Adobe® XMP Toolkit SDK Overview

This overview introduces version 4.4 of the Toolkit SDK for the Extensible Metadata Platform (XMP).

Version 4.4 of the Adobe XMP Toolkit SDK provides documentation and libraries for working with the XMP data model; for reading, writing and manipulating XMP metadata in various file formats. Refer to the *XMP Specification* (included in this SDK) for detailed descriptions of the XMP data model, file type support, and schemas supported in Adobe products. The XMP Toolkit SDK is published under a BSD License.

SDK components

The XMP Toolkit SDK contains two libraries, XMPCore and XMPFiles.

- **XMPCore**

  This library supplies an API for parsing, manipulating, and serializing metadata, according to the XMP data model and regardless of the file format of the data it describes. The XMPCore API is provided by the classes `XMPMeta`, `XMPIterator`, and `XMPUtils`; a complete API Reference is available in both HTML and Javadoc.

  XMPCore is provided as a C++ implementation with project files for:

  - Windows Vista / XP (including 64 bit support) using Visual Studio 2005 (VC++ 8)
  - Mac OS X 10.4 / 10.5 using Xcode 2.4 or higher or Xcode 3, creating universal binaries for PPC and Intel processors.
  - A makefile for building XMPCore (not XMPFiles) library and samples is provided for Linux and GCC 4.*x*.

  A Java implementation of XMPCore is also provided, to be used with J2SE Version 1.4.2 or higher. Project files for Eclipse 3.2 and an Ant build file are included.

- **XMPFiles**

  This library supplies an API for locating, adding, or updating the XMP metadata in a file. The API allows you to retrieve the entire XMP Packet, which you can then pass to the XMPCore component in order to manipulate the individual XMP properties.

  XMPFiles contains a number of “smart” file handlers that know how to efficiently access the XMP in specific file formats. See the *XMP Specification Part 3, Storage in Files* for details of how XMP is embedded in the supported file formats. The SDK also includes a packet scanner that can be used for unknown file formats. Scanning is much less efficient than using the smart handlers, and should be used only if necessary.

XMPFiles is provided as a C++ implementation with project files for:

- Windows Vista / XP (including 64 bit support) using Visual Studio 2005 (VC++ 8)
- Mac OS X 10.4/10.5 using Xcode 2.4 or higher, Xcode 3

Dependencies

These publicly available components are needed to build the C/C++ libraries:

- The Expat XML parser is needed for XMPCore on all platforms.
- The zlib compression library is needed for XMPFiles on all platforms.
- Apple's QuickTime for Windows SDK is needed for XMPFiles on Windows.

See instructions on obtaining and installing these tools in the `ReadMe.txt` files in the placeholder folders for each tool, and in the *XMP Toolkit SDK Programmer's Guide*.

There are no dependencies for the Java version of XMPCore.

XMP Toolkit SDK changes

VERSION 4.4.2:

- Handlers for additional file formats, including ASF (WMA, WMV), FLV; MPEG4; SWF; folder-based video formats AVCHD, P2, SonyHDV, and XDCAM; UCF (see *XMP Specification Part 3, Storage in Files*).
- Additional schemas to support document histories, composed documents, and temporal metadata (see *XMP Specification Part 2, Standard Schemas*).
- XCode projects work in XCode 3
- VS8 projects for Windows now include 64-bit build targets for Windows.
- Expanded, updated, and reorganized documentation. The *XMP Toolkit SDK Programmer's Guide* has been renamed and updated for new features. The *XMP Specification* has been split into three parts; *Part 1, Data and Serialization Models Part 2, Standard Schemas*, and *Part 3, Storage in Files*.
- Additional and updated sample code. See "Sample code and tools" on page 3.

VERSION 4.1.1: Added the XMPFiles library and the Java version of XMPCore. Visual Studio 2005 (VC8) projects replaced Visual Studio .Net 2003 (VC7) projects. Code Warrior projects were removed.

VERSION 3.5: Added Xcode projects for building universal binaries in Mac OS. Added functions to the `SXMPUtils` class to support the latest Adobe DNG SDK.

