

Reference: 2009/02/09-8.3

# Outside In Viewer Technology SDK

## Quick Start Guide

This document provides an overview of the Outside In Viewer Technology Software Developer's 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

<b>PRODUCT OVERVIEW</b>	<b>3</b>
<b>TARGET AUDIENCE</b>	<b>3</b>
<b>USE CASES</b>	<b>3</b>
<b>AVAILABLE DOWNLOADS</b>	<b>3</b>
<b>INSTALLATION</b>	<b>4</b>
<b>DIRECTORY STRUCTURE</b>	<b>4</b>
<b>ARCHITECTURE</b>	<b>5</b>
<b>INTEGRATION</b>	<b>5</b>
<b>INFORMATION ON COMMON ISSUES</b>	<b>7</b>

## Product Overview

Outside In Viewer Technology (OIVT) renders high-fidelity representations of files without using the files' native applications. It enables client-side file viewing, printing, and copy/paste functionality of nearly 500 file formats. It includes an annotation API, which allows the developer to highlight, hide, or insert text in a document “view” without altering the underlying native file. Its use is appropriate for solutions that can benefit from desktop viewing functionality.

Outside In Viewer Technology 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. It is also available as an ActiveX control for Microsoft Visual Basic or other ActiveX-compliant development environments.

## Target Audience

OIVT is for Software Developers who wish to integrate file viewing, annotation and printing functionality into their applications.

## Use Cases

OIVT is used in applications that view, print, and annotate content including:

- rich document management clients
- rich e-mail clients
- multi-platform viewing environments
- computer forensics

## Available Downloads

The Viewer Technology 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](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 technology for that platform.

HP-UX RISC-32

HP-UX RISC-64 (with 32 bit binaries)

IBM AIX 32-bit (with iSeries)

Linux x86-32

Linux zSeries  
 Solaris (Sun SPARC-32)  
 Solaris (x86-32)  
 Windows (IA-64)  
 Windows (x86-32)  
 Windows (x86-64)  
 Windows ActiveX (x86-32)

## Installation

To install the evaluation version of the OIVT, 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>\files</code>	Contains sample files designed to exercise the technology.
<code>\lib (Windows Only)</code>	Contains the library (.lib) files for <b>sccca.dll</b> , <b>sccta.dll</b> , <b>sccra.dll</b> , <b>sccda.dll</b> and <b>sccfi.dll</b> .
<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.

## Architecture

The basic architecture of the Viewer Technology 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 nearly 500 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 display engine. It is also responsible for running the filter to rebuild any data not already in the cache that is requested by the display engine.

The Display Engine is responsible for reading data from the normalization and caching module and displaying it in the view window. It is also responsible for all user interface, clipboard and printing for a given data type. There are currently six display engines: Document, Spreadsheet/Database, Bitmap, Drawing, Archive and Hex. Display Engines are loaded on demand by the view window.

The View Window controls all of the modules for this product.

## Integration

The best way to begin working with OIVT 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 OIVT Developer Guide

The Developer's Guide for OIVT will provide more detailed information about getting started, including

- Implementation on Windows and Unix
- Using the View Window
- Messages
- Using Redirected I/O
- Options
- Sample Applications

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

## **Viewer Technology Sample Applications**

OIVT 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.

OIVT includes the following sample applications.

### **annoex (Windows Only)**

This sample demonstrates the view window's advanced annotation and positioning capabilities

### **annotate**

This sample is essentially the same as **SIMPLE** except that it uses the Outside In Raw Text and Annotation ability to annotate all occurrences of the text "the" in the document.

### **apitest (UNIX Only)**

This sample provides a comprehensive test of the API functionality.

### **drawpage**

This application demonstrates the Draw Page feature that can be used to display thumbnails and write pages to bitmaps.

### **mdiview (Windows Only)**

This application shows how multiple view windows can be supported in a multiple document interface setting.

### **mfcview (Windows Only)**

This application shows a simple viewer implementation using Microsoft Foundation Class.

### **options (Windows Only)**

This sample app demonstrates a broad set of messages.

### **print (Windows Only)**

This application shows various print options for Windows.

### **printd (UNIX Only)**

This application code demonstrates how to implement printing in the Outside In Viewer Technology using a dialog on UNIX platforms.

### **prints (UNIX Only)**

This application code demonstrates how to implement printing in the Outside In Viewer Technology for UNIX platforms

## **redirect**

This sample is essentially the same as **SIMPLE** except that it redirects the Outside In input routines (**Read, Seek, Tell, ...**) through its own routines.

## **search**

This sample is essentially the same as **SIMPLE** except that it adds the ability to search for strings in the file being viewed.

## **simple**

This sample demonstrates the simplest implementation of the Viewer Technology. It shows the basics of how to create a view window, view a file, print the viewed file and copy to the clipboard.

## **welcome (Windows Only)**

This application displays the **welcome.doc** file.

# **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.