FileNet Document Warehouse for SAP

Programmer's Handbook

Release 5.1

March 2003

FileNet Corporation
3565 Harbor Boulevard
Costa Mesa, California 92626
800.FILENET (345.3638)
Outside the U.S., call:
1.7 1 4 .3 2 7 .3 4 0 0
www.filenet.com

Notices

FileNet has made every effort to keep the information in this document 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. In no event will FileNet be liable for direct, indirect, special incidental, or consequential damages resulting from any defect in the documentation, even if advised of the possibility of such dam-ages. No FileNet agent, dealer, or employee is authorized to make any modification, extension, or addition to the above statements.

Contents

About this Manual

The Document Warehouse for SAP (DWSAP) Programmer's Handbook provides information for extending the existing product by writing and integrating new components. It is also possible to create new applications using the Document Management System (DMS) object model, which is part of DWSAP 5.1.

Skill Level Requirements

The programmer will need a good understanding of the FileNet and SAP systems, as well as a working knowledge of Windows XP/2000 and the preferred development environment. Specifically, the programmer will need to understand the following:

- SAP R/3 3.1, 4.0, 4.5 or 4.6 systems.
- Windows XP/2000 procedures.
- The development environment (this manual uses Microsoft Visual Basic 6.0 and ASP).
- Microsoft's ActiveX architecture.
- Programming with FileNet IDM Objects.
- Object oriented Design and programming concepts.

Related Publications

The following FileNet manuals and SAP documentation will prove useful in software installation, operation, and administration:

- DWSAP Installation & Configuration Manual.
- FileNet IDM Desktop online documentation.
- 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 kinds 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

Bold Type Indicates menu names, menu items, component names

Blue text *Indicates a link to another topic, a link to another section in the same topic, or a link to an external topic.*

`Code: Indicates code samples, syntax, class names, and parameters`

Education

FileNet provides various forms of instruction. Please visit the Global Learning Services in FileNet's Service & Support area at www.filenet.com.

Comments and Suggestions

FileNet invites all customers to communicate with the Documentation group on any question or comment related to FileNet manuals and online help. Send email to docs@filenet.com. We will make every effort to respond within one week. Your suggestions help us improve the products we deliver.

1

Document Warehouse for SAP Release 5.1

This chapter describes the following topics related to Document Warehouse for SAP (DWSAP):

- Overview of Client Components and their architecture.
- Concept of a Custom View.
- Use of DMS Components.

DWSAP Overview

FileNet has integrated its product suite with SAP system, using SAP ArchiveLink interface. The result is an application, FileNet DWSAP. DWSAP enables business applications of SAP to process FileNet documents. It also allows document and data archiving. DWSAP consists of:

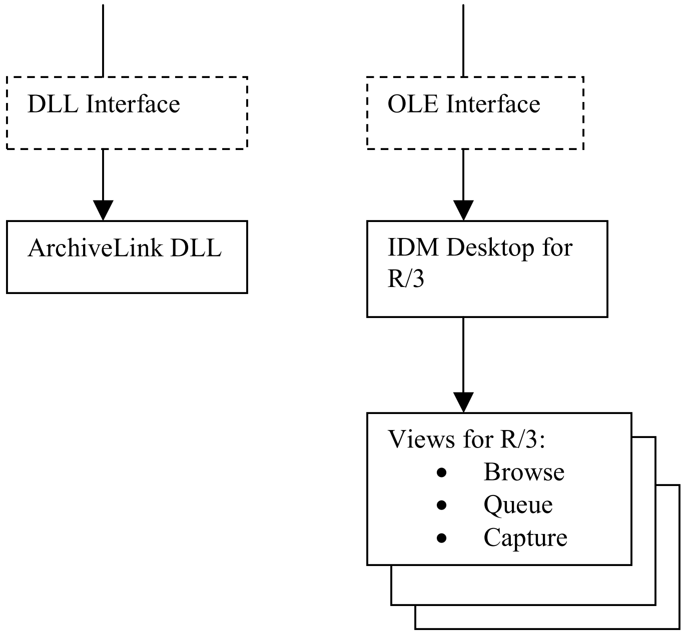
- **IDM Services for R/3**: Refer to server components that run on Windows 2000 server platform.
- **IDM Desktop for R/3**: Refers to client components that run on the Windows platforms, such as Windows 2000 and Windows XP.

System Architecture of IDM Desktop for R/3

The interfaces that IDM Desktop for R/3 supports to SAPGUI and Archive Link software of SAP are:

- DLL-Interface: This interface is used by ArchiveLink Viewer to access the documents, which is implemented in ArchiveLink DLL.
- OLE-Interface: This interface is used by SAPGUI to display documents in the IDM Viewer and linking them with SAP transactions. IDM Desktop for R/3 implements it. To obtain a document for linking, the interface uses Views.

The figure below shows the different components of IDM Desktop for R/3:



Linking a Document

Linking a document 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 FileNet Image Services (IS) or Content Services (CS). The document id is used to display the document later. Prior to linking a document to SAP transaction, the user opens IDM Desktop for R/3, selects the document to be assigned, displays it in FileNet IDM Viewer, switches to SAPGUI, and links the document.

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

- FileNet Browse for R/3: Allows the user to browse through the folders of IS or CS, with a Microsoft Explorer-style user interface.
- FileNet Queue for R/3: Allows the user to access the contents of a FileNet WorkFlo queue, which holds documents that are ready for processing in SAP system.

These Views provide flexible ways of finding or creating documents for linking.

DWSAP enables the user to write a Custom View, which IDM Desktop for R/3 can use along with the standard Views. This Programmer's Handbook discusses the concepts involved in writing an own View and lists the interfaces, methods, and parameters the Custom View supports.

Linking SAP Master Objects

The Document Management System (DMS) interface provided by SAP system connects external documents, folders, and objects from FileNet repositories, with master data within SAP system. Master data refers to a material master, an equipment master, or a customer master. Different dynamic link libraries delivered with DWSAP release 5.1 provide this functionality. The sample applications of DWSAP 5.1 to illustrate the use of DMS objects are:

- Thick Client: IDM Desktop for R/3 is delivered with an out-of-the-box application to link FileNet objects with SAP material master records. This functionality is available through DMS menu integrated with the view FileNet browse for R/3.
- Thin Client: DMS sample Web application is delivered with DWSAP 5.1 server components. It enables to link FileNet objects with SAP material master. The DMS components are packaged as a part of server Add-on. The sample application can be accessed through a URL.

