Panagon Document Warehouse for SAP

Programmer's Handbook

Release 5.0.1
April 2002

Table of Contents

Legal Notice

This document contains information proprietary to FileNET Corporation (FileNET). Disclosure, reproduction, or use of any FileNET proprietary information from any part of this document is prohibited without prior written permission from FileNET.

Even though FileNET has tested the hardware and software and reviewed the documentation, FileNET makes no warranty or representation, either express or implied, with respect to the hardware, software, or documentation, their quality, performance, merchant-ability, or fitness for a particular purpose. FileNET has made every effort to keep the information in this manual current and accurate as of the date of publication or revision. However, FileNET does not guarantee or imply that this document is error free or accurate with regard to any particular specification. As a result, this product is sold as is, and you the purchaser are assuming the entire risk as to its quality and performance.

In no event will FileNET be liable for direct, indirect, special, incidental, or consequential damages resulting from any defect in the hardware, software, or documentation, even if advised of the possibility of such damages. In particular, FileNET shall have no liability for any programs or data stored in or used with FileNET products, including the costs of recovering such programs or data.

Some states do not allow the exclusion or limitations of liability for incidental or consequential damages, so the above limitation or exclusion may not apply to your installation. Certain rights may vary from jurisdiction to jurisdiction.

No FileNET agent, dealer, or employee is authorized to make any modification, extension, or addition to the above statements.

Microsoft®, Windows® and Windows NT® are registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

SAP®, R/3®, ABAP/4®, SAP ArchiveLink®, SAP Business Workflow®, are registered trademarks of SAP AG.

About this Manual

The Panagon Document Warehouse™ for SAP Programmer's Handbook, dated April 2002 supports the 5.0 & 5.0.1 release of the Panagon Document Warehouse for SAP software. It covers the issue of extending the existing product by writing new components for Views and searches for SAP master objects. Furthermore it is possible to create new applications using the DMS object model, which is part of the client software.

Skill Level Requirements

The programmer will need a good understanding of the FileNET and SAP systems, as well as a working knowledge of Windows NT/2000 or Windows 95/98 and the preferred development environment. Specifically, he will need to understand the following:

- FileNET system operations
- SAP 3.1, 4.0, 4.5 or 4.6 system
- PC Operations
- Windows 95/98 or Windows NT 4.0/2000 procedures
- the preferred development environment (this manual uses Microsoft Visual Basic)
- Microsoft's ActiveX architecture
- programming with FileNET's Panagon IDM Desktop
- OOA and OOP

Related Publications

The Panagon Document Warehouse software integrates the standard FileNET system with the SAP system via the SAP ArchiveLink interface. The following FileNET manuals and SAP documentation will prove useful in software installation, operation, and administration:

- Panagon Document Warehouse for SAP Installation & Configuration Manual
- Panagon IDM Desktop online documentation
- Panagon IDM Desktop for R/3 online documentation
- Online documentation for SAP System Release 3.1, 4.0, 4.5 and 4.6

Tips, Notes, Cautions, and Background Information

Throughout this manual, there are assorted brief messages designed to draw your attention to different kind of information:

Tip: Indicates hints to improve efficiency of performing a task.

Note: Provides important information, such as situations that can affect the outcome of an operation or affect some other part of the system.

Caution: Signals, where possible loss of data or time may occur.

Background: *Explains the concepts behind specific features and provides helpful information for understanding the design of SAP ArchiveLink and Document Warehouse for SAP.*

Education

FileNET offers introductory and advanced classes for system administrators, developers, management, and support personnel. These classes combine lecture and lab sessions to provide both conceptual understanding of the FileNET system and practice in its operation. For more information on class content and schedules, please visit the Education topics in the Services and Support area of FileNET's web site (www.filenet.com).

You can also use the following phone numbers to request information and to ask questions.
Local: 1.714.327.3412
Toll free: 1.888.FNEDUC8 (888.363.3828)

Comments & Suggestions

FileNET invites all customers to communicate with the Documentation group on any question or comment related to FileNET manuals and online help. Fax, phone, mail, or email any questions or comments to Mike Calvert at one of the numbers or addresses listed below. We guarantee a response to each communication within one week. Your suggestions help us to improve the products that we deliver.

Mike Calvert
Director of Documentation
FileNET Corporation
3565 Harbor Boulevard
Costa Mesa, California 92626-1420
Phone: 1.714.327.3449
Fax: 1.714.327.7153
Email: calvert@filenet.com

Introduction

This chapter describes the architecture of Panagon Document Warehouse. First, an overview of the Client Components of Panagon Document Warehouse for SAP and their architecture is given. Then, the concept of a custom View is discussed. Furthermore there is an overview regarding using DMS.

Panagon Document Warehouse for SAP Overview

FileNET has integrated its standard system with the SAP system using SAP ArchiveLink software. The result is an application called FileNET Panagon Document Warehouse for SAP. Panagon Document Warehouse enables SAP's business applications to process FileNET document images and provides document and data archiving.

Panagon Document Warehouse for SAP consists of two parts: IDM Services for R/3 (server components) and IDM Desktop for R/3 (client components). IDM Services for R/3 run on Windows NT/2000 server platform. IDM Desktop for R/3 runs on the Windows platforms (Windows NT 4.0, or Windows 95/98 or Windows 2000). This Programmer's Handbook discusses extending IDM Desktop for R/3.

System Architecture of the IDM Desktop for R/3

The IDM Desktop for R/3 supports two different interfaces to SAPGUI and SAP's Archive Link software:

- the DLL-Interface
  This interface is used by the ArchiveLink Viewer to access the documents it is asked to display. It is supported by the IDMForR3.DLL.
- the OLE-Interface
  This interface is used by SAPGUI to display documents in the Panagon IDM Viewer, to obtain documents for linking etc. It is supported by the Desktop for R/3. For obtaining a document for linking, it uses one of a number of Views (formerly known as "Catalog Interface Modules" (CIMs).

