
Omtool DMS Libraries v4.0

Installation and Integration

Guide for the Custom DMS

May 2013



Omtool, Ltd.

6 Riverside Drive
Andover, MA 01810
Phone: +1/1 978 327 5700
Toll-free in the US: +1/1 800 886 7845
Fax: +1/1 978 659 1300

Omtool Europe

25 Southampton Buildings
London
WC2A 1AL
United Kingdom
Phone: +44/0 20 3043 8580
Toll-free in the UK: +44/0 80 0011 2981
Fax: +44/0 20 3043 8581

Web: <http://www.omtool.com>

© 2019 by Omtool, Ltd. All rights reserved. Omtool, AccuRoute and the Company logo are trademarks of the Company. Trade names and trademarks of other companies appearing in this document are the property of their respective owners.

Omtool product documentation is provided as part of the licensed product. As such, the documentation is subject to the terms outlined in the End User License Agreement. (You are presented with the End User License Agreement during the product installation. By installing the product, you consent to the terms therein.)

Permission to use the documentation is granted, provided that this copyright notice appears in all copies, use of the documentation is for informational and non-commercial or personal use only and will not be copied or posted on any network computer or broadcast in any media, and no modifications to the documentation are made. Accredited educational institutions may download and reproduce the documentation for distribution in the classroom. Distribution outside the classroom requires express written permission. Use for any other purpose is expressly prohibited by law.

Omtool and/or its suppliers make no guaranties, express or implied, about the information contained in the documentation. Documents and graphics contained therein could include typographical errors and technical inaccuracies. Omtool may make improvements or changes to the documentation and its associated product at any time.

Omtool support and sales

Online resources

The Omtool web site provides you with 24-hour access to documentation, software updates and other downloads, and detailed technical information that can help you troubleshoot issues. Go to <http://www.omtool.com/support> and log in using your customer number. Then click one of the following:

- **Knowledge Base** to access technical articles.
- **Downloads & Docs** to access online documentation, software updates, and downloads.

Customer service and technical support

Contact Omtool Customer Service or Technical Support using any of the following methods:

- **Phone:** +1/1 978 327 6800 or +1/1 888 303 8098 (toll-free in the US)
- **Fax:** +1/1 978 659 1301
- **E-mail:** customerservice@omtool.com or support@omtool.com

Technical support requires an active support contract. For more information, go to <http://www.omtool.com/support/entitlements.cfm>.

Sales, consulting services, licenses, and training

Contact Omtool Sales using any of the following methods:

- **Phone:** +1/1 978 327 5700 or +1/1 800 886 7845 (toll-free in the US)
- **Fax:** +1/1 978 659 1300
- **E-mail:** sales@omtool.com

Contents

Section 1: Installing the DMS Connector for a Custom DMS

Introducing the Omtool Custom DMS SDK v1.0	1-1
Purpose	1-1
Overall design	1-2
Server requirements.....	1-2
Pre-installation considerations	1-3
Document naming	1-3
Document XML schema	1-3
Capabilities XML schema	1-4
Registering a new custom DMS extension manually.....	1-5
Installing the Custom DMS sample files.....	1-5
Adding the Custom DMS connector	1-6
Completing the Custom DMS connector Configuration Wizard.....	1-6
Configuring a new Omtool Simple DMS Sample connector.....	1-6
Configuring a new Omtool StoreDocument Sample connector.....	1-7
Configuring an existing Omtool Simple DMS Sample connector.....	1-7
Configuring an existing Omtool StoreDocument Sample connector.....	1-8
Enhancements and resolved issues in this release.....	1-8
Release Notes	1-9

Section 2: Building an Omtool Custom DMS Integration

Using the Custom DMS extension API syntax	2-1
GetCapabilities.....	2-1
Configure.....	2-1
Initialize.....	2-2
ShowProfileUI.....	2-2
StoreDocument.....	2-2
Using the Omtool Custom DMS extension samples.....	2-3
vbSampleExtensionsimple.vbp	2-3
vbSampleExtensionstoreDoc.vbp	2-3