2
Views

A View is an ActiveX control that allows selecting a document and also provides functions for linking the selected document to SAP transaction. Document Warehouse for SAP (DWSAP) allows writing 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.

The exact functionality expected of a View 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 meet some behavior requirements. It has to know about some settings stored in the registry for IDM Desktop for R/3. It can also use the components provided with IDM Desktop for R/3 to logon, or logoff from the FileNet libraries. In the following sections, this has been explained.

Startup, Shutdown and Save Settings

The methods that should be exported by a View are listed in the table below:

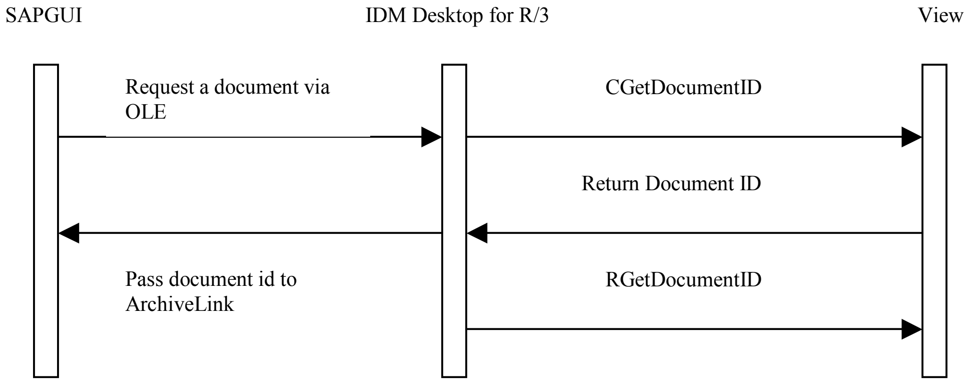
Function Name	Description
StartUp ()	Is called after instantiating View in the MDI child windows OLE Container. StartUp may also be called while re-initializing the view.
ShutDown ()	Is called before destroying the instance of the View.
SaveSettings ()	Is called explicitly by IDM Desktop for R/3, when the user requests to save the settings, for example through menu item, or while closing an application. A View should not save its settings in method ShutDown.

Providing a Document ID

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

When SAPGUI requests a document Id, IDM Desktop for R/3 passes this request to the currently active View. The currently active View is the one with the focus. Calling the `CgetDocumentId` function does this. The

View returns its current document or folder id and the type of object. If IDM Desktop for R/3 is able to pass this id to SAPGUI, it calls the `RGetDocumentId` function. This function informs the View that the id has been processed successfully. The following figure shows the interaction between IDM Desktop for R/3 and a View.



Therefore, the `CGetDocumentId` function is called to get an object id, but the object id is not passed to SAPGUI until the `RGetDocumentId` function is called. For example, FileNet Queue for R/3 returns the document id of its currently selected queue entry. If the document id has been passed to SAPGUI successfully, the queue entry must be deleted from the queue. If it fails to pass the document id to SAPGUI, the queue entry should not be deleted. Hence, the deletion must occur in the function `RGetDocumentId`.

CGetDocumentId

The following are the parameters for the function CGetDocumentId:

- `SLibraryId`: refers to the Library that the document or folder has to belong to.
- `LRetObjId`: refers to the document or folder ID.
- `lRetObjType`: refers to the object type idmObjTypeDocument (=1) or idmObjTypeFolder (=2).
- `bRetSuccess`: refers to the success code returned(0 = OK,1 = no Object Id returned).

In order to pass the Id and type of currently selected object to IDM Desktop for R/3, the following steps need to be followed in function `CGetDocumentId`:

1. Check whether the View has a current valid object id to be passed on to SAPGUI. If it does not have, inform the user by either sending an

error message, or by giving an option to switch to Advanced Interaction. For more information, see Advanced Interaction.

2. If the View has an object id, check whether the object belongs to the currently selected library.
3. Remember the id as a reference for `RGetDocumentId` function, and return the id along with the object type. If the library does not match, inform the user and switch to Advanced Interaction.
4. On all further calls to the `CGetDocumentId` function, check whether View is awaiting a call to the `RGetDocumentId` function.

Caution: - The functions `CGetDocumentId` and `RGetDocumentId` are always called in pairs. This means that two succeeding calls to `CGetDocumentId` without a call to `RGetDocumentId` in between are not allowed.

RGetDocumentId

The following are the parameters for the function `RGetDocumentId`:

- `vObjId`: refers to unchanged `vRetObjId` of `CgetDocumentId`.
- `bSuccess`: refers to the status of the request (returns True if object is succesfully passed to SAPGUI and False if object id is not succesfully passed to SAPGUI).

The following steps need to be followed in function `RGetDocumentId`:

1. Check whether View actually awaits a call to `RGetDocumentId` function.
2. If not, ignore the call. Otherwise, use the incoming object id as a reference to the preceding call to `CGetDocumentId` function.
3. Process the current object according to the behavior of View. For example, on success, FileNet Queue for R/3 deletes the locked queue entry and fetches the next entry. On failure, it unlocks the queue entry and informs the user.

Advanced Interaction

There are cases where a request has to be withheld so that it can be processed, or cancelled later. Advanced Interaction takes care of such cases. The following examples can be categorized into such cases:

- When a user tries to assign a document to an SAP object and the currently active View cannot provide a document id, SAPGUI displays a message that no documents were scanned in. View may not be able to provide a document id because no document had been selected.

- It is also possible that the user selects an object, but SAPGUI requests a document id from a different library. In this case, the user should be informed about it, and the user should be able to make a new selection. The document id of the newly selected document should be returned to SAPGUI.

The key to Advanced Interaction is the `CGetDocumentId` function that does not return until the user selects a valid object, such as a document, or explicitly cancels the assignment. The advanced interaction can be achieved by providing the following menu items:

- Continue assignment: allows the user to reselect the FileNet object and continue processing.
- Cancel assignment: allows the user to cancel the processing for current request.

