OfficeAdapterDecoder Design Implications

NOTE before Proceeding:

As of 10.7 we introduced OfficeDecoder as a replacement for OfficeAdapterdecoder. Support STRONLGY recommends using it instead of OfficeAdapterDecoder.

However, he OfficeAdapterDecoder is still part of DotImage .. it is distinct from the newer OfficeDecoder.

OfficeDecoder: Present in 10.7 and newer

- Uses the Perceptive filter objects
- Does not require Office license
- MAY NOT render pages precisely as Office shows them
- Support STRONGLY recommends this be used
- Please see INFO: Changes Introduced in 10.7 for OfficeDecoder details

OfficeAdapterDecoder: Present in 10.6 and newer

- Requires MS office 2010 or later to be installed and licensed on the machine
- has multiple security issues (see below for more details)
- Does render pages more exactly as office (which is why we didn't remove it

completely)

WordDecoder: Present only in 10.5

Was removed from future versions as it was unsuitable to the task

Customers who bought licenses for Atalasoft Word Decoder were provided replacement Office Decoder licenses

Do not use

SUMMARY: Customers should use the OfficeDecoder.. it's faster and more secure.. the OfficeAdapterDecoder remains available. but we recommend against its use in the strongest of terms.

Design Implications

The OfficeAdapterDecoder is an implementation of our MultiFramedImageDecoder class for the rendering of Microsoft Office documents (Word, PowerPoint and Excel) with the goal of having output AtalaImage obects which very closely represent how they are displayed within their respective Office applications. COM automation is used internally to achieve this, which has a variety of drawbacks. This article aims to be a summary of the Microsoft Knowledge Base entry https://support.microsoft.com/en-us/kb/257757 "Considerations for server-side Automation of Office", outlining some of the more important points as well as demonstrating possible improvements that can be implemented for some of the performance issues.

Licensing

The OfficeAdapterDecoder requires a licensed installation of Microsoft Office to operate. Prior to using the OfficeAdapterDecoder in a production environment, it is advisable to contact a Microsoft representative to address Microsoft Office licensing concerns.

INFO: OfficeAdapterDecoder Design Implications

Security

In a server-side environment, macros can represent a security threat. The default setting Office uses to handle macros is "Disable all macros with notification" this is acceptable and will prevent the OfficeAdapterDecoder from running potentially dangerous macros. However, it is advisable to ascertain that this is the current setting before using the OfficeAdapterDecoder in a production environment. For a guide on how to ensure macros are disabled in each Office application, see:
https://support.office.com/en-us/article/Enable-or-disable-macros-in-Office-documents-7b4fdd2e-174f-47e

Performance

The process used by the OfficeAdapterDecoder to create AtalaImages is as follows:

1. An instance of the required Office application is started.
2. The document stream is written to a temporary file.
3. The Office application opens this temporary file.
4. The Office application exports this document to PDF.
5. The PDF is then rasterized to AtalaImage objects using our PdfDecoder class.
6. The Office application is closed.

Below are listed some of the tools provided in the SDK to help boost the performance of this process.

The OfficeSession class

This class encapsulates the lifetime of the Office application instances in an implementation of IDisposable. When the OfficeAdapterDecoder reads images it can use the applications loaded by the OfficeSession. The Close method should be called to ensure the Office applications are closed when no longer needed.

This eliminates steps 1 and 6 in the process outlined above from having to be performed for each document / page. Instead, the Office applications can remain open and running in the background until they are no longer needed.

The ICache interface

An instance of this interface can be provided to the OfficeAdapterDecoder and will control the caching of the intermediary PDFs which Office generates. This can substantially improve performance, eliminating steps 2 through 4 in the OfficeAdapterDecoder process after the document has been exported the first time. The SDK provides 2 basic implementations of ICache: DisposableValueCache and LruDisposableValueCache.

DisposableValueCache is the basic implementation of ICache for values which implement IDisposable and need those values disposed when removed from the cache. This cache must be emptied manually when appropriate, by calling the Clear method.

LruDisposableValueCache subclasses DisposableValueCache and features automated removal of elements when a certain capacity is met, prioritizing the removal of the least-recently-used items. Clear will not need to be manually called when using this cache implementation.

The OfficeAdapterDecoder can accept an ICache which associates the stream containing the original document with a DisposableTempFile (Which represents the temporary PDF which Office produced).

Multiple OfficeSessions for multiple cores

In some highly-threaded environments (Such as in the requests handled by the WebDocumentRequestHandler class) the .NET 4 “BlockingCollection” class can be used to allow multiple office sessions to be active concurrently (Which the OS will then distribute across the available resources).

Attached is a sample project featuring is a simple implementation of the WebDocumentRequestHandler which includes caching and makes use of a BlockingCollection of OfficeSessions initialized to the number of processors found on the server.

Special Considerations for IIS Deployment

When deploying a solution using OfficeAdapterDecoder in a web application, there are additional permissions required for the decoder to function.

Please see HOWTO: Set Needed COM Permissions In IIS for OfficeAdapterDecoder

INFO: OfficeAdapterDecoder Design Implications

Original Article:

Q10415 - INFO: OfficeAdapterDecoder Design Implications

Atalasoft Knowledge Base
https://www.atalasoft.com/kb2/KB/50083/INFO-OfficeAdapterDecoder-Design-Imp...