The following diagram shows the different components involved:
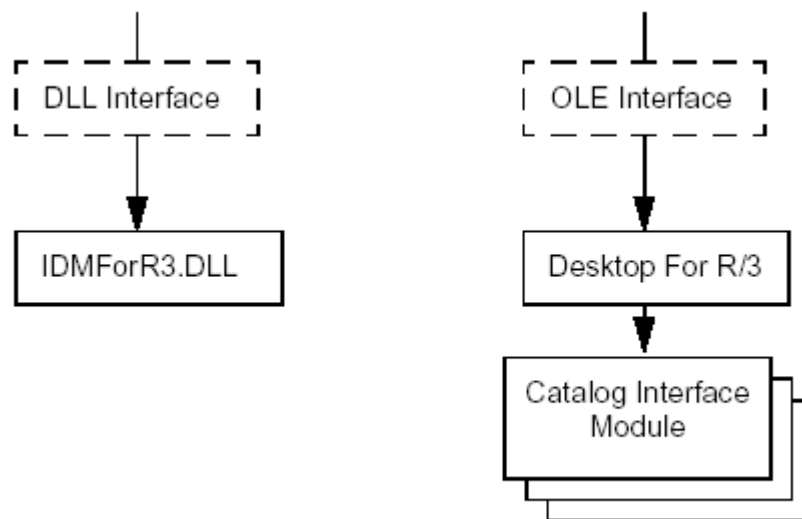
Linking a Document (Views)

Before linking a document to an SAP transaction, the user usually opens IDM Desktop for R/3, selects the document he wants to assign, displays it in the Panagon IDM Viewer, and then switches to SAPGUI and links the document. Linking means that SAPGUI calls IDM Desktop for R/3 and requests a document id. This

document id is the link between an SAP object and a document in Panagon IDM Services and will be used for displaying the document later.

There are several ways to select a document for assignment. Each way is implemented as a View. Two standard Views are shipped with IDM Desktop for R/3:

- Panagon Browse for R/3
  Lets the user browse in the folder hierarchy of IDM Image Services or IDM Content Services with a Microsoft Explorer-style interface. Even searching is possible.
- Panagon Queue for R/3
  Lets the user access the contents of a FileNET WorkFlo queue that holds a number of documents ready for processing in the SAP system.

Although these two Views provide flexible ways of finding or creating documents for linking, it is possible that these standard Views do not fulfill the specific needs of a customer. Therefore, Panagon Document Warehouse gives the ability to write a custom View that the IDM Desktop for R/3 uses in addition to the standard Views. This Programmer's Handbook discusses the concepts involved in writing an own View and lists the interfaces, methods and parameters this custom View has to support.

Linking a SAP Master Object (DMS)

The DMS interface provided by the SAP system connects external documents, folders, and objects from FileNET repositories with master data within the SAP system. Master data might be a material master, an equipment master, or a customer master. This functionality is provided by different dynamic link libraries delivered with Panagon Document Warehouse for SAP release 5.0.1. These dynamic link libraries are the foundation for an object model based on Microsoft COM technology. The IDM Desktop for R/3 will be delivered with out-of-the-box functionality for connecting FileNET objects with SAP material master. These functionality is available via the DMS menu.

Views

Function Overview

Panagon Document Warehouse for SAP allows to write a custom View. IDM Desktop for R/3 provides an MDI window and a generic MDI child form containing an empty OLE Container control. This OLE Container contains the run-time instance of a View. Each View instance belongs to a distinct MDI child-window instance. A View is an ActiveX Control that provides specific methods and a certain basic behavior. What the View actually does, i.e. how it obtains a document id, is up to the customer. A View can have its own menu, its own status bar and has to meet some behavior requirements. It has to know about some IDM Desktop for R/3-wide settings stored in the registry and can use the LoginManager feature of IDM Desktop for R/3. In the following sub-sections and sections information on these topics are given.

- Basic methods: See section "Startup, Shutdown and Save Settings" on page 9.
- Interaction with the IDM Desktop for R/3: See sub-section "Providing a Document ID" on page 9.
- How to help the user finding a document: See sub-section "Advanced Interaction" on page 11
- Guidelines on achieving the same look and feel as the standard Views: See sub-section "View Behavior" on page 11.
- For information regarding saving SLUs and LoginManager, please refer to section "LoginManager" on page 24.
- For general information about using the Windows Registry for reading and writing settings and about logging, please refer to section "General Settings" on page 26.

It helps to understand what is needed in order to write a custom View:

Startup, Shutdown and Save Settings

The following basic methods have to be exported by the ActiveX Control:

StartUp ()	IDM Desktop for R/3 calls "StartUp" after having instantiated the View in the MDI child window's OLE Container. "StartUp" may also be called when re-initializing.
ShutDown ()	"ShutDown" is called before destroying the instance of the View.
Save Settings ()	The method "SaveSettings" is called explicitly by IDM Desktop for R/3 when the user requests to save the settings (for example via menu item or when closing an View or the application). An View should not save its settings in method "ShutDown".

Providing a Document ID

This section discusses the general interaction between the IDM Desktop for R/3 and the active View when SAPGUI requests a document id.