Section 3: Configuring Clients

Client requirements	3-1
Configuring AccuRoute Desktop.....	3-1
Configuring AccuRoute Client.....	3-2
Configuring FaxCenter.....	3-2
Configuring Swiftwriter	3-2
Client deployment methods for AccuRoute Desktop.....	3-3
Configuring Custom DMS for client rollout.....	3-3
Configuring Custom DMS manually	3-3

Glossary

Section 1: Installing the DMS Connector for a Custom DMS

This section includes:

- [Introducing the Omtool Custom DMS SDK v1.0](#) (1-1)
- [Server requirements](#) (1-2)
- [Pre-installation considerations](#) (1-3)
- [Registering a new custom DMS extension manually](#) (1-5)
- [Installing the Custom DMS sample files](#) (1-5)
- [Adding the Custom DMS connector](#) (1-6)
- [Completing the Custom DMS connector Configuration Wizard](#) (1-6)
- [Enhancements and resolved issues in this release](#) (1-8)
- [Release Notes](#) (1-9)

Introducing the Omtool Custom DMS SDK v1.0

Purpose

The Omtool Custom DMS SDK is a toolkit for developers to use to build new integrations between Omtool's products and external applications that store or process documents. The Omtool Custom DMS SDK extends Omtool's DMS Libraries architecture. An extension developed using the Custom DMS SDK will work with any of Omtool's existing applications that use the Omtool DMS Libraries. These Omtool applications are:

- AccuRoute Desktop
- AccuRoute Client
- DMS connector on the AccuRoute or Genifax server
- FaxCenter client (an optional client for AccuRoute or Genifax)

Omtool's engineering team has developed a number of DMS providers that work in the Omtool DMS Libraries architecture. These DMS providers have been developed using Microsoft C++ and are tightly linked to the internal software development and build environment of Omtool's core products. The Custom DMS SDK provides a layer that allows an external developer to build an extension that extends a custom DMS provider. Omtool has developed a custom DMS provider that links into the Omtool DMS Libraries; the external developer can then develop an extension to the custom DMS provider that will implement the custom integration functionality. Developing a custom DMS

extension is a simple process. It is intended that development be done in Visual Basic and that the extension has no compile or link dependencies on any of the Omtool core software. The Omtool Custom DMS SDK provides two sample implementations of custom DMS extensions to be used as a reference for developers and, in some cases, as a starting point for customization.

This section describes the Omtool Custom DMS SDK and how to use it to build a custom DMS extension. Developers may typically choose to use this Omtool Custom DMS SDK to build integrations with external document management, records management, case management, litigation support, content management, collaboration and document delivery systems.

If you are interested in obtaining the Omtool Custom DMS SDK, please contact Omtool Sales. (Go to [Omtool support and sales](#) on the inside front cover.)

Overall design

At a high level, the Omtool application (AccuRoute Desktop, for example) communicates with the Omtool DMS Libraries. The Omtool DMS Libraries can be configured to interface with a custom DMS provider; the custom DMS provider can be configured to use a specific custom DMS extension. Omtool is responsible for developing the Omtool application, the DMS Libraries and the custom DMS provider whereas the external developer is responsible for developing the custom DMS extension. Any number of DMS providers can be operational on a single system at the same time (for example, the iManage DMS provider and the custom DMS provider) and any number of custom DMS extensions can be operational on a single system at the same time.

The custom DMS provider allows a custom DMS extension to support two basic functions. The first invokes a document profile/import/check-in UI, and the second invokes a custom **StoreDocument** function. The document profile UI is intended for use on client applications to present the user with a dialog box that allows the user to describe the document that is being saved/sent to the external system. For example, in the case of a document management system this dialog box would prompt the user for the location in the document storage location to save the document to (server, folder, etc.) as well as values for any required or optional fields used to categorize the document (subject, billing code etc.). The **StoreDocument** function is intended for use on the Omtool server (in a DMS connector) and will save/send the document with its associated metadata to the external system.

The document profile UI is implemented as single **IDispatch** method call to **ShowProfileUI** on the configured **ProgID** of the extension. This method will return a single XML string that contains proprietary information used only by the DMS extension and its agents. The XML data then will later be handed back to the extension's **StoreDocument** method for use when saving/sending the document to the external system.