The behavior that standard Views implement in the `CGetDocumentId` function is shown in the listing below:

```
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 GUI Guidelines

The following guidelines should be considered, while designing a Custom View:

- A View should have a menu and a status bar.
- View should be negotiated to the middle position because left and right are reserved for the MDI window.
- The MDI window and the MDI child window do not provide a status bar or any other visual means to display information.
- When the MDI child window and the OLE Container are resized, the View should adapt to the new size.

3
DMS

Introduction

Each SAP R/3 system comes with a built-in Document Management System (DMS). SAP R/3 DMS organizes documents, and connects document files from external repositories to objects within SAP R/3. The base unit of DMS is a Document Info Record (DIR). DIR is a Meta record that provides information about a linked document. DIR stores all the business information for a document. It can be linked to an SAP R/3 master record, as well as an external document file. The external file can be a file in the file system or an object in an optical archive, available through the ArchiveLink interface. A DIR acts as a connector between SAP R/3 master data and external document files. DWSAP provides DMS components for the following DMS functionality:

- Create and modify DIRs in SAP R/3.
- Link objects from FileNet repositories such as document, specific version of a document, folder, and stored search, to DIRs.
- Access information of SAP R/3 master records.
- Functions to search and select SAP R/3 master records.

DMS components expose DMS object model based on Component Object Model (COM) technology and COM objects for providing the DMS functionality. These objects can be instantiated in thick client and thin client environment. To call functions and objects within SAP R/3, DMS components use SAP BAPI interface.

Caution: The SAP BAPI interface for DMS functions is available with release 4.5 of SAP system. BAPI interface is the officially supported interface for further releases of SAP.

This chapter provides information about the DMS object model. The programming examples in this manual use the syntax of Visual Basic 6 and Active Server Pages (ASP) because the object model is based on Microsoft COM technology.

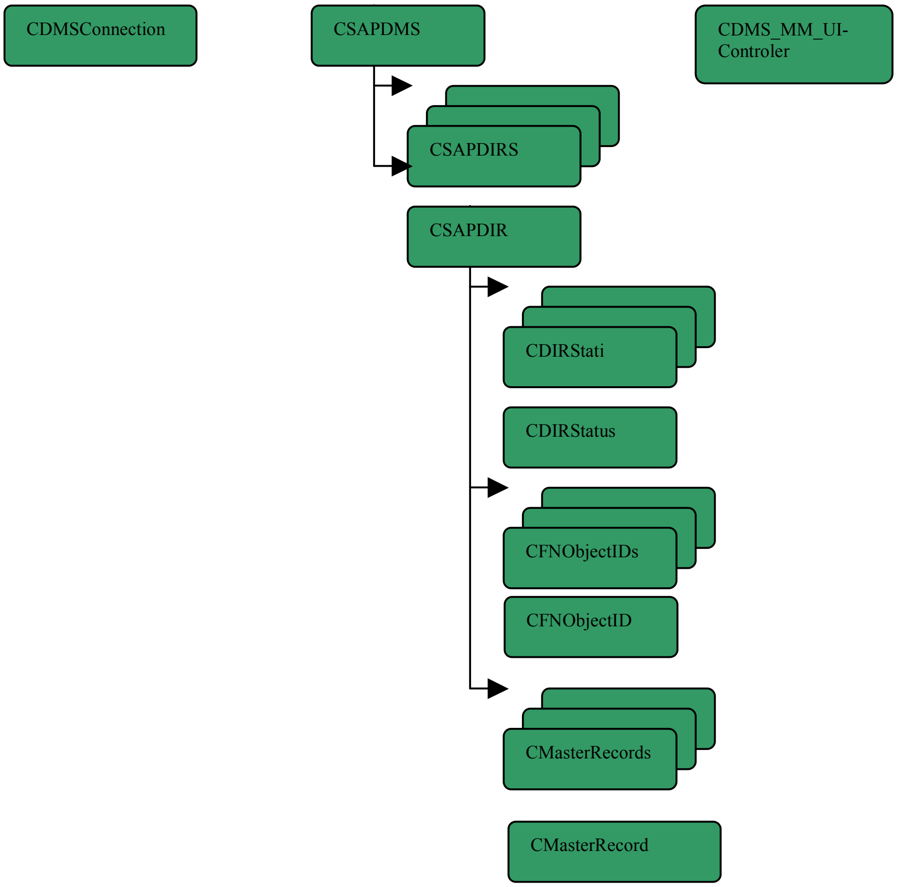
Caution: DMS functionality based on CAD interface is no longer supported by SAP. COM objects use BAPI interface to call DMS functions. This means that SAP system must be of release 4.5 or higher. SAP releases 4.0 or less are not supported for DMS functions.

DMS Components

The DMS Components are delivered in the form of three ActiveX dynamic link libraries that provide DMS COM objects. These libraries, installed with DWSAP Client and Server add-on software, are:

- DMSConnection.dll.
- DMSComponent.dll.
- DMSUIComponent.dll.

DMS Components require additional SAP ActiveX controls such as wdtfuncs.ocx, wdtaocx.ocx, and wdtlog.ocx, and another dynamic link library for RFC communication, librfc32.dll. The following figure gives an overview of COM objects that enable to create applications to connect different types of SAP master records such as material master, equipment master or customer master with FileNet objects. With DWSAP 5.1, these DLLs can be instantiated in a thick client and a thin client environment.



DMSConnection.dll

The DMSConnection.dll provides an object to connect to a SAP system. The `CDMSConnection` object enables to logon and logoff from SAP system.

- `IsConnected`: This function returns the information about connection to SAP system. It returns true if a connection to SAP system exists and false otherwise.
- Logon: Function to log on to an SAP system. The following are the parameters for the function `Logon`:
    - `User`: refers to the SAP user Id
    - `Password`: refers to the password for the user
    - `Client`: refers to the SAP Client
    - `Language`: refers to the language supported by SAP

  - ApplicationServer: refers to the IP address or alias name of SAP server
  - SapSystemNumber: refers to SAP system number
  - RfcTrace: represents the option to switch on/off SAP trace file
  - RouterString: indicates the router string to a SAP server
  - RouterString: indicates the router string to a SAP server

- Logoff: Function to logoff from an SAP System.
- GetConnection: This function returns a SAPConnection object. This object holds properties about the current SAPConnection. This object can be used in different controls provided by SAP.

  The CDMSConnectionPool object implements connection pool in a web environment. If a connection with the same parameters already exists; this object returns the same connection object to the user and a new connection is not established. This object uses the CDMSConnection object to logon/logoff from SAP system. In order to keep track of the connection established by a user in a session, this object should be used in the web pages with a session scope.
- GetDMSConnection: Checks for the availability of a connection to SAP system. The GetDMSConnection function returns CDMSConnection object. If a connection with the supplied parameters exists, then this function returns the same connection; else, it establishes a new connection with the parameters provided. The following are the parameters for the function GetDMSConnection:
  - User: refers to the SAP user Id.
  - Password: refers to the password for the user.
  - Client: refers to the SAP Client.
  - ApplicationServer: refers to the IP address or alias name of SAP server.
  - SapSystemNumber: refers to SAP system number.
  - Language: refers to the language supported by SAP.

DMSComponent.dll

DMSComponent.dll offers a set of objects to link FileNet objects with SAP master record objects. The central object of DMSComponent.dll is CSAPDMS object.DMSConnection.dll needs a reference to the CDMSConnection object. This reference can be set with the setConnection function of the CSAPDMS object. CSAPDMS allows creating a CSAPDIR object, with the functions CreateDIR and GetDIR.

SAP Document Info Record (DIR) is the link object between master record and FileNet objects. The object in DMSComponent.dll corresponding to SAP DIR is the CSAPDIR object. CFnObjectId and CMasterRecord objects are available for FileNet and SAP master record objects respectively. Additionally, for all these objects, collection objects exist which are CSAPDIRs, CFnObjectIds, and CmasterRecords.

CSAPDMS

CSAPDMS object offers functions to create CSAPDIR object and retrieve data from an existing SAP DIR object. To access the SAP system, DMSComponent.dll needs a reference to a CDMSConnection object, which delivers the necessary logon information. This reference can be set with the setConnection function of the CSAPDMS object. The getDIRs function returns a collection with all CSAPDIR objects that are linked to a FileNet object. The following functions are provided for CSAPDMS object :

- SetConnection: Provides a reference to CDMSConnection object used by DMSConnection.dll. This function should be called before using the CSAPDMS object.
    - CDMSConnection: refers to the connection object, which holds a connection with the SAP system.
- CreateDIR: Creates a CSAPDIR object. This function should be used to create a CSAPDIR object without any corresponding DIR element in the SAP system. The following are the parameters for the function CreateDIR:
    - DocumentType: refers to the type of DIR to be created in SAP system.
    - DocmentNumber: represents the document number.
    - DocumentPart: represents the DIR Part number.
    - DocumentVersion: represents the DIR Version number.

Note: Different DIR types are defined in SAP IMG under **Cross Applications → Document Management System**. If * is used for **DocmentNumber** parameter, SAP automatically generates the document number for the DIR.

- GetDIR: Creates a CSAPDIR object of an existing DIR object with information such as existing links to FileNet objects and master record. The following are the parameters for the function GetDIR:
    - DocumentType: refers to the type of DIR to be created in SAP system.

  - `DocmentNumber`: represents the document number.
  - `DocumentPart`: represents the DIR Part number.
  - `DocumentVersion`: represents the DIR Version number.

- `GetDIRs`: Retrieves the DIR objects linked to a FileNet object. Using information of the retrieved DIR, a collection of DIR objects is created. The following are the parameters for the function `GetDIRs:`