Interaction between IDM Desktop for R/3 and View

When SAPGUI requests a document id, IDM Desktop for R/3 passes this request to the currently active View, the one with the focus, through a call to method "CGetDocumentId". The View returns its current document or folder id and an object type. If IDM Desktop for R/3 can pass this id to SAPGUI, it calls "RgetDocumentId" to inform the View that the id has been processed successfully.
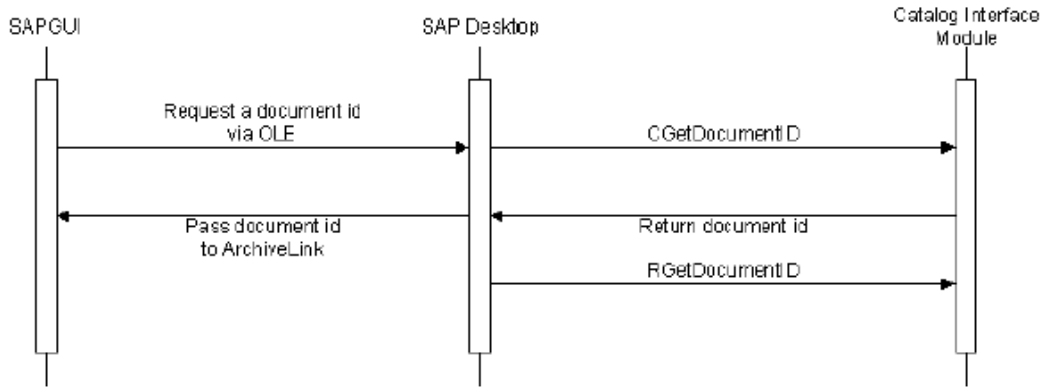
This means that an View should be aware that "CGetDocumentId" is called to get an object id, but the object id has not been passed to SAPGUI until the call to "RgetDocumentId". This is important for some Views. The View Queue for example, returns the document id of its currently selected queue entry. This queue entry will be deleted if the document id has been passed to SAPGUI successfully, but must not be deleted if it fails. This means that the deletion must occur when processing the "RgetDocumentId".

The Method "CGetDocumentId"

CGetDocumentId ()	
Parameter:	*Description*
sLibraryId	IN – Library document or folder has to belong to (might be empty and can be ignored then)
lRetObjId	OUT - document or folder id.
lRetObjType	OUT - idmObjTypeDocument (=1) or idmObj-TypeFolder (=2).
bRetSuccess	OUT - (0 = OK,1 = no Object Id returned)

What to do in "CGetDocumentId"?

Find out first, if the View has a current valid object id to be passed on to SAPGUI. If this is not the case, the user needs to be informed and it is also possible to switch to "Advanced Interaction mode" (for more information, please refer to sub-section "Advanced Interaction" on page 11). If the View has an object id, check whether the object belongs to the given library. If so, remember the id as a reference for "RgetDocumentId" and return the id along with the object type. If the library does not match, inform the user and possibly switch to "Advanced Interaction mode". On all further calls to "CGetDocumentId", check whether the View is awaiting a call to "RgetDocumentId". These are always called in pairs, so two succeeding calls to "GetDocumentId" without a call to "RGetDocumentId" in between is not allowed.

For example, View Queue locks the currently selected queue entry and remembers the document id.

The Method "RGetDocumentId"

RGetDocumentId ()	
Parameter:	*Description*
vObjId	IN – unchanged vRetObjId of CgetDocumentId
bSuccess	IN - True = object was succesfully passed to SAPGUI, False = object id was not succesfully passed to SAPGUI

What to do in "RgetDocumentId"?

First, check whether the View actually awaits a call to "RgetDocumentId". If not, ignore the call. Use the incoming object id as a reference to the preceeding call to "CGetDocumentId". Process the current object according to the Views behavior now.

For example, on success, View Queue deletes the locked queue entry and may fetch the next entry. On failure it unlocks the queue entry and informs the user.

Advanced Interaction

When a user tries to assign a document to an SAP object and the currently active View cannot provide a document id (e.g. because no document has been selected), SAPGUI displays an informational message that no documents were scanned in. It could also be possible that the user selects an object, but SAPGUI requests a document id from a different library. In this case the user should get an information about it and should be able to make a new selection. The document id of the newly selected document will be returned to SAPGUI. Advanced Interaction is a possible concept to allow this.The key to Advanced Interaction is the "CGetDocumentId" method, which does not return until the user selects a valid object (document or folder) or explicitly cancels the assignment. This can be done by providing two menu items, one for continuing the assignment and one for cancelling it.

The standard Views implement the following behavior in "CGetDocumentId":

```
Public Sub CGetDocumentId(...)
//Advanced Interaction
If (Nothing Selected) Or
        (Selection does not match requested Library) Then
        MsgInfo "Please select a document or folder before ..."
        //Continue Assignment is enabled by other methods
        //as soon as something is selected for assignment
        //Cancel is always enabled
        Disable MenuItem "Continue Assignment"
        Enable MenuItem "Cancel Assignment"
        //Wait until one of the two menu items has been clicked
        WaitForEvent "Continue Assignment" Or "Cancel
        Assignment"
        Disable MenuItem "Continue Assignment"
        Disable MenuItem "Cancel Assignment"
End If
//Rest of CGetDocumentId
If Something Selected Then
        ...
End Sub
```

View Behavior

GUI Guidelines

An View can have a menu. It has to be negotiated to the middle position (left and right are reserved for the MDI window). An View can also have a status bar. The MDI window and the MDI child window do not provide a status bar or any other visual means for displaying information. An View should react to resizing. When the MDI child window and the OLE Container are resized, the View should adapt to the new size.

DMS

Function Overview

The IDM Desktop for R/3 comes with a module (Panagon Browse for R/3), allowing to select a document, a specific version of a document, or a folder, and linking this object to a material master within the SAP system. This module uses the DMS object model based on COM technology, along with some COM objects for searching and retrieving information about material master in SAP. In order to connect FileNET objects to additional types of master data within SAP, Panagon Browse for R/3 offers an interface for adding individual search and retrieval modules/objects.