The **StoreDocument** mechanism can be implemented as one of two possible selections. The default mechanism implemented in the custom DMS provider itself and requiring no additional development by the external developer saves both the document(s) and the metadata (which is both Omtool-specific data about the document and the custom DMS extension's previously supplied XML data) to a drop folder that has been configured by the system administrator. The downstream application would then have to poll that drop folder to find and process the document and any associated metadata. A second mechanism, which requires the external developer to implement a **StoreDocument** method, allows the custom DMS extension to implement its own storage mechanism via an **IDispatch** call to **StoreDocument** on the configured **ProgID** of the extension.

Server requirements

The system where you install the DMS connector for Custom DMS must meet the following minimum requirements:

- AccuRoute v4.0

- DMS connector license
- AccuRoute Desktop v4.0

Pre-installation considerations

The custom DMS extension must register itself with the custom DMS provider shell so that it can be configured and used. Omtool suggests that this occurs when the custom DMS extension is installed. In order to register with the custom DMS provider shell the following entry must be made in the registry.

Under the `HKLM\Software\Omtool\CustomDMSExtensions` key, the `ProgID` of the COM object implementing the custom DMS extension must be placed as a string value. The name of the value is not relevant, but it is advised that it be unique to avoid conflicts with other installed custom DMS extensions.

Example:

```
HKLM\Software\Omtool\CustomDMSExtensions
Omtool.SampleExtensionOmtool.Sample.1
```

Document naming

When a document is processed and stored by the custom DMS extension - either with the custom DMS provider shell's default **StoreDocument** implementation or with the custom DMS extension's **StoreDocument** method—the filenames for the document created will adhere to the following naming convention.

The root name of the document will always be based on its original name. If the base file name already exists in the destination folder, a Date/Time stamp (in the format of `YYYYMMDDHHMMSS`) will be added.

Example:

```
MyDocument-20040909085504.PDF
```

When the Custom DMS is storing a document directly into a folder, the associated XML data will be stored in a file that shares the same root name, but with the extension `.XML`

Example:

```
MyDocument-20040909085504.PDF
MyDocument-20040909085504.XML
```

Document XML schema

The XML returned to the custom DMS extension via its chosen storage mechanism will be in the following format

```
<Document>
  <Custom>
    <Custom document-XML generated by the ShowProfileUI function>
  </Custom>
```

```

<Omtool>
  <DocumentName>[value from prDMSDocName</ DocumentName >
  <Originator>[value from prOriginator]</Originator>
  <FinalFormat>[value from prFinalFormCode]</FinalFormat>
  <BillingCode>[value from prBillingCode]</BillingCode>
  <OriginatorANI>[value from prOriginatorANI]</OriginatorANI>
  <OriginatorCSI>[value from prOriginatorCSI]</OriginatorCSI>
  <ProviderTSI>[value from prProviderTSI]</ProviderTSI>
  <JobIDText>[value from prJobIDText]</JobIDText>
  <OmtoolServer>[value from prDMSOmtoolServer]</OmtoolServer>
  <DateSubmitted>[value from prDateSubmitted] </DateSubmitted>
  <TemplateTags>
    <Tag name='Tag Name'>[Tag value]</Tag></TemplateTags>
  <RuleCustomProperties>
    <Property name='Prop Name'>[Prop value] </Property>
  </RuleCustomProperties>
</Omtool>

```

```
</Document>
```

Note that **TemplateTag** is an optional additional property that can be set on a message when it is submitted for processing through an Omtool server. A **RuleCustomProperty** is an optional additional property that can be set when creating a rule in the Omtool server.

Capabilities XML schema

The XML returned by the Custom Document extension **GetCapabilities** method will be in the following format. Extra nodes and values will be ignored.

```

< Capabilities >
  <Name>[name of extension]</Name>
  <Description>[description of extension]</Description>
  <Version>[Version of the custom DMS extension supported]</Version>
</ Capabilities >

```

Registering a new custom DMS extension manually

To create and enable a custom DMS extension the developer should:

- 1 Edit and build a custom DMS extension DLL (based on Omtool samples).
- 2 Register the DMS extension DLL with `regsvr32 <DLL path>`.
- 3 Create the following registry key on the computer that will use the custom DMS extension:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Omtool\CustomDMSExtensions\
```

- 4 Create the following string value for this key:

```
Name <friendly name of extension>, TYPE REG_SZ, Data <ProgID of extension>
```

With these steps complete then Custom DMS extension will be available in the enumerated list of DMS systems visible from the Omtool application.

The Custom DMS includes an InstallShield project that performs the above steps. Developers can modify this InstallShield project as appropriate and use it to install their own custom developed custom DMS extension.

Installing the Custom DMS sample files

When you install the Custom DMS sample files, the sample Custom DMS extension dll files are installed and registered for you.

To install the Custom DMS sample extension .dll files:

- 1 Open Windows Explorer and go to `...\Omtool\Tools\CustomDMSExample\`.
- 2 Double-click `setup.exe`. The **Welcome** screen appears.
- 3 Click **Next**. The **License Agreement** screen appears.
- 4 Read the License Agreement. and click **Yes** if you agree to the terms. (Click **No** to exit the setup if you do not agree to the terms.) The **Choose Destination Location** screen appears.
- 5 Click **Next** to accept the default destination location. (If you want to change the destination location, click **Browse**, navigate to the directory where you want to install the Custom DMS sample extension .dll files, click **OK**, and then click **Next**.) The **Start Copying Files** screen appears.
- 6 Review the installation settings and click **Next** to begin installing the Custom DMS sample extension .dll files.
- 7 Click **Finish**.

Adding the Custom DMS connector

Note The **Export Profile Information** and **Create Default Profile Information** buttons are not applicable for the Custom DMS connector at this time.

To add the Custom DMS connector to an AccuRoute server:

- 1 Right-click **Connectors** in the console tree and select **New > DMS**. The **License** screen appears.
- 2 Type the license key for the Custom DMS connector in the **License** text box and click **Next**. The **Display Name** screen appears.
- 3 Type a friendly name for this connector in the **Name** text box and click **Next**. The **DMS Type** screen appears.
- 4 Select a Tier 1 Custom DMS from the drop-down list and click **Next**. For example, if you are using the Omtool Custom DMS sample files, you can select either **T1 - Omtool StoreDocument Sample** or **T1 - Omtool Simple DMS Sample**. The **Congratulations** screen appears.
- 5 Click **Finish**.

The DMS configuration wizard launches and prompts you to configure the connector to support Custom DMS.

Note If you wish to install both sets of sample files, perform this procedure for each set.

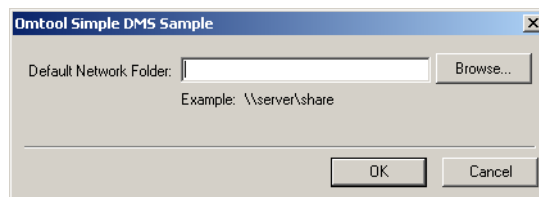
Completing the Custom DMS connector Configuration Wizard

Configuring a new Omtool Simple DMS Sample connector

The screen that appears immediately after you complete the new connector wizard differs depending on your custom application.

For the T1-Omtool Simple DMS Sample connector, the **Omtool Simple DMS Sample** screen appears immediately after you complete the new connector wizard.

Figure 1-A: Omtool Simple DMS Sample screen



To complete the configuration wizard for a new Omtool Simple DMS connector:

- 1 Specify the location of the **Simple DMS** folder in the **Default Network Folder** text box. (Use a UNC path. For example: `\\ComputerName\FolderName`) Enter the location of the Simple DMS folder, or click **Browse**, go to the **Simple DMS** folder, select it, and click **OK**.

If you are configuring the DMS connector on an AccuRoute server, you can leave this field blank. When routing documents on behalf of an AccuRoute Desktop or AccuRoute Client user, the server uses the DMS configuration data from the workstation where the user generated the Routing Sheet.

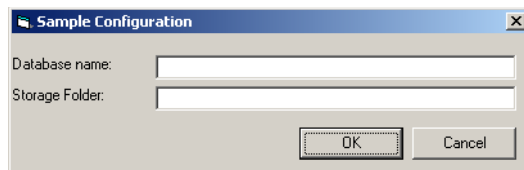
- 2 Click **OK**.

Configuring a new Omtool StoreDocument Sample connector

The screen that appears immediately after you complete the new connector wizard differs depending on your custom application.

For the T1-Omtool StoreDocument Sample connector, the Sample Configuration screen appears immediately after you complete the new connector wizard.

Figure 1-A: Omtool StoreDocument Sample Configuration screen

**To complete the configuration wizard for a new Omtool StoreDocument Sample connector:**

- 1 Enter the database name in the **Database name** text box. For example: `documents.mdb`
- 2 Specify the location of the Simple DMS folder in the **Default Network Folder** text box. (Use a UNC path. For example: `\\ComputerName\FolderName`) Enter the location of the StoreDocument DMS folder, or click **Browse**, go to the StoreDocument DMS folder, select it, and click **OK**.

If you are configuring the DMS connector on an AccuRoute server, you can leave this field blank. When routing documents on behalf of an AccuRoute Desktop or AccuRoute Client user, the server uses the DMS configuration data from the workstation where the user generated the Routing Sheet.

- 3 Click **OK**.

Configuring an existing Omtool Simple DMS Sample connector

Note The **Export Profile Information** and **Create Default Profile Information** buttons are not applicable for the Custom DMS connector at this time.

To complete the configuration wizard for an existing Omtool Simple DMS Sample connector:

- 1 Click **Connectors** in the console tree, and double-click the **Omtool Simple DMS Sample** connector in the details pan.
- 2 Click the **Advanced** tab.

- 3 Locate the **DMS Configuration** section, and click **Configure**. The **Omtool Simple DMS Sample** screen appears.
- 4 Perform steps one through two in [Configuring a new Omtool Simple DMS Sample connector](#) (1-6).
- 5 Click **OK**.

Configuring an existing Omtool StoreDocument Sample connector

Note The **Export Profile Information** and **Create Default Profile Information** buttons are not applicable for the Custom DMS connector at this time.

To complete the configuration wizard for an existing Omtool StoreDocument Sample connector:

- 1 Click **Connectors** in the console tree, and double-click the **Omtool StoreDocument Sample** connector in the details pane.
- 2 Click the **Advanced** tab.