  - `FnObjectID`: represents the Id of the FileNet object linked to DIRs.

CSAPDIR

The `CSAPDIR` object represents an SAP DIR object. The primary key of an SAP DIR object is a combination of the DIR type, DIR number, the DIR part, and DIR version. SAP DIR object is used as a link between a master record and FileNet objects. The following functions are provided for `CSAPDIR` object.

- `SaveAsNew`: Creates a new SAP DIR object in the SAP system with the information provided with `CSAPDIR` object. This means that the function can be called on `CSAPDIR` object without any corresponding SAP DIR objects. If either `MasterRecords` or `CFnObjects` properties are given, links to the SAP DIR object are created.
- `Save`: Saves the properties of the local `CSAPDIR` object on a DIR existing in the SAP system. This function is similar to `SaveAsNew` function except that it does not create a new DIR within SAP system but modifies the properties of a DIR existing in the SAP system. The `save` function compares links of the local `CSAPDIR` object with those of SAP DIR, and updates the links of SAP DIR within the SAP system.
- `Delete`: Deletes the SAP DIR, identified by `CSAPDIR` object, within the SAP system.

Note: The SAP DIR will not be deleted direct. Instead a ‘Delete Indicator’ will be set.

- `GetStatusList`: Returns a list of status for a DIR.
- `GetStatusDescription`: Retrieves the description of the current status from SAP system. The returned description depends on the chosen login language.
- `SaveAsNew`: Creates a SAP DIR object with the information of `CSAPDIR` object within SAP system. This means that the function can only be called on `CSAPDIR` object, without any corresponding DIR objects in SAP. If either the `MasterRecords` or `CFnObjects` properties, or both are given, links to the SAP DIR object will be created.

- `SetConnection`: Provides a reference to `CDMSConnection` object used by the DMSConnection.dll. This function should be called before using the `CSAPDIR` object. The following are the parameters for the function `SetConnection`:
    - `CDMSConnection`: refers to the connection object, which holds a connection with the SAP system.

The following are the properties provided on the `CSAPDIR` object:

- `DeleteIndicator`: represents the Delete Indicator for a DIR in SAP.
- `DirStatus`: represents the current status of a SAP DIR. The new status of a DIR depends the value of the current status of the DIR.
- `DirType`: represents the type of SAP DIR. The valid types for SAP DIR must be configured in the SAP system.
- `DirId`: represents the ID of a SAP DIR.
- `DirPart`: refers to the part number of SAP DIR. It is two characters long.
- `DirVersion`: refers to the version number of a SAP DIR. It is three characters long.
- `DirDescription`: refers to the short description of SAP DIR.
- `Username`: represents the name of SAP user responsible for creating SAP DIR in SAP system.
- `FnObjectIds`: represents the Ids of FileNet objects linked to the SAP DIR. It returns the collection of `FnObjectIds` that contains the functions to add or remove FileNet objects.
- `MasterRecords`: represents the Ids of SAP Master records linked to SAP DIR. It returns the collection of `MasterRecords` that contains the functions to add or remove FileNet objects.

CMasterRecord

`CMasterRecord` object represents the SAP system master record. To link a master record object to a SAP DIR, the master record type and master record ID parameters are necessary. The following are the properties provided on the `CMasterRecord` object:

- `MasterRecordId`: represents the Id of an SAP master record. This Id can be a name or a number.
- `MasterType`: represents the type of SAP master record.

- DeleteIndicator: specifies whether the SAP DIR can be unlinked from the master record linked to it.

Note: Information about the different master record types is available within SAP IMG system **Cross Applications → Document Management System**.

CMasterRecords

CMasterRecords object represents a collection of CMasterRecord objects. It implements functions such as, Add, Item, Count, and Remove that are supported by collections. The CMasterRecords object also provides the function clear (), to remove all the entries from the collection.

CDirStatus

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

- Status: represents the status of a DIR in SAP. The status can have only two characters such as FR, AA etc. These statuses are defined in SAP.
- Description: represents the description of a status of a DIR.

CDIRStati

CDIRStati object refers to a collection of CDirStatus objects. In addition to implementing the basic functions supported by collections i.e. 'Add', 'Item', 'Count' and 'Remove', the CDIRStati object provides the function 'clear ()'.

- Clear: Removes all entries from the CDIRStati collection.

CFnObjectID

CFnObjectID object identifies a FileNet object. It has the following three properties:

- FnObjectId: identifies the Id of the FileNet object. It can be a conventional id or a smart id.
- ArchiveId: represents the archive Id defined in a SAP system where the document is available.
- ArchiveObject: represents the Archive Object defined in SAP system.

CFnObjectIDs

CFnObjectIDs object refers to a collection of CFnObjectID objects. In addition to implementing the basic functions supported by collections i.e. 'Add', 'Item', 'Count' and 'Remove', the CFnObjectIDs object provides the function 'clear ()'.

- Clear: Removes all entries from the CFnObjectIds collection.

DMSUIComponent.dll

DMSUIComponent.dll is a component that allows searching and displaying SAP material master objects. The object responsible for this is the CDMS_MM_UIControler object. To access the SAP system the DMSUIComponent uses a CDMSConnection object. To access the SAP system, DMSUIComponent.dll needs a reference to a CDMSConnection object, which delivers the necessary logon information. This reference can be set with the setConnection function of the CDMS_MM_UIControler object.

CDMS_MM_UIControler

CDMS_MM_UIControler object searches for, and displays the material master records in SAP. This object displays the data in its own dialog box, which means that this object cannot be instantiated in a Web-environment. Additional objects have been exposed to achieve this functionality in a Web page.

- DisplayMasterRecord: Displays information for a given SAP material master. It opens a dialog box to display the details of the material master records linked to the selected FileNet object. The following parameters need to be passed to the function DisplayMasterRecord:
    - MasterRecord: represents the Id of material master for which information is to be displayed.
- SearchForMasterRecord: Opens a dialog box to search and choose a material master. The function returns the name of the selected material master object.

Since, CDMS_MM_UIControler object relies on a dialogs to display the user interface to search and display material master records, the above functions cannot be used in a Web environment using ASP pages. Therefore, DMSUIComponent.dll has been modified and objects, CSearchManager and CDisplayManager, have been exposed for instantiation in a Web environment.

CMaterial

CMaterial object represents SAP system material. The properties for this object are:

- MaterialId: represents the material id in SAP.
- MaterialDescription: represents the description for material id.
- Division: represents the division to which the material belongs.
- Office: represents the office to which the material belongs.
- MaterialGroup: represents the group for the material in SAP.
- ProductHierarchie: represents the product hierarchy for the material.
- UnitOfMeasure: represents the unit of measure for the material.
- OldMaterialNumber: represents the old material number for the material.

CMaterials

CMaterials object refers to a collection of CMaterial objects. In addition to implementing the basic functions supported by collections i.e. 'Add', 'Item', 'Count' and 'Remove', the CMaterials object provides the function 'clear ()'.

- Clear: Removes all entries from the CMaterials collection.

CsearchManager

CsearchManager object searches a material master data in SAP, and returns a collection of material master records. CsearchManager object does not use any dialog boxes to present the user interface to search a material master record. This means that it can be instantiated in a Web environment. The only limitation to this approach is that the user needs to write a separate ASP page to implement the user interface for search. Web page SearchForMaterialMaster.asp in DMS web application makes use of this object. The following functions are provided for CSearchManager:

- getMaterialMasters: Searches for the material master, and returns the result as CMaterials object. The following are the parameters for the function getMaterialMasters:
  - sMaterialId: represents the material Id which needs to be searched for in SAP.
  - sMaterialDesc: represents the description for material as a search criteria for searching for a material.
- setConnection: CSearchManager object needs a reference to a connection to SAP system that holds the necessary logon information.

This function provides a reference to the `CDMSConnection` object used by DMSConnection.dll. This function should be called before using the `CSearchManager` object. The following are the parameters for function `SetConnection`:

  - `CDMSConnection`: refers to the connection object, which holds a connection with the SAP system.

CDisplayManager

`CDisplayManager` object searches the material master data in SAP and returns it as a CMaterial object. Similar to the `CsearchManager` object, the `CDisplayManager` object can also be instantiated in a Web environment. Data returned by `CDisplayManager` class can be displayed using HTML. The `CDisplayManager` object has the same limitation as that of `CsearchManager` objects i.e. a separate ASP pages needs to be written to implement the user interface. Web page DislpayMaterialMaster.asp in DMS web application makes use of this object. The following functions are provided for `CDisplayManager`:

- getMaterialMaster: Searches for the material master and returns the result as CMaterial object. The following are the parameters for the function `getMaterialMaster`:
  - sMaterial: represents the material id for which data needs to be displayed.
- SetConnection: `CDisplayManager` object needs a reference to a connection to SAP system that holds the necessary logon information. This function provides a reference to the `CDMSConnection` object provided by DMSConnection.dll. This function should be called before using the `CDisplayManager` object. The following are the parameters for function `SetConnection`:
  - `CDMSConnection`: refers to the connection object, which holds a connection with the SAP system.

Collaborating DMS Components

The COM objects work together, although they are packed in different dynamic link libraries. Some functionality of the COM objects depends on the existence of other objects. For example, since a connection to an SAP system is needed for connecting FileNet objects to master records, a connection has to be established with an SAP system before a link can be established between a FileNet object and a master record. This can be done using `CDMSConnection` object. Therefore, if a connection has been established with the `login` function of the `CDMSConnection` object, a `CSAPDMS` object can be created. This connection can be passed on to `CSAPDMS` using the function `SetConnection`.

CSAPDMS object is the root object for operations with Document Info Records (DIRs) within SAP system. The CSAPDMS object provides functions for creating DIR objects. DIR objects refer to a representation of DIRs within the SAP system. Other methods such as GetDIRs return a collection of DIR objects that are linked to a FileNet object.

The CDMS_MM_UIControler object offers methods for searching for material masters within the SAP. Information of material masters can be shown in a Graphical User Interface (GUI). This object is used for searching for and displaying material master records in SAP and uses the connection provided by the CDMSConnection object. CSearchManager and CDisplayManager objects search and display material and master records, respectively, in a Web based environment. They are used in the DMS sample web application.

DMS Components and Thick-client (Enhanced FileNet Browse for R/3)

FileNet Browse for R/3 can be configured to use a number of ActiveX Components that implement the DMS UI Component Interface. To enhance the existing FileNet Browse for R/3, the Custom ActiveX should be provided to the FileNet Browse for R/3. This can be done with the Configuration Tool, under DMS UIComponents. The following information needs to be added for configuring Custom UI Components:

- ClassID of the new component (Libraryname.Classname).
- Menu name for linking the new master record.
- Menu name for displaying the new master record.
- Type of the new master record has to be added there.

The interface for UIComponents defines the following three functions:

- To search master records.
- To display information of a master record.
- To set a reference to a CDMSConnection object.

The setConnection function sets reference to the CDMSConnection object. SAP System can be accessed using the CDMSConnection object. If the new Custom UI component uses an object different from CDMSConnection to access SAP system, the implementation of this function is not necessary. To be compatible with all other UI components the setConnection should exist on the Custom UI component. If the new UI component uses the CDMSConnection object to access the SAP system, the implementation of the setConnection function should store the received reference to the CDMSConnection object within a local variable.

The searchForMasterRecord function searches the material master. The steps to perform within this function are:

a. Logon to the SAP system, if necessary.

b. Provide a GUI, where the User can add search criteria for the master record.

c. Show a GUI with the result of the search.

d. Return the ID of the selected master record.

The `DisplayMasterRecord` function displays detailed information of a master record. The implementation for the method should include:

a. Logon to the SAP system.

b. Read the detailed information for the given master record from the SAP system.

c. Show a GUI with the information of the master record.

No programming is required to add menu entries for new UI components. The new UI component should be correctly configured with the configuration tool to use the UI component efficiently.

DMS Components and Thin-client (DMS sample web application)

DMS components delivered with DWSAP 5.1 can be instantiated in a Web environment. To illustrate this, a sample DMS Web application has been developed using ASP. Using this application, multiple users can simultaneously link the FileNet objects to SAP DIRs.

With the DMS sample application, the user can:

- Browse the contents of FileNet Libraries and select the FileNet objects that need to be linked to DIRs.
- Link FileNet objects to DIRs.
- Create new versions of DIRs.
- Unlink DIRs from the FileNet objects.
- Link FileNet objects to Material Master.
- Displaying Material Master Records linked to FileNet object.

Constraints and Considerations

- Seamless integration of UI components with the ASP application is not possible because UI Components make use of dialogs that cannot be instantiated in a Web environment. The DMS sample web application does not detect the Custom UI components. The responsibility of integrating Custom UI components with the sample DMS web application for lies with the customers.
- DMS sample web application has been built on top of FileNet IDM web services (IDMWS). It makes use of standard web pages of IDMWS to browse through the FileNet libraries and allows selecting FileNet objects that need to be linked.

Programming Examples

The following examples explain how to use the delivered COM objects with Visual Basic and ASP. For more information on how DMS components can be instantiated in a Web environment, refer to ASP pages of DMS Sample Web application that are distributed with DWSAP 5.1.

Establishing a connection with SAP System

Both `DMSComponent` and `DMSUIComponent` objects need a connection object to access SAP system. . Subsequent to creating all objects, connection to an SAP system is established using the logon function of the `CDMSConnection` object. The reference to this object will be set to the `CSAPDMS` object and to the `CDMS_MM_UIControler` object. Both objects are now able to call RFC functions within the SAP system.

In a Web environment, the connection is established using the `CDMSConnectionPool` object. Connection Pool should be stored with session scope. Therefore, the same connection pool is available for a user session and connection can be provided to the user from this pool. The connection can be taken from the connection pool with the `getDMSConnection` function.

‘Declare objects