Therefore this manual provides information about the DMS object model and the Panagon Browse for R/3 interface. Since the object model is based on Microsoft COM technology, all programming examples in this manual are using the syntax of Visual Basic 6. The official supported programming languages by FileNET are Microsoft Visual Basic 6 and Microsoft Visual C++ 6, but the objects shall work with all programming languages supporting COM technology.
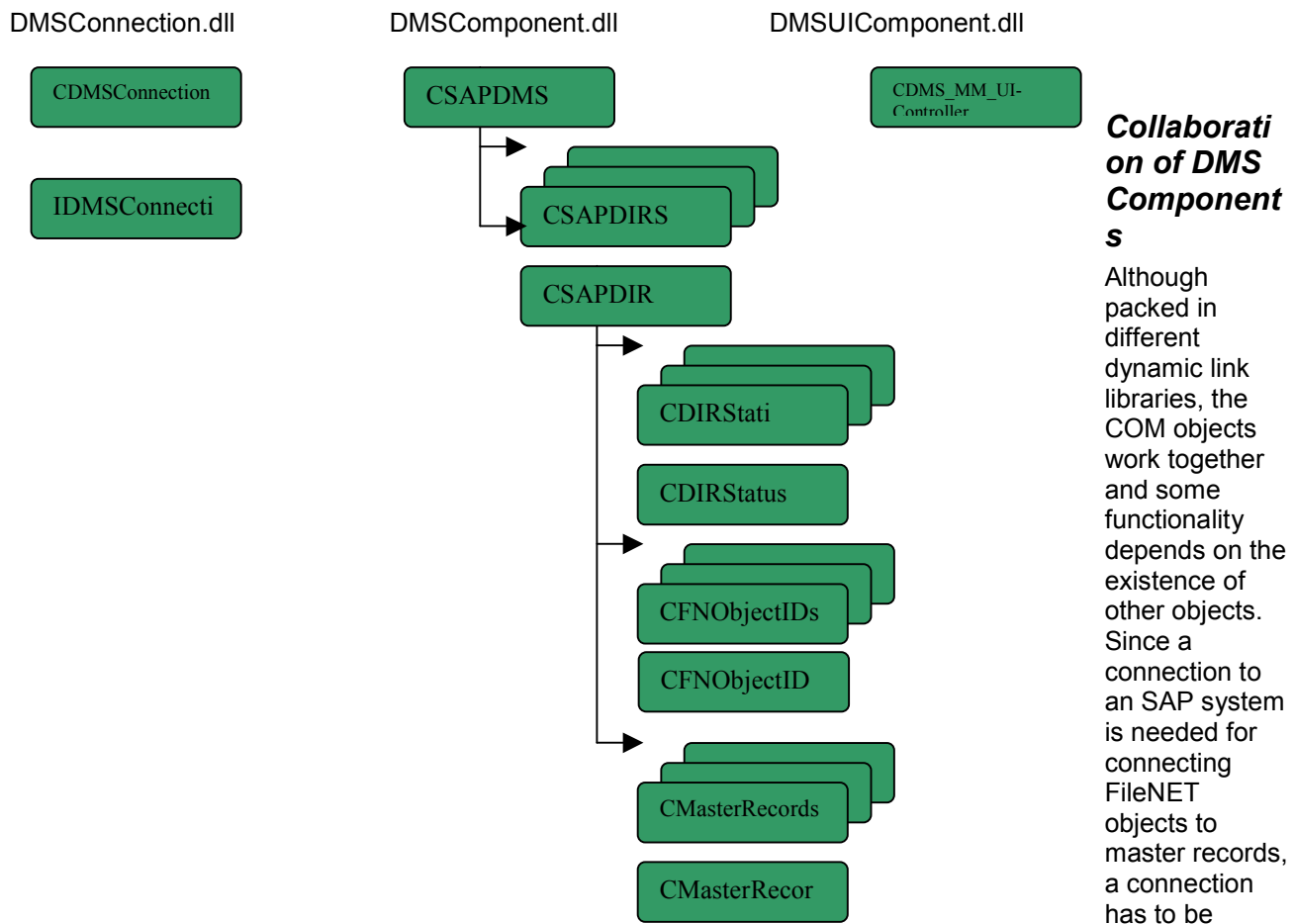
The interface that calls functions and objects within SAP, is called the BAPI interface (Business Application Programming Interface). The SAP BAPI interface for DMS functions is available with release 4.5 of the SAP system. The BAPI interface is the official supported interface for further releases of SAP. DMS functionality based on the older, so-called CAD interface is no longer supported by SAP. Since the COM objects are using the BAPI interface for calling DMS functions, the SAP system must be at least release 4.5 or higher. SAP releases 4.0 or less are not supported for the DMS functionality.

In the following sub-sections and sections information on these topics are given.

- An overview of the collaboration of the DMS Components, please refer to section "Collaboration of DMS Components" on page13.
- An overview of all objects and their functions of the DMSConnection.dll, please refer to section "DMSConnection.DLL" on page 13.
- An overview of all objects and their functions of the DMSComponnent.dll, please refer to section "DMSComponent.DLL" on page 15.
- An overview of all objects and their functions of the DMSUIComponent, please refert to section "DMSUIComponent.DLL" on page 18.
- For information regarding saving SLUs and LoginManager, please refer to section "LoginManager" on page 24.
- For general information about using the Windows Registry for reading and writing settings and about logging, please refer to section "General Settings" on page 26.

DMS Components

The DMS Components are delivered in form of three ActiveX dynamic link libraries, which provide the DMS COM objects. These libraries are installed with the Panagon Document Warehouse for SAP client software. The names of these dynamic link libraries are the DMSConnection.dll, the DMSComponent.dll, and the DMSUIComponent.dll. Three additional SAP function ActiveX controls will be installed (wdtfuncs.ocx, wdtocx.ocx, and wtdlog.ocx), which are needed by the dynamic link libraries, and another dynamic link library for the RFC communication (librfc32.dll). The components provided by SAP will only be installed, if they are not already installed by the SAP client installation (SAPGUI). The following figure gives an overview of the COM objects. With the help of the provided COM objects, a programmer can create powerful applications for connecting all different types of SAP master records with FileNET objects, in an easy to use way.