- 3 Locate the **DMS Configuration** section, and click **Configure**. The **Sample Configuration** screen appears.
- 4 Perform steps one through three in [Configuring a new Omtool StoreDocument Sample connector](#) (1-7).
- 5 Click **OK**.

Enhancements and resolved issues in this release

This release of the Omtool DMS Libraries Custom DMS integration includes the following enhancements and resolved issues:

Enhancement: Custom DMS is now supported.

The following issues are resolved as of the AccuRoute Desktop v2.1.02 Update:

- In the Custom DMS connector properties, clicking **Create** under the **Default Profile Information** section on the Advanced tab resulted in the Profile dialog opening, but only half of it was visible. (See Knowledge Base article K03517). Ref. A6514
- The Profile dialog box would not close until you clicked **OK** twice, (See Knowledge Base article K03518.) Ref. A6516
- Without having configured the Custom DMS connector, a user was allowed to enter the profile details. (See Knowledge Base article K03525.) Ref. A6587
- When the Custom DMS profile window opened, focus was on the parent window. Focus shifted to the Profile window only when it was selected. (See Knowledge Base article K03625.) Ref. A6588
- When a Custom DMS window was closed in AccuRoute Desktop by clicking the Windows close icon (X), the following message appeared: “Failed to profile document” (See Knowledge Base article K03532.) Ref. A6589
- The tab order in the Sample Configuration window was not correct. (See Knowledge Base article K03626.) Ref. A6592

- The Custom DMS Profile window was not aligned in the Omtool Server Administrator. The positioning of the window did not display the right hand side of the window. (See Knowledge Base article K03633.) Ref. A6604
- If the extension of a document was changed in the Profile window (for example: from OmtoolStub.doc to OmtoolStub.pdf), the delivered document did not open correctly. (See Knowledge Base article K03632.) Ref. A6740

Release Notes

This section contains release notes and troubleshooting tips that are specific to the Omtool DMS Libraries.

Tip The Omtool Knowledge Base contains resolutions and informational articles on technical issues. To access the knowledge base, go to <http://www.omtool.com/support/index.cfm> and log in. Click Knowledge Base in the navigation pane. Search the knowledge base using an article number or keyword.

FaxCenter supports only one DMS application: FaxCenter can support only one DMS application, even though other Omtool applications such as the AccuRoute Desktop, and AccuRoute Client, and Swiftwriter can support multiple DMS applications concurrently.

All Omtool applications on the same system share the Omtool DMS Libraries configuration: When multiple Omtool applications are installed on the same system, these applications share the same DMS configuration data. If you make changes to the DMS configuration settings for one Omtool application, those changes affect all Omtool applications installed on the system.

Never open, edit, or delete a placeholder in a DMS application: When an AccuRoute Desktop or AccuRoute Client user creates a routing sheet for a document that should be routed to your company's DMS application, the Omtool DMS Libraries invoke the DMS application and creates a placeholder for the document. When the AccuRoute server routes the actual document to the DMS application, the DMS application overwrites the placeholder with the actual document. Never open, edit, or delete this placeholder.

Section 1: Installing the DMS Connector for a Custom DMS

Section 2: Building an Omtool Custom DMS Integration

This section includes:

[Using the Custom DMS extension API syntax](#) (2-1)

[Using the Omtool Custom DMS extension samples](#) (2-3)

Note Custom DMS connectors will only have limited functionality. You will be able to perform storage, document and profile tasks only. Billing lookup and validation, as well as merging, printing, and showing documents are not supported.

Using the Custom DMS extension API syntax

GetCapabilities

`Public Function GetCapabilities() As String`

This is an optional method for custom DMS extensions.

This method allows the custom DMS extension to provide the Omtool DMS Libraries and applications various information about the custom DMS extension. The most common information that would be provided by the extension would be the name of the extension, to be used in display within the Omtool applications (for example, “XYZ Document Management System”).

If this method returns a failure or unparseable XML, then it will simply be ignored and internal defaults assumed.

The name and description returned in the Capabilities XML will be used at selection and configuration time for both the administrator and end-user. If `GetCapabilities` is not implemented or returns a failure, the extension's name will be defaulted to the name of the value stored in `HKLM\Software\Omtool\CustomDMSExtensions` used to register the custom DMS extension with the custom DMS provider.

See [Capabilities XML schema](#) (1-4).

Configure

`Public Function Configure() As String`

This is an optional method for custom DMS extensions.

The `Configure` method allows the custom DMS extension to configure itself as deemed necessary. If it returns any value other than `S_OK` (success), then the custom DMS extension will be regarded as unusable.

Configure will be called after the administrator has selected “New Custom DMS connector” and selected the specific custom DMS extension. This allows the custom DMS extension to configure any required resources on the server. It is optional, but advisable, that configuration is stored in the returned XML data. This allows for the potential that the custom DMS extension is being remotely administrated. The configuration XML data returned will be stored within the Custom DMS configuration on the Omtool server. It will later be passed to the Initialize method before any other method (such as ShowProfileUI, StoreDocument) is called.

The schema of the Configuration XML is left to the discretion of the custom DMS extension developer. The custom DMS extension does not validate it.

Initialize