```
Dim oConnection          As DMSConnection.CDMSConnection

Dim oSAPDMS              As DMSComponent.CSAPDMS

Dim oDMS_UI              As
DMSUIComponent.CDMS_MM_UIControler
```

‘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)
```

…

‘Declare objects

```
Dim    oConnection,    oSAPDMS,    oSearchManager,
oDisplayManager, oDMSConnectionPool
```

```
'Create references to objects.

Set                    oConnection                    =
Server.CreateObject("DMSConnection.CDMSConnection")

Set                  oConnectionPool                  =
Server.CreateObject("DMSConnection.CDMSConnectionPool"
)

Set                       oSAPDMS                     =
Server.CreateObject("DMSComponent.CSAPDMS")


Set                   oSearchManager                  =
Server.CreateObject("DMSUIComponent.CSearchManager")


Set                  oDisplayManager                  =
Server.CreateObject("DMSUIComponent.CDisplayManager")
```

'Get a connection to SAP system with the CDMSConnectionPool object.

```
Set      oConnection       =       oDmsConnectionPool
getDMSConnection("basis","basis","800",_
"dehams86",0,"EN")
```

'Provide the connection object for all depending object.

```
Call oSAPDMS.setConnection(oConnection)

Call oSearchManager.setConnection(oConnection)

Call oDisplayManager.setConnection(oConnection)

…
```

Searching for Material Master and Display Information

The `searchForMasterRecord` and the `displayMasterRecord` functions search and display the SAP material master. The `searchForMasterRecord` function opens a search dialog box, 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)
…
Dim sMaterialID
Dim oCMaterials, oCMaterial


'Create reference to objects
Set                    oCMaterials                    =
Server.CreateObject("DMSUIComponent.CMaterials")
…
```

'Call the `getMaterialMasters` function of the

`CSearchManager` object returns the result of search as `CMaterials` object, which is a collection of `CMaterial` objects. Each member of the CMaterials collection, which is a CMaterial object, represents the details of the material. Unlike VB, the details of the material need to be displayed using HTML. Material Id and material description are passed as parameters to the function.