DMSConnection.dll DMSComponent.dll DMSUIComponent.dll

Collaboration of DMS Components

Although packed in different dynamic link libraries, the COM objects work together and some functionality depends on the existence of other objects. Since a connection to an SAP system is needed for connecting FileNET objects to master records, a connection has to be established. The CDMSConnection object can do this, meaning if a connection with the login method of the connection object got created, a CSapDMS object can be created, which automatically uses this connection. Objects can be registered at the CDMSConnection object and will be notified in case the connection parameters have been changed. An interface (IDMSConnectionObserver) is provided, which can be used to register own objects at the CDMSConnection object.

The CSapDMS object is the root object for operations with Document Info Records within the SAP system. The CSapDMS object provides methods for creating DIR objects, a local representation of Document Info Records within the SAP system. Other methods return a collection of DIR objects, which are linked to a FileNET object.

The CDMS_MM_UIController object offers methods for searching for material masters within the SAP. Information of material masters can be shown in a graphical user interface. This object is used for functionality of the advanced Panagon Browse for R/3 and uses the connection provided by the CDMSConnection object.

DMSConnection.DLL

The DMSConnection.dll provides one object for connecting to a SAP system and an interface for objects, which need to observe this connection. The CDMSConnection offers functions to logon to a SAP system and to log off from a SAP system.

All custom COM objects, which use the CDMSConnection object and need information about changes on the CDMSConnection object, must implement the IDMSConnectionObserver interface and must be registered on the CDMSConnection object. A registered custom COM object will be notified when the state of the CDMSConnection object has changed.

Using Observer Pattern

With the Observer Pattern it is possible to define one to many dependencies between objects. If the observed object changes its state, all other objects will be informed about the changes. Within the DMSConnection.dll the Observer Pattern ensures that all components use the same connection object to the SAP System.
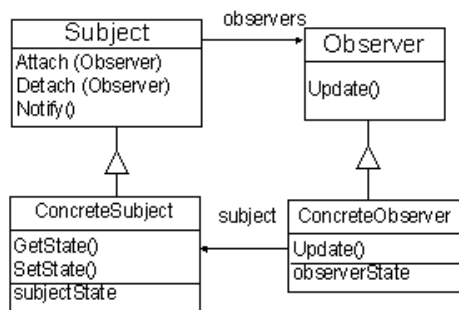
CDMSConnection

The CDMSConnection object provides functions to logon to SAP and to logoff from SAP. The 'IsConnected' function returns the information, if a connection to a SAP system exists. With the register function a reference to an Observer object can be added to the CDMSConnection. The registered object will be notified, if the state of the CDMSConnection object has changed. With the unregister function the CDMSConnection object resets its reference to the Observer object.

Function overview:

Logon()	Function to log on to a SAP system.
Parameter:	*Description*
User	User
Password	Password
Client	SAP client (mandant)
Language	A SAP supported language
ApplicationServer	IP address or aliasname of the SAP Server
SapSystemNumber	System number
RfcTrace	On/Off write trace file
RouterString	Router string to a SAP Server
LogonDialog	Use a logon dialog instead of a silent logon

Logoff()	*Function to log off from a SAP system.*

Register()	Function to register an Observer Object to the DMS Connection object.
Parameter:	Description
IDMSConnectionObserver	Reference to the object that gets registered. The object must be a type of IDMSConnectionObserver.

Unregister()	Function to unregister an Observer Object from the DMS Connection object.
Parameter:	Description
IDMSConnectionObserver	Reference to the object that gets unregistered. The object must be a type of IDMSConnectionObserver.

IsConnected()	Indicates if the CDMSConnection object holds a connection to the SAP system.

GetConnection()	These function returns a SapConnection object. This object

	holds properties about the current SapConnection and can be used in different delivered SAP system controls.

IDMSConnectionObserver

The IDMSConnectionObserver interface defines the method and the property that will be used from the CDMSConnection, if the state of the CDMSConnection object has changed. To ensure that the CDMSConnectionObserver can call this function, only objects which have implemented this interface can register at the CDMSConnection object.

Function overview:

Update()	Definition of the interface for all observers.
ObserverKey	Identifier for the observer object. Will be set after registration on the CDMSConnection object.

DMSComponent.DLL

The DMSComponent.dll offers a set of objects for linking FileNet objects with SAP master record objects. The central object of the DMSComponent.dll is the CSapDms object. To work correctly, the DMSConnection.dll needs a reference to the CDMSConnection object. This reference can be set with the 'setConnection' function of the CSapDms object. Furthermore, the CSapDms allows to create a CSapDir object with the functions 'CreateDir' and 'GetDir' locally.

The link object between master record and FileNET objects is the SAP Document Info Record (DIR). The corresponding object in the DMSComponent.dll is the CSapDir object. For the FileNET objects and the SAP master record objects two objects are available (CFnObjectId and CMasterRecord). Additional, for all those objects special collection objects exists (CSapDirs, CFnObjectIds, und CMasterRecords).

CSapDms

The CSapDms object is the central object within the DMSConnection.dll. The object offers functions to create CSapDir object and to retrieve data of an existing SAP DIR object. For accessing the SAP system the DMSComponent.dll needs a reference to a CDMSConnection object, which delivers the necessary logon information. The responding function for this step is the 'SetConnection' function. The 'getDirs' function returns a collection with all CSapDir objects, which are linked to a certain FileNET object.

Function overview:

SetConnection()	Function that provides a reference to the CDMSConnection object, that gets used by the DMSConnection.dll. This function should be done before using the CSapDms object.
Parameter:	*Description*
CDMSConnection	Connection object to call RFC functions within an SAP system.