```
Public Sub Initialize(ByVal sConfigurationXML As String)
```

This is an optional method for all custom DMS extensions.

The **Initialize** method is called previous to other methods (with the exception of when **Configure & GetCapabilities** are called) to give the object a chance to reconstitute the settings the administrator made during the **Configuration** call.

This method is only called if it exists on the object.

ShowProfileUI

```
Public Sub ShowProfileUI(ByRef strDocXML As String, ByRef strDocName As String,  
ByVal lParentWnd As Long)
```

This is a required method for all custom DMS extensions.

The **ShowProfileUI** method is called when an end-user has issued a request to store a document through the chosen custom DMS extension. For example, in the AccuRoute Client when the end-user adds a new recipient and selects a particular Custom DMS as the recipient type, **ShowProfileUI** for the selected Custom DMS would be invoked.

The **ShowProfileUI** method is only required to set a valid XML string in strDocXML, and set the document's name in strDocXML. No other processing is required. It is strongly suggested that this method display a UI and acquire any properties necessary to store and process the document(s) later. The properties stored in the XML are not limited to end-user captured values, and may contain any data that the custom DMS extension wants to store along with the document.

lParentWnd will contain a valid window handle that the **ShowProfileUI** function can utilize to properly display the profile UI.

When a client of the Custom DMS requests to Profile a document, the custom DMS extension's object will be created via its configured **ProgID**, **Initialize** will be called (optionally), the **ShowProfileUI** method will be called, and then the object will be released (and subsequently destroyed).

See [Document XML schema](#) (1-3).

StoreDocument

