ORACLE

Reference: 2007/10/31-8.2

Outside In Content Access Technology SDK
Quick Start Guide

This document provides an overview of the Outside InContent Access Software Developerôs 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

Outside In Content Access extracts text and metadata from more than 400 file types and automatically translates the text and properties from multiple possible encodings into a single encoding specified by the developer. It is optimized for performance, interactively providing data to the host application in memory as the input file is processed. The technology is widely used in search and data forensic applications and is designed for high-throughput server environments

Content Access 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

Content Access is for Software Developers who wish to integrate the searching of the supported file types into their applications. It can also be used in conjunction with HTML Export or Image Export to provide annotation features in the output of those SDKs.

Use Cases

Content Access is used in applications in a number of markets including:

- indexing & search engines
- security & computer forensics products
- content repositories

Available Downloads

The Content Access SDKs for each supported platform are contained in archive files that can be downloaded from Oracle Technology Network. The link todownload 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 theContent Access for that platform.

FreeBSD (x86-32)

HP-UX (IA-64)

HP-UX 32-bit (PA-RISC)

IBM AIX 32-bit

Linux (IA-64)

Linux (IBM zSeries)

Linux (x86-32)

Linux (x86-64)

Solaris (Sun SPARC-32)

Solaris (x86-32)

Windows (IA-64)

Windows (x86-32)

Windows (x86-64)

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:

*****	Contains a working copy of the technology and compiled executables of the sample applications.
\common	Contains the C include files needed to build or rebuild the technology.
\docs	Includes HTML and PDF versions of the SDK Developer Guide.
\ExportMaps (Windows Only	Contains a number of sample Export Maps designed to produce attractive output
\files	Contains sample files designed to exercise the technology.
\lib (Windows Only)	Contains the library (.lib) files for **sccem.dll**, **sccem.dll**, **sccca.dll**, **sccta.dll**, **sccra.dll**, **sccex.dll** and **sccda.dll** and **sccfi.dll**.
\resource	Contains localization resource files.

`\samples`	Contains a number of subdirectories, each one holding the source code for a different sample application.

Architecture

The basic architecture of the Content Access 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 Content Access module.

The Content Access module reads data from the normalization and caching module and repackages it in a way that is convenient for the developer. This repackaging process includes mapping characters to a particular character set and converting some data (such as paragraph and cell breaks) into representative characters. There is also a Text Access module that is virtually identical to the Content Access module, though it does offer some subtle functional differences.

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 a generic handle to the requested file, which can then be used to run more specialized processes.

Data Access conserves resources by creating only one file handle and one normalization and caching handle for each file, even if it is opened in multiple Content Access instances.

Integration

The best way to begin working with Content Access 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 Content Access Developer Guide

The Developer Guide for Content Access provides more detailed information about getting started, including

- Implementation on Windows and Unix
- Data Access Common Functions
- Content Access and Text Access Functions
- 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.

Content Access Sample applications

Content Access 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.

Content Access includes the following sample applications.

casample

This sample application shows a typical usage of the Content Access API. Because this is intended as a simple template or reference for common Content Access usage, it creates only rudimentary output. However, it does initialize, exercise and cleanup Content Access output. Content Access requires the usage of the Outside In Data Access module. Therefore, this application also demonstrates usage of a portion of Data Access.

tademo (Windows Only)

This sample application provides a simple demonstration of text access. The text from a file is read a block at a time and displayed in the **tademo** window. The **`TAReadFirst`** and **`TAReadNext`** functions are directly tied to menu options, and the block size may be set by the user. An option is also provided to save the text to a file.

taredir (UNIX Only)

This sample application provides a means of using the API presented in this guide without the need for Motif libraries. All extracted text is output to the standard output device, or can be redirected to a file or another device.

textdemo (UNIX Only)

The sample code in the **textdemo** files shows how to use the API presented in this guide. This application is essentially identical to the Windows-only application **tademo**,.

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.