```
Set  oCMaterials  =  oSearchManager.getMaterialMasters
(CStr(Request.Form("TxtMaterialId")),CStr(
Request.Form("TxtDescription")))
…
```

Browse through the collection of `CMaterial` objects, and display each `CMaterial` member of this collection using HTML.

Similarly, to display linked material master records, call the function `getMaterialMaster` of `CDisplayManager` object. This function returns the result as `CMaterial` object. The details of the material contained in this object can be displayed in an ASP page, using HTML.

```
Set oCMaterial = oDisplayManager.getMaterialMaster("M-
03")
```

Connecting a FileNet Object to SAP Master Record

The objects stored in the `CMasterRecord` collection and the FnObjects collection of the `CSAPDIR` object, are linked to each other. 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 should be created and added to the master records collection of the `CSAPDIR` object. To link a FileNet object a `CFnObject` object should 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”)
```

Note: The parameter to create a DIR object depends on the chosen DIR Type. The sign for automatic DIR number creation differs.

```
Set oFnObjectId = New CFNObjectID
```

‘Set properties for objects to link.

```
oMasterRecord.MasterRecordID = sMaterialID
oMasterRecord.MasterRecordType = sMasterType
```

Note: The MasterType is stored in the Configuration Program, under DMS→UIComponents

```
oFnobjectId.FnObjectId = sFnObjectId
```

‘Add MasterRecord and FileNet object to link to DIR object.

```
Call oSAPDIR.MasterRecords.add(oMasterRecord)
Call oSAPDIR.FnObjectIds.Add(oFnobjectId)
```

‘Set connection for SAP DIR

```
Call oSAPDIR.setConnection(oConnection)
```

‘Set properties for DIR

```
oSAPDIR.DIRStatus = sInitailDirStatus
```

Note: The initial status for DIR is stored under the DMS configuration

```
oSAPDIR.DIRDescription = sDirDescripton
…
```

‘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
```

‘Declare necessary objects

```
Dim oSAPDIR, oMasterRecord, oFnObjectId
```

‘Create all objects

```
Set oMasterRecord =
Server.CreateObject("DMSComponent.CMasterRecord")

Set oFnObjectId =
Server.CreateObject("DMSComponent.CFNObjectID")

Set oSAPDIR = oSAPDMS.createDIR(“DRW”,”*”,”000”,”00”)
```

Note: The parameter to create a DIR object depends on the chosen DIR Type. The sign for automatic Dir number creation differs.

‘Set properties for objects to link.

```
oMasterRecord.MasterRecordID = sMaterialID

oMasterRecord.MasterRecordType = sMasterType
```

Note: The MasterType is stored in the Configuration Program, under **DMS→UIComponents**.

```
oFnobjectId.FnObjectId = sFnObjectId
```

‘Set connection for SAP DIR

```
Call oSAPDIR.setConnection(oConnection)
```

‘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 = sInitailDirStatus
```

Note: The initial status for a DIR is stored under the DMS configuration

```
oSAPDIR.DIRDescription = sDirDescripton
```

…

‘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
```

Changing an existing Document Info Record within SAP System

To change properties of an existing SAP DIR object, a local `CSAPDIR` object should be created. The `getDIR` function of the `CSAPDMS` object returns an initialized `CSAPDIR` object, containing all links and properties of the SAP DIR. The `Save` function of a `CSAPDIR` object realizes all changes made to 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”)
```

‘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
```

‘Declare objects

```
Dim oExistingDIR
```

‘Create DIR object with information of an existing SAP DIR. Uses the DMSComponent.dll

```
Set oExistingDIR = oSAPDMS.getDIR(“DRW”,”10000000123”,
”000”,”00”)
```

‘Set reference to a connection object for this DIR

```
Call oExistingDIR.setConnection(oConnection)
```

‘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
```

Deleting the Links between a FileNet Object and a Master Record

To delete the links between a FileNet object and master record, both MasterRecords and FnObjects collections should be cleared. Subsequently, the `Save` function should be called to save the changes to a DIR in the SAP system. The Delete function sets the DeleteIndicator of the SAP DIR. The SAP DIR still exists within the SAP system and it is possible to access the SAP DIR object.

…

```
Call oExistingDIR.MasterRecords.Clear
```

```
Call oExistingDIR.FNObjectIds.Clear
Call oExistingDIR.save
Call oExistingDIR.delete
```

4

Creating a Custom Display Connector

IDM Desktop for R/3 uses Display Connectors to display FileNet objects, requested by SAPGUI. It is now possible to configure Custom Display Connectors that are responsible to display these objects. The Display Connector should be delivered as an ActiveX component. If the ActiveX component is registered successfully, use the Configuration Program of the IDM Desktop for R/3 to configure the Custom Display Connector.

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. The IDM Desktop for R/3 expects the following two functions on the Display Connector object:

- `DisplayObject`: Opens the program responsible for displaying the FileNet object. The following are the parameters for the `DisplayObject` function:
    - `ObjectId`: represents the Id of the FileNet object to display.
    - `ObjectType`: represents the object type of the document to display (e.g. `DwFolder`, `DwDocument`, `DwLocalFile` etc.). all allowed object types are defined in the Enum `DwObjectType`.
    - `LibraryName`: specifies the library to which the object belongs. This is an optional parameter.
- `CloseViewer`: Closes the viewer.