```
Public Sub StoreDocument(ByVal sXML As String, ByVal sDocumentPath As String)
```

This is an optional method for all custom DMS extensions.

The **StoreDocument** method is called when a document is routed to the custom DMS extension. Routing occurs via the Omtool's server normal routing mechanism to the DMS connector configured to handle the custom DMS extension.

StoreDocument is responsible for parsing the passed in XML and storing the document in whatever way the custom DMS extension requires.

The document in **bstrDocumentPath** will be deleted once this method has returned. The implementation in **StoreDocument** must make a copy of this document before it returns if it requires the document to exist beyond the lifetime of the method call.

When a client of the custom DMS extension requests to store a document, the custom DMS extension's object will be created via its configured **ProgID**, **Initialize** will be called (optionally), the **StoreDocument** method will be called, and then the object will be released (and destroyed).

See [Document XML schema](#) (1-3).

Using the Omtool Custom DMS extension samples

The Custom DMS SDK includes two sample implementations of a custom DMS extension. Both are implemented using Visual Basic. The primary difference between the two is one does not implement a **StoreDocument** method and therefore uses the standard **StoreDocument** behavior implemented in the custom DMS provider shell and the other does implement a custom **StoreDocument** method.

vbSampleExtensionsimple.vbp

This sample only implements the **GetCapabilities** and **ShowProfileUI** functions. Since the **StoreDocument** is not implemented in this sample, there will be no custom UI while the DMS is being configured, and when a document is routed to the sample DMS the default processing will occur. While the DMS is being configured a default screen will be shown that will prompt for a default storage folder for documents that get submitted through the custom DMS. This is implemented outside the scope of the custom DMS extension. Also, with no **StoreDocument** implemented, the custom DMS provider shell will automatically pass documents and accompanying xml file to the pre-configured storage folder.

vbSampleExtensionstoreDoc.vbp

This sample demonstrates a fully implemented custom DMS extension. The **Configure** function will prompt with a custom DMS configuration screen with fields for a MS Access database to store document information, and a storage folder path. The **StoreDocument** function copies the file pointed to by its path to previously configured storage folder, and then the document xml data and document path are written to the configured MS Access database.

Section 3: Configuring Clients

This section includes:

- [Client requirements](#) (3-1)
- [Configuring AccuRoute Desktop](#) (3-1)
- [Configuring AccuRoute Client](#) (3-2)
- [Configuring FaxCenter](#) (3-2)
- [Configuring Swiftwriter](#) (3-2)
- [Client deployment methods for AccuRoute Desktop](#) (3-3)

Client requirements

The system where you install the MyFolder integration must meet the following minimum requirements:

- AccuRoute Desktop v4.0 or AccuRoute Client v4.0
- FaxCenter v3.4
- The AccuRoute properties of the Registered User settings must have the Custom DMS type enabled.
- The Custom DMS extension must be registered on each workstation. See [Registering a new custom DMS extension manually](#) (1-5).

Configuring AccuRoute Desktop

To add Custom DMS functionality to AccuRoute Desktop:

- 1 Start AccuRoute Desktop and log in.
- 2 Click **Tools** on the menu bar and select **Configure DMS**.
- 3 Select **Custom DMS** from the menu that appears. The screen that appears differs depending on your custom application.
- 4 At this point, the configuration is identical to that of the connector configuration. For example, perform steps one through two in [Configuring a new Omtool Simple DMS Sample connector](#) (1-6) or steps one through three in [Configuring a new Omtool StoreDocument Sample connector](#) (1-7).

Configuring AccuRoute Client

To add Custom DMS functionality to AccuRoute Client:

- 1 Start AccuRoute Client and log in.
- 2 Click **Tools** on the menu bar and select **Configure Custom DMS**. The screen that appears differs depending on your custom application.
- 3 At this point, the configuration is identical to that of the connector configuration. For example, perform steps one through two in [Configuring a new Omtool Simple DMS Sample connector](#) (1-6) or steps one through three in [Configuring a new Omtool StoreDocument Sample connector](#) (1-7).

Configuring FaxCenter

