

Reference: 2007/10/31-8.2

Outside In XML Export Technology SDK Quick Start Guide

This document provides an overview of the Outside InXML Export Software Developers Kit (SDK). It includes download instructions, installation overview, architecture description and other topics that will help readers to get started working with the SDK . Pointers are given throughout to the Developer Guide and sample applications. Readers should also use the other documents available in the Outside In section of Oracle Technology Network.

Contents

Product Overview	3
Target Audience	3
Use Cases	3
Available Downloads	4
Installation	4
Directory Structure	5
Architecture	5
Integration	6
Information on common Issues	7

Product Overview

Outside In XML Export converts and normalizes the content of supported file types into XML defined by Oracle's FlexionDoc schema.

This schema uses XML tags designed to closely mirror the information in files created in popular business applications. In addition to the document content itself, FlexionDoc includes information related to document properties, structure, and formatting. The resultant XML can be consumed by an application or further transformed by the developer via Extensible Style Language Transformations into a developer-defined schema.

Outside In XML Export is appropriate for use in any application that can benefit from normalizing documents into XML, which explicitly describes all the elements of the document's content, structure, properties, and formatting.

XML Export is delivered as a Software Development Kit (SDK) with C based Application Programming Interfaces (APIs). It is available for a number of operating systems, listed below.

Target Audience

XML Export is for Software Developers who wish to extract detailed document content, properties and structure from documents processed by their applications.

Use Cases

XML Export is used in applications that will benefit from XML based conversion of files including:

- data forensics
- search & indexing
- data analysis
- document repurposing

Available Downloads

The XML Export SDKs for each supported platform are contained in archive files that can be downloaded from Oracle Technology Network. The link to download these file is available on the same page from where you downloaded this document, or from the link below:

http://www.oracle.com/technology/products/content-management/oit/oit_dl_otn.html

Each of the following downloads include all the files needed to evaluate/implement the XML Export for that platform.

Linux (x86-32)

Solaris (Sun SPARC-32)

Windows (x86-32)

Installation

To install the demo version of the SDK, copy the contents of the archive (available on the Web site) to a local directory of your choice.

For Windows versions, unzip the archive to the directory of your choice.

For UNIX versions of the SDK, copy the tgz file corresponding to your platform (to a local directory of your choice). Decompress the tgz file and then extract from the resulting tar file as follows:

```
gunzip tgzfile
```

```
tar xvf tarfile
```

Directory Structure

The installation directory contains the following directory structure:

<code>*</code>	Contains a working copy of the technology and compiled executables of the sample applications.
<code>\common</code>	Contains the C include files needed to build or rebuild the technology.
<code>\docs</code>	Includes HTML and PDF versions of the SDK Developer Guide.
<code>\lib (Windows Only)</code>	Contains the library (.lib) files for sccex.dll , sccda.dll and sccfi.dll .
<code>\resource</code>	Contains localization resource files.
<code>\samples</code>	Contains a number of subdirectories, each one holding the source code for a different sample application.
<code>\xxfiles</code>	Contains sample files designed to exercise the technology.

Architecture

The basic architecture of the XML Export is the same across all supported platforms. The input filters form the base of the architecture. Each one reads a specific file format or set of related formats and sends the data to the normalization and caching module through a standard set of function calls. There are more than 150 of these filters that read more than 400 distinct file formats. Filters are loaded on demand by the data access module.

The normalization and caching module is responsible for caching a certain amount of data from the filter and returning this data to the export filter

The Export Filters are architecturally similar to input filters, and write out a specific format based on information coming from the normalization and caching module. The export filters generates XML in this case.

The export module implements the export API and loads and runs individual export filters.

The Data Access module implements a generic API for access to files. It understands how to identify and load the correct filter for all the supported file formats. The module delivers to the developer a generic handle to the requested file, which can then be used to run XML Export.

XML Export ships with the Flexiondoc schema. Schemas can be presented in the form of a DTD (Document Type Definition) or XML Schema (schema). The Flexiondoc schema is provided in both forms.

Integration

The best way to begin working with XML Export is to examine the documentation and sample applications. From there you can begin to plan the integration of this technology into your own application.

The XML Export Developer Guide

The Developer Guide for XML Export provides more detailed information about getting started, including

- Implementation on Windows and Unix
- Data Access and Export Functions
- Messages
- Using Redirected I/O
- Options
- Sample Applications

The Developer Guide also contains a list of the filter libraries and the supported formats and platforms for the specific Outside In SDK.

XML Export Sample applications

XML Export includes the executables and source code for a number of sample applications. Use the sample application executables to see examples of some of the features of the SDK. The source code for each sample application will illustrate how that functionality is implemented.

XML Export includes the following sample applications.

export (Windows Only)

This sample application allows the user to run a single source file through XML Export. The user can choose the source file, an output file and set the various XML Export options.

exsimple

This simple command line driven program allows the user to run a single source file through XML Export. The user can choose the source file, an output file and set the various XML Export options.

xxredir

This sample application is based on the **exsimple** sample application. It is designed to demonstrate how to use redirected IO and callbacks in XML Export.

xxsample

This sample application is an extremely basic implementation of XML Export that uses all of the default settings for every option.

ExJava / exporter

The **ExJava** Java wrapper, working in tandem with the **exporter** sample application, provides a working example of one method of interfacing with Oracle's C-based SDK products from a Java application. **Export.jar** is a Java API wrapper used by a Java application to control the **exporter** executable and set conversion options. **exporter** is a C-based executable which performs conversions using the modules in the Outside InExport SDK

Information on common Issues

The following sections of the Developer Guides for each SDK address issues that new users of the Outside In Technology often ask. You may want to read these sections in particular during your evaluation process.

The Basics

These sections of the Windows / UNIX Implementation Details chapters describe how to start calling the API, sending and receiving messages, and other functions specific to each SDK. Most of the topics covered in this section are illustrated by the sample application code.

Linux Compiling and Linking

This section of Unix Implementation Details chapter contains information about library compatibility concerns for your specific flavor of Linux

Runtime Considerations (UNIX)

This section of the Unix Implementation Details chapter contains information about running in a variety of UNIX environments. See especially the *X Server Requirement* and *System Fonts* sub-sections.

Running in a 24x7 environment

This section of the Implementation Issues chapter discusses process isolation when running Outside In Technology in 24 x 7 environments.