CreateDir()	Function to create a new CSapDir object. This function should only be used in order to create a new CSapDir object with no corresponding DIR element in the SAP system.
Parameter:	*Description*
DocumentType	Type of DIR known in the SAP system[1]
DocmentNumber	Document number[2]
DocumentPart	DIR Part number
DocumentVersion	DIR Version number

---

[1] The different DIR types are defined within the SAP IMG (Cross Applications -> Document Management System)

GetDir()	Function to create a CSapDir object with the information's of an existing DIR object including all existing links to FileNET object(s) and master record(s).
Parameter	Description
DocumentType	Type of DIR known in the SAP system
DocumentNumber	Document number
DocumentPart	DIR part number
DocumentVersion	DIR Version number

GetDirs()	Function to retrieve all DIR objects linked to a FileNET object. With the information of the retrieved DIR a collection of DIR objects will be created.
Parameter:	Description
FnObjectID	ID of the FileNET object, which is linked to DIRs

CSapDir

The CSapDir object represents a SAP DIR object. The primary key of a SAP DIR object is the combination of the DIR type, DIR number, the DIR part, and DIR version. In this use case the SAP DIR object is only used as a link between the master record and the FileNET object. Therefore only the important properties are represented on a local CSapDir object.

SaveAsNew()	This function creates a new SAP DIR object with the information of the CSapDir object within the SAP system. Therefore the function can only be called on CSapDir with no corresponding SAP DIR objects. If the property MasterRecords and/or the property CFnObjects is/are given, the links to the SAP DIR object will be created.
Save()	The function saves the properties of the local CSapDir object within the SAP system on an existing DIR. The function is similar to the 'saveAsNew' function, except it does not create a new DIR within the SAP system. The save function compares the links of the local CsapDir object with the links of the SAP DIR and updates the links of the SAP DIR within the SAP system.
Delete()	The delete function deletes the SAP DIR, identified by the CSapDir object, within the SAP system[3].
DeleteIndicator()	Property that represents the DeleteIndicator.
DirStatus()	Current status of a CSapDir. A CSapDir cannot have any new status, valid new status depends on the current status. The function GetStatusList returns a list with all valid status for this CsapDir object.
GetStatusList()	Returns a list of possible status.
GetStatusDescription()	Retrieves the description of the current status from the SAP system. The returned description depends on the chosen login language.
DirType()	Type of the CSapDir object. Valid types must be configured within the SAP system.
DirId()	Identifier of the CSapDir object.
DirPart()	Part number, two characters

---

[2] SAP generates automaticly the document number, if the special character "*" is used for the document number
[3] The SAP DIR will not be deleted direct. Instead a 'Delete Indicator' will be set.

DirVersion()	Version number, three characters
DirDescription()	Short description of the CSapDir.
Username()	Name of the user, who added the CSapDir to the SAP system
FnObjectIds()	Returns the FnObjectIds collection. This collection provides functions to add or remove FileNET objects.
MasterRecords()	Returns the MasterRecords collection. This collection provides functions to add or remove MasterRecord objects.

CMasterRecord

The CMasterRecord object is the representation of a SAP system master record. To link a master record object to a SAP DIR, two information are necessary, the master record type and the master record ID.

MasterRecordId()	Id of a master record. This ID can be a name or a number.
MasterType()	Name of the type of master record[4].
DeleteIndicator()	Identifies the linked master record for unlinking form the CSapDir and the SAP DIR

CMasterRecords

The CMasterRecords object is a collection of CMasterRecord objects.

Basic functionalty:

Collections implement the functions 'Add', 'Item', 'Count' and 'Remove'. Additionally the CMasterRecords collection provides the function 'clear()'.

Clear()	Removes all entries from the collection.

CDirStatus

The CDirStatus object represents a status of a SAP DIR. The CDirStatus object has two properties, the status and the status description.

Status()	Only two characters for the status can be used.
Statusdescription()	Description of the status. The description depends on the logon language.

CDirStati

The CDirStati object is a collection of CDirStatus objects. Function overview, see 'Basic functionality' of CMasterRecord.

Clear()	Removes all entries from the collection.

CFnObjectID

The CFnObjectID object has three properties to identify a FileNET document.

FnObjectId ()	The identifier of a FileNET object. This could be a normal ID or a

---

[4] Information about the different master record types are available within the SAP IMG system (Cross Applications -> Document Management System)

	SmartID[5].
ArchiveId ()	The archive ID defined in the SAP system, where the document is available
ArchiveObject ()	

CFnObjectIDs

The CFnObjectIDs object is a collection of CFnObjectID objects. Function overview, see 'Basic functionality' of CMasterRecord.

Clear()	Removes all entries from the collection.

DMSUIComponent.DLL

The DMSUIComponent.dll allows searching and displaying SAP material master objects. The responsible object for this functionality is the CDMS_MM_UIControler object. To access the SAP system the DMSUIComponent uses a CDMSConnection object. To make the connection object available to the DMSUIComponent.dll, the CDMS_MM_UIControler offers the function 'setConnection'.

CDMS_MM_UIControler

The CDMS_MM_UIControler is an examplary implementation of the DMSUIComponent interface.

DisplayMasterRecord ()	Display detailed information for a given SAP material master. The information will be displayed in an own dialog.
Parameter:	*Description*
MasterRecord	The name of the material master to display.

SearchForMasterRecord ()	Open a Dialog for searching and choosing a material master. The function returns the name of the selected material master object.

Enhanced Panagon Browse for R/3

Panagon Browse for R/3 can be configured to use a number of ActiveX Components implementing the DMS UI Component Interface. To enhance the existing Panagon Browse for R/3 the custom ActiveX has to be provided to the Panagon Browse for R/3. This can be done with the new Configuration Tool, under 'DMS UIComponents'. The information to add there, are the ClassID of the new component (Libraryname.Classname), the menu name for linking the new master record and a menu name for displaying the new master record. Additional, the type[6] of the new master record has to be added there.

Interface description

The interface for UIComponents defines three functions. One function for searching master records, one function for displaying information of a master record and the third function for setting a reference to a CDMSConnection object.