To add Custom DMS functionality to FaxCenter:

- 1 Start FaxCenter and log in.
- 2 Click **Tools** on the menu bar and select **Configure DMS**. The Document Management Configuration screen appears.
- 3 Select Custom DMS and click **Configure**. The screen that appears differs depending on your custom application.
- 4 At this point, the configuration is identical to that of the connector configuration. For example, perform step one in [Configuring a new Omtool Simple DMS Sample connector](#) (1-6) or steps one through two in [Configuring a new Omtool StoreDocument Sample connector](#) (1-7).
- 5 Click **OK**.

Configuring Swiftwriter

To add Custom DMS functionality to Swiftwriter:

- 1 Start the Swiftwriter Configuration application in the Control Panel¹. The Swiftwriter Configuration screen appears.
- 2 Click **Custom DMS** in the list so that the option text is highlighted and click **Properties**. The screen that appears differs depending on your custom application.
- 3 At this point, the configuration is identical to that of the connector configuration. For example, perform step one in [Configuring a new Omtool Simple DMS Sample connector](#) (1-6) or steps one through two in [Configuring an existing Omtool StoreDocument Sample connector](#) (1-8).
- 4 Select **Custom DMS**, click **Apply**, and click **OK**.

1. If you cannot access the Control Panel, consult a network administrator. Access to the Control Panel might be disabled.

Client deployment methods for AccuRoute Desktop

There are two methods available for deploying clients to individual workstations with Custom DMS functionality:

- Configure AccuRoute Desktop to support Custom DMS on one client workstation. Then, copy the DMS Libraries configuration file to the Omtool DMS libraries setup directory where it can be propagated to client workstations during installation. Please see [Configuring Custom DMS for client rollout](#) (3-3) below.
- Install AccuRoute Desktop on each individual workstation and perform the configuration steps for Custom DMS manually.

Configuring Custom DMS for client rollout

This procedure is entirely optional. However, if you do not complete this procedure before deploying the Omtool DMS Libraries, you (or the user) must configure the libraries to support Custom DMS manually after you install AccuRoute Desktop.

To pre-configure the Omtool DMS Libraries for deployment:

- 1 Install AccuRoute Desktop v4.0 on a workstation.
- 2 Configure the Omtool DMS Libraries to support Custom DMS. Select the settings that should apply to all workstations in your environment.
- 3 Go to the \Program Files\Common Files\Omtool directory on the workstation where AccuRoute Desktop v4.0 is installed, and copy the **OmDMSDB.xml** file to the clipboard.
- 4 Go to the Omtool server where AccuRoute Desktop setup files are located, and paste the file to the following location. When prompted to replace the existing file, click **Yes**.

...\OmtoolServer\Clients\AccuRouteDesktop\Common\Omtool

You have pre-configured the Omtool DMS Libraries setup with the configuration data for Custom DMS. When you are ready to deploy Custom DMS, launch **setup.exe** in AccuRoute Desktop v4.0 installation directory.

Configuring Custom DMS manually

To configure Custom DMS manually, see [Configuring AccuRoute Desktop](#) (3-1).

Glossary

Custom DMS extension

This component is written and maintained by the external developer. It is the component that plugs into the custom DMS provider shell that communicates with the external system in accordance with the custom DMS extension API. It is implemented as an IDispatch-based COM object.

Custom DMS extension API

This is the specification that the custom DMS extension must adhere to in order to work with the custom DMS provider shell.

Custom DMS provider shell

This component is written and maintained by Omtool. It is the component that the custom DMS extension communicates with to interact with the DMS Libraries and the Omtool application.

DMS Libraries

This is the architecture and components used to interface Omtool applications with external document storage and communication systems. DMS specifically refers to Document Management System. Many of these external applications are in fact document management systems. However, other systems may be records management, case management, litigation support, content management, collaboration and document delivery systems.

DMS provider

A DMS provider is an interface for communication with a specific external DMS system. Omtool engineers develop DMS providers. The custom DMS provider shell is one such DMS provider. The iManage DMS provider is another such DMS provider.

