewhere, such as your home folder.

You may incorporate sample code from these examples into your applications without restriction, although the sample code has been provided "as is" and the responsibility for its operation is completely yours. However, you should not redistribute the source as "Apple Sample Code" if you make changes to it. If you're going to re-distribute the source, we require that you make it clear in the source that, although the code derived from Apple Sample Code, you've since made changes to it.

Two approaches are included in this material:

- Two sample AppleScript projects with XSLT stylesheets that transform information in a Cinema Tools XML file into tab-delimited text files suitable for display in a spreadsheet application such as Numbers or Excel.
- Two sample Xcode projects that parse Cinema Tools XML into a tree, access certain elements, and then send output to the Console.

Additionally, the article [Transforming Cinema Tools XML](#) provides some background information about this Cinema Tools feature.

The Cinema Tools XML File

Cinema Tools, version 4.0 and later, can export information about film lists in two formats: XML and PDF. The XML file provides information in a format that other tools can read and process, offering the prospect of more closely integrating Cinema Tools information into the workflow. The PDF file provides information in a human-readable format.

The folder `CTXML` contains two files:

- `sequence1.xml` – an XML file generated by Cinema Tools. You can use this file as a source file for both the XSLT and Xcode examples.
- `sequence1.pdf` – a PDF file generated by Cinema Tools. It presents the same information as the XML file in human-readable format.

The XSLT AppleScript Examples

XSLT stands for “Extensible Stylesheet Language for Transformations” and is an official recommendation of the Worldwide Web Consortium (W3C). The folder `XSLT` contains the following files:

- `cutlist.xml` – a sample XSLT stylesheet that extracts cut list information from a Cinema Tools XML file and transforms this information into a tab-delimited text file.
- `XSLTproc.scpt` – an AppleScript script for exercising an XSLT stylesheet.
- `Pull List.app` – an AppleScript droplet application for transforming a Cinema Tools XML pull list into a tab-delimited text file.

Using the XSLT Stylesheet

Using a XSLT stylesheet requires a source XML file, the stylesheet file, and an XSLT processor. In Mac OS X, the processor is available from the command line and is called `xsltproc`. From the Terminal application, you can run `xsltproc`, specifying the output file path, the stylesheet to use, and the source XML, as well as several other options. (For details, type `man xsltproc` at the command line.)

Much easier than using the command line, however, is the the included AppleScript (`XSLTproc.app`) application to perform the XSLT processing for you.

1. Double click the XSLTproc AppleScript application to launch it.
2. Respond to the prompts for the locations of the source Cinema Tools XML file (`sequence1.xml`) and the stylesheet file (`cutlist.xml`). (If you have other XML files or other stylesheets you want to use, you can choose those files.)

The script runs the `xsltproc` command line tool with the parameters you’ve specified. The resulting tab-delimited text file is saved in the same directory as the source XML file and is opened in the TextEdit application.

You can open this text file in Numbers or Excel to see the information displayed in a worksheet.

You can use a text editing application to modify or extend the stylesheet `cutlist.xml` in order to transform other information in a Cinema Tools XML file.

Using the Pull List droplet application

The AppleScript droplet application `Pull List.app` is a fully functional example that exports a tab-delimited pull list from a Cinema Tools XML file. After exporting XML film lists from Cinema Tools, drop the XML file on the Pull List script application and choose a location to save the tab-delimited pull list text file. You could also double click the Pull List script application and respond to the prompt to choose a Cinema Tools XML file.

The Pull List script application has an XSLT stylesheet embedded within it. Control-click on the application in Finder to “Show Package Contents”. In the Resources folder in the Contents folder, you will find the stylesheet called `pulllist.xml`. You can edit this stylesheet with a text editor to extend or modify it. You can easily add, delete, or rearrange the columns of data that are exported in the tab-delimited text file by editing the embedded stylesheet.

The Pull List AppleScript application will accept multiple XML files dropped on it at one time and it will export a tab-delimited text file for each XML file.

Building the AppleScript applications

The scripts for both the XSLTproc and the Pull List applescript applications can be modified by opening those applications in the Script Editor, found under `/Applications/AppleScript/`. After editing the script for XSLTproc, save the script as the file format “Application” so that it will launch when double clicked. After editing the script for the Pull List application, save the script as the file format “Application Bundle”. Do not check any options check boxes.

After saving the application bundle for the Pull List script, you must open the package and copy the stylesheet “pulllist.xml” or whatever you have named it, into the Resources folder of the package.

The Xcode Projects

The best way to programmatically process the individual elements in a Cinema Tools XML file (or any XML file, for that matter) is to first parse the XML into a tree structure. Mac OS X ships with two different code libraries for doing this: the CoreFoundation framework and the Foundation framework.

The folder `xcode` contains two Xcode projects that use these libraries:

- CF XML Tool uses the CoreFoundation library and is written in C/C++. The main code for the project is in the file `main.cpp`.
- NS XML Tool uses the Foundation library and is written in Objective-C. The main code for the project is in the file `Tool.m`

Each project builds a Console application (`cfxmltool` or `nsxmltool`, respectively) that you can execute from the command line. The application opens a specified XML file, parses the XML into a tree, accesses certain elements of the tree, and then prints information about these elements to the Console.

Building and Running the Xcode Projects

To build and run the Xcode projects, you need to have Apple's Developer Tools installed on your system. Also make sure that `MacOSX10.4u.sdk` is installed in `/Developer/SDKs/`. Then follow these steps:

1. Copy the Xcode projects from the disk image to a writable volume so that you have space to build the projects.
2. Double-click one of these sample project files to open it in Xcode.
(`CF XML Tool.xcodeproj` OR `NS XML Tool.xcodeproj`)
3. Click the `Build` icon to build the executable. The resulting executable is stored in `[project folder]/Build/Release`.
4. Open the Terminal application.
5. Drag the executable (`cfxmltool` or `nsxmltool`) to the Terminal window. (You can drag the file either from the Executables group in the Xcode window or from the Finder.) This creates a path to the file.

6. Drag the Cinema Tools XML source file (`sequence1.xml`) to the Terminal window.
7. Make sure the Terminal window is active and then press Return to run the program. The program output appears in the Terminal window. (You can write the Console output to a file using the redirect operator on the command line.)

The sample code in these Xcode projects is a starting point. Using one of these sample projects, you should be able to extend the code to access any or all of the elements in a Cinema Tools XML film list.