The function to set the reference to the CDMSConnection object is the 'setConnection' function. The parameter is the CDMSConnection object. With the CDMSConnection object, it is possible to access a SAP System. If the new custom UI component use a different object to access the SAP system, the

---

[5] In order to identify a FileNET object, a so-called smart id is used. The format of the smart id includes the object type, the identifier, and in case it is a document of a Content Services repository, the version of a document (e.g. DOC001100100, DOC001100100:01, FLD00150001 and STS001125225)
[6] Information about the different master record types are available within the SAP IMG system (Cross Applications -> Document Management System)

implementation of this function is not necessary. Nevertheless, to be compatible with all other UI components the 'setConnection' must exist on custom UI component. If the new UI component shall use the CDMSConnection object to access the SAP system, the implementation of the 'setConnection' function should include the following steps:

- Store the received reference to the CDMSConnection object, within a local variable

The name of the function to search for material master is 'searchForMasterRecord'. The steps to do within this function are:

- Logon to the SAP system, if necessary.
- Provide a GUI, where the User can add search criteria for the master record
- Show a GUI with the result of the search
- Return the ID of the selected master record

The function to display detailed information of a master record is the 'DisplayMasterRecord' function. The implementation for the method should include:

- Logon to the SAP system
- Reading the detail information for the given master record from the SAP system
- Show a GUI with the information of the master record

Add menu entries

To add menu entries no programming is needed. The only precondition for using the new UI component is that the new UI component is configured with the Configuration Tool correctly.

Programming Examples

The examples give a short overview, of how to use the delivered COM objects.

Establish a connection to a SAP System

Both the DMSComponent and the DMSUIComponent need a connection object to access to a SAP system. The object to logon to a SAP system is the oConnection object. After creating all objects, a connection to a SAP system is established with the logon function on the oConnection object. The reference to this object will be set to the oSapDMS object and to the oDMS_UI object. Both objects are now able to call rfc functions within the SAP system.

'Declare objects

Dim oConnection   As DMSConnection.CDMSConnection

Dim oSAPDMS   As DMSComponent.CSAPDMS

Dim oDMS_UI   As DMSUIComponent.CDMS_MM_UIController

'Create references to objects.

Set oConnection = New DMSConnection.CDMSConnection

Set oDMS_UI = New DMSUIComponent.CDMS_MM_UIController

Set oSAPDMS = New DMSComponent.CSAPDMS

'Logon to the SAP system with the CDMSConnection object.

Call oConnection.logon("martin","geheim","600","EN", _

                              "dehams86",0,true,"",false)

'Provide the connection object for all depending object.

Call oSAPDMS.setConnection(oConnection)

Call oDMS_UI.setConnection(oConnection)

…

Searching for Material Master and Display Information

The delivered DMSUIComponent.dll allows the user to search for SAP material master and to display detailed information of a SAP material master. Two responsible functions are the 'searchForMasterRecord' and the 'displayMasterRecord' function. The 'searchForMasterRecord' function opens a search dialog, where the user can search for master records object within the SAP system. After the user has found the master record and has selected it, the function returns the ID of the selected master record. To display information of a SAP master record the function 'displayMasterRecord' is called with the returned master record ID as parameter.

Dim sMaterialID   As String

…

'Call the search function on the DMSUIComponent object. Function returns the selected object. Precondition is that the oDMS_UI object, has a valid connection to a SAP system.

sMaterialID = oDMS_UI.searchForMasterRecord()

'Displays the content of the selected MasterRecord

Call oDMS_UI.displayMasterRecord(sMaterialID)

…

Connect a FileNET Object to an SAP Master Record

The objects which are stored in the CMasterRecord collection and the FnObjects collection of the CsapDir are linked to each other. Both collections are containers for special types of objects. The CMasterRecord collection is the container for CMasterRecord objects and the FnObjects collection is a container for CFnObjects. To link a master record object to a CSapDir, a CMasterRecord object has to be created and added to the master records collection of the CSapDir object. To link a FileNET object a CFnObject object has to be created and added to the FnObjects collection. To identify a SAP master record, the properties master record ID and the master record type must be set. To identify a FileNET object the property CFnObjectID must be set. A new CSapDir object can only be saved, if the DirStatus is specified. To submit the changes to the SAP system, the save function on the CSapDir object has to be called.

'Declare necessary objects

Dim oSAPDIR   As CSAPDIR

```
Dim oMasterRecord      As CMasterRecord
Dim oFnObjectId        As CFnObject


'Create all objects
Set oMasterRecord = New CMasterRecord
Set oSAPDIR = oSAPDMS.createDIR("DRW","*","000","00")⁷
Set oFnObjectId = New CFNObjectID


'Set properties for objects to link.
oMasterRecord.MasterRecordID = sMaterialID
oMasterRecord.MasterRecordType = sMasterType⁸
oFnobjectId.FnObjectId = sFnObjectId


'Add MasterRecord and FileNet object to link to DIR object.
Call oSAPDIR.MasterRecords.add(oMasterRecord)
Call oSAPDIR.FnObjectIds.Add(oFnobjectId)


'Set properties for DIR
oSAPDIR.DIRStatus = sInitailDirStatus⁹
oSAPDIR.DIRDescription = sDirDescripton⁷
…


'Save DIR. At this point the RFC functions are called to create the DIR within the SAP system and to link the
two objects to the DIR
Call oSAPDIR.saveAsNew
```

Change an existing Document Info Record within a SAP System

To change properties on an existing SAP DIR object, a local CSapDir object has to be created. The 'getDir' function on the CSapDMS object returns a full initialized CSapDir object, including all links and the main properties of the SAP DIR object. The 'save' function on a CSapDir object realizes all changes made on the local CSapDir object.

```
'Declare objects
Dim oExistingDIR       As CSAPDIR


'Create DIR object with information of an existing SAP DIR. Uses the DMSComponent.dll
Set oExistingDIR  = oSAPDMS.getDIR("DRW","10000000123", "000","00")
```

---

[7] The parameter to create a DIR object depends on the chosen DIR Type. In particular the sign for automatic Dir number creation differs.
[8] The MasterType is stored in the Configuration Program, under 'Dms->UIComponents'
[9] The data is stored under the 'Dms' configuration

'Set a new DIR Description and a new DIR Status

oExistingDIR.DIRDescription = "Top Secret"

oExistingDIR.Status = "SP"

'Save the changes of the DIR

Call oExistingDIR.save

Delete all Links between a FileNET Object and a Master Record

To delete all links between a FileNET object and master record, both collections (Master Records and FnObjects) have to be cleared. After that, the 'save' function realizes the changes within the SAP system. Finally the delete function sets the DeleteIndicator of the SAP DIR. The SAP DIR still exists within the SAP system. Furthermore, it is possible to access the SAP DIR object.