VERSION 3.2: A complete rewrite of the XMPCore, for a more convenient API, and a smaller, faster, more robust implementation.

Downloading the XMP Toolkit SDK

The single ZIP file download can be used in Mac OS®, Windows®, and UNIX®/Linux®.

Note that source and text files have UNIX-style line endings using ASCII linefeed. The Visual Studio project files have Windows-style carriage-return/linefeed line endings.

XMP Toolkit SDK contents

The downloadable ZIP file contains the following folders under the root folder `XMP-Toolkit-SDK-4.4.0`:

`/`	At the root level, the license agreement (`BSD_License.txt`) and this overview (`XMP-Toolkit-SDK-Overview.pdf`).
`build/`	Projects and makefiles for building the C/C++ version of the XMP Toolkit SDK in Mac OS, Windows, and (for XMPCore) UNIX/Linux.
`docs/`	The three-part *XMP Specification*, the *XMP Toolkit SDK Programmer's Guide*, and the API reference documentation (`API/index.html`).
`java/`	The Java implementation of XMPCore, with the Javadoc documentation, project files for Eclipse 3.x, and sample code. The `readme.txt` file describes how to set up projects in Eclipse.
`public/include/`	The header files and glue code that clients of the XMP Toolkit SDK must include.
`samples/`	Sample and tutorial source code and build projects, with the necessary resources to run the sample code. See "Sample code and tools" below.
`source/`	The source code that implements the XMP Toolkit SDK libraries.

`third-party/`
`expat/`
`QTDevWin/`
`zlib/`
`MD5/`
	Place holders for third party source files which are needed for the XMP Toolkit SDK, including `ReadMe.txt` files with information on how to obtain and install the tools. MD5 source code, needed by both components for MD5 hash computation, is included.

Sample code and tools

The SDK provides a set of samples that illustrate coding techniques for various tasks. In addition to the source code for each sample, there is a project file for use with a platform-specific IDE.

- ➤ Project files for MS Visual Studio 2005 are in the folder `<`*`xmpsdk`*`>\samples\build\vsnet8`.
- ➤ Project files for XCode 2.4.1 are in the folder `<`*`xmpsdk`*`>/samples/build/xcode2`
- ➤ Build files for GCC are in `<`*`xmpsdk`*`>/samples/build/gcc3.2`

The source code for the samples is in `<`*`xmpsdk`*`>/samples/source`. When you build them, the compiled code is written to `<`*`xmpsdk`*`>/samples/target/`, to a platform-specific folder with `debug` and `release` subfolders.

These command-line sample applications are provided:

`ReadingXMP` *filename*	Demonstrates the basic use of the XMPFiles and XMPCore components, obtaining read-only XMP from a file and examining it through the XMP object.
`ModifyingXMP` *filename*	Demonstrates how to open a file for update, and modifying the contained XMP before writing it back to the file.
`CustomSchema`	Demonstrates how to work with a custom schema that has complex properties. It shows how to access and modify properties with complex paths using the path composition utilities from the XMP API.
`XMPCoreCoverage` `XMPFilesCoverage`	These demonstrate syntax and usage by exercising most of the API functions of each XMP Toolkit SDK component, using a sample XMP Packet that contains all of the different property and attribute types.
`XMPIterations`	Demonstrates how to use the iteration utility in the XMPCore component to walk through property trees.
`DumpMainXMP` *filename*	Uses the XMPFiles component API to find the main XMP Packet for a data file, serialize the XMP, and display it.
`DumpScannedXMP` *filename*	Scans a data file to find all embedded XMP Packets, without using the XMPFiles API or smart handlers. If a packet is found, serializes the XMP and displays it.

In addition, these command-line development tools are provided:

`dumpfile`	Recursively parses the structure of the given file and prints a view of the file structure to standard output. This tool is not intended to extract metadata from files, or for any kind of use in production. It is a development aid that can help you determine whether a file that cannot be read by XMPFiles is malformed.
`xmpcommand`	Performs basic XMP actions such as get, put, and dump. Can be used for testing and scripting automation.

For additional information about how to use the samples and tools, and for tutorial walkthroughs of the basic samples, see the *XMP Toolkit SDK Programmer's Guide*.