…

Call oExistingDIR.MasterRecords.Clear

Call oExistingDIR.FNObjectIds.Clear

Call oExistingDIR.save

Call oExistingDIR.delete

…

Creating a Custom Display Connector

IDM Desktop for R/3 uses Display Connectors to display FileNet objects, requested by the SAPGUI. It is now possible to configure custom Display Connectors, which are responsible for displaying these objects. The Display Connector has to be delivered as an ActiveX component. If the ActiveX component is registered successfully, use the Configuration Program of the IDM Desktop for R/3 for configuring the custom Display Connector. The IDM Desktop for R/3 expects two functions on the Display Connector object.

Interface Description for new Display Connectors

The installation of IDM Desktop for R/3 copies a class file to the destination path, which describes the interface for Display Connectors. There is also an Enum included, which is part of the display function.

DisplayObject ()	
This function opens the responsible program to display the FileNET object.	
Parameter:	*Description*
ObjectId	Id of the object to display
ObjectType	Object type of the document to display (DwFolder, DwDocument, DwLocalFile, etc.) All allowed types are defined in the Enum DwObjectType.
Libraryname	Optional parameter. Specifies the library where the object is stored.

CloseViewer ()
Closes the viewer.

LoginManager

IDM Desktop for R/3 provides an OLE Automation Server (ActiveX Object) to implement automatic background logoff from an IDM Services Library after a certain period of time. This object can be used by calling its methods to provide LoginManager in the custom View.

Function Overview

The ConnectionManager.dll contains the necessary object with the following methods:

SetUILogon ()	
Determines if the FileNet logon dialog is used or not.	
Parameter:	*Description*
bUILogon	IN – Use FileNet logon dialog (true use dialog – false does not use FileNet dialog)

LogonToLibrary ()	
Logon to a specific IDM Services Library	
Parameter:	*Description*
Library	IN - Reference to an IDMObjects.Library object
AutoLogoffPeriod	IN - Period in minutes. (Zero means AutoLogoff-Period is disabled.)

Note: If a connection to the library already exists, this existing connection will be used. If not, a new connection will be established by logon to the IDM Services Library. A logon to an existing connection resets the LoginManager timer.

LogoffFromLibrary ()	
Logoff from a specific IDM Services Library.	
Parameter:	*Description*
Library	IN – Reference to an IDMObjects.Library object

Note: The connection object (and the Autologoff timer) will be deleted.

LogoffFromAllLibraries
Logoff from all (open) IDM Services Libraries.

Note: All connection objects (and the LoginManager timer) will be deleted.

StopTimer()	
Stop the AutoLogoff timer for a specific IDM Services Library.	
Parameter:	*Description*
Library	IN – Reference to an IDMObjects.Library object

RestartTimer()

Restart the AutoLogoff timer for specific IDM Services Library.	
Parameter:	*Description*
Library	IN – Reference to an IDMObjects.Library object

Note: The AutoLogoffPeriod is set to the timerange specified during "Logon-ToLibrary()".

General Settings

Storing Settings in the Windows Registry

All custom components should be aware of several IDM Desktop for R/3-wide settings, e.g. logging, and may store its own settings. The following sections list the relevant keys, value names and values for someone writing a custom View.

These settings are stored in **\\HKEY_LOCAL_MACHINE\SOFTWARE\FileNET\Document Warehouse\Preferences** . The following table shows the relevant keys and their meanings:

Key	Value Name	Description
\SapSystems\<activeSapSystem\SapArchive\Archive ID>		Archive Mappings
	Library ID	Mapping of SAP Archive ID to IDM Library ID: For IS:<object>:<domain>:<or-ganization> For CS: <service>^<server>
cDesktop\Views\<name of View>		Views configured to be used by IDM Desktop for R/3 (usually entered through the Configuration Program
	Class	Program ID for View
	Title	Display Title for this View
	IsCustom	Indicates that the View is a custom View
\cDesktop		
	Autologon	Boolean for automatic logon
	Autologoff	Boolean for automatic logoff
	AutologoffTime	Period for automatic logoff

Only the keys and values under \Views\<your View name> are writeable, all other keys are considered as read-only.

Setting for DMS are avaiable within the following sections:

Key	Value Name	Description
\Dms\UIComponents\<name of new component>		Custom UIComponent
	MasterType	The master type of the new master record
	MenuDisplayCaption	Menu name within the Browse View to display the new Master Record
	MenuLinkCaption	Menu name within the Browse View to link a Master Record to a FileNet object

The setting for custom DisplayConnectors are in the section:

Key	Value Name	Description
\cDesktop		

Key	Value Name	Description
	DefaultDisplayConnector	Name of the DefaultDisplay-Connector (Libraryname.Classname)

Logging

All components can log their messages to the same log file as IDM Desktop for R/3 (see registry keys in section "Storing Settings in the Windows Registry" on page 26).The log level determines the type of information that should be logged. Here is the format of a log entry:

**mm/dd/yyyy hh:mm:ss <L> <ComponentName> <module.function>:
<message text>**

The following information shall be provided by the ComponentName:

- <L> = [e|w|i] (log level of message: error, warning, or info)
- <ComponentName> is the (short) name of the component (reserved names are Browse, Capture, IDMForR3 and Queue)
- <module.function> identifies the function were the message was logged
- <message text> is the log message itself