5
Logon/Logoff from FileNet Libraries

IDM Desktop for R/3 provides an ActiveX Object to logon and logoff from FileNet Libraries. The DwLogon.dll is the component that contains this object. The object is the `ConnectionManager` Object. This object also enables automatic background logoff from a FileNet Library after a certain period of time. This object can be used in the Custom Views to:

- Logon to a library.
- Logoff from a library.
- Logoff from all libraries.
- Automatic logoff from library after a certain period of time.

The following functions are provided for the `ConnectionManager` object:

- `SetUILogon`: Determines whether the FileNet logon dialog box is used or not. The following are the parameters for `SetUILogon` function:
  - `bUILogon`: used to enable or disable the FileNet logon dialog. If this is set to true, FileNet logon dialog box is used to logon to FileNet library. Otherwise, it it is set to false, then the dialog is not used.
- `LogonToLibrary`: Performs a logon to a specific FileNet library. The following are the parameters for `LogonToLibrary` function:
  - `Library`: specifies the library to which logon is to be performed. It is a reference to an IDMObjects.Library object.
  - `AutoLogoffPeriod`: specifies the period in minutes after which the user is automatically logged off from the selected library. The value of zero means that autologoff is disabled.

Note: A new connection to the library is established by logging on to the FileNet Library, if a connection does not already exist. A logon to an existing connection resets the timer.

- `LogoffFromLibrary`: Performs a logoff from a specific FileNet library. The following are the parameters for `LogoffFromLibrary` function:
  - `Library`: specifies the library for which logoff is to be performed.

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

- `LogoffFromAllLibraries`: Performs a logoff from all open FileNet libraries.

Note: All connection objects will be deleted.

- `StopTimer`: Stops the Auto logoff timer for specific FileNet library. The following are the parameters for `StopTimer` function:
  - `Library`: refers to the library for which the timer needs to be stopped. This parameters is an object of IDMObjects.Library.
- `RestartTimer`: Restarts the Autologoff timer for a specific FileNet library. The following are the parameters for `RestartTimer` function:
  - `Library`: refers to the library for which the timer needs to be restarted. This parameters is an object of IDMObjects.Library.

Note: The AutoLogoffPeriod is set to the time range specified during `LogonToLibrary`() function.

6
General Settings

Storing Settings in the Windows Registry

All Custom components should be aware of several settings used for IDM Desktop for R/3 e.g. logging. The Custom component may also store its own settings. These settings are stored in the registry.

These settings are stored in

\\HKEY_LOCAL_MACHINE\SOFTWARE\FileNet\Document Warehouse\Preferences.

The table below lists the relevant keys, value names, and their description for a Custom component:

Key	Value Name	Description
\Logging		Logging Settings
	FileLogFileName	Name of the log file
	FileLogLevel	Log level for logging to a file
	FileMaxLogFileSize	Maximum size of the log file.
	FileUseDailyLogFile	Setting to specify whether a daily log file needs to be created
	NtEventLogLevel	Setting to specify log level while logging to a windows event log.
\SapSystems\<active SapSystem\SapArchive\Archive ID>		Archive Mappings
	Library ID	Mapping of SAP Archive ID to FileNet Library ID: For IS:<object>:<domain>:<organization> 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

Key	Value Name	Description
	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 write able, all other keys are considered as read-only.

The table below lists the settings for DMS:

Key	Value Name	Description
\Dms\UIComponents\<name of new component>		The name of a 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 table below lists the settings for Custom DisplayConnectors:

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

7 Logging

All components can log their messages to the same log file as IDM Desktop for R/3.The log level determines the type of information that should be logged. The format of a log entry is:

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

The following information shall be provided by the Component:

- <L> = [e|w|i] log level of message: error, warning, or info.
- <ComponentName>: Refers to 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>: Refers to the log message itself.

Appendix A – Software License Agreement

FileNet End User Software License Agreement

READ THIS NOTICE CAREFULLY, THE SOFTWARE IS THE PROPRIETARY INTELLECTUAL PROPERTY OF FILENET CORPORATION AND IS SUBJECT TO THE MINIMUM TERMS AND CONDITIONS SET FORTH BELOW. THESE TERMS AND CONDITIONS MAY BE SUPERCEDED BY THE TERMS AND CONDITIONS OF THE SOFTWARE LICENSE ENTERED INTO BY YOUR EMPLOYER FOR THE USE OF FILENET SOFTWARE. BY USING THE SOFTWARE, YOU ACKNOWLEDGE IT IS FILENET PROPRIETARY INTELLECTUAL PROPERTY AND THAT A VALID SOFTWARE LICENSE WITH FILENET CORPORATION IS APPLICABLE. THEREFORE AT A MINIMUM, YOU AGREE TO BE BOUND BY THE FOLLOWING FILENET END USER SOFTWARE LICENSE TERMS AND CONDITIONS (HEREINAFTER “Agreement”):

1. **Definition of Software**

   The software consists of software owned by FileNet, as well as software owned by certain third party providers (“Third Party Providers”). Each software product includes any documentation relating to or describing such software, such as, logic manuals, flow charts, reference materials, and improvements or updates provided by FileNet (software and documentation collectively called "Software").

2. **Grant of License**

   A. Each Software product, including any documentation relating to or describing such Software, such as, but not limited to, manuals, flow charts and improvements or updates provided by FileNet (collectively "Software"), is furnished to End User under a personal, non-exclusive, nontransferable license solely for End User's own internal use on End User’s servers and client devices (“System”) in compliance with this license and all applicable laws and regulations. End User agrees that this license does not permit End User to: (1) use the Software for a service bureau application or (2) rent, lease, or sublicense the Software; (3) modify or remove any proprietary notices; or (4) transfer the Software without prior written consent from FileNet. The Software is licensed to the End User, not sold.

   B. The Software may only be copied, in whole or in part (with the proper inclusion of FileNet's copyright notice and any other proprietary notice and/or trademarks on such Software), as may be necessary and incidental for archival purposes or to replace a worn or defective copy.

   C. Title to and ownership of the Software and any portions (or any modifications, translations, or derivatives thereof, even if unauthorized) and all applicable rights in patents, copyrights and trade secrets in the Software shall remain exclusively with FileNet and its licensors, if any. Software provided hereunder is valuable, proprietary, and unique, and End User agrees to be bound by and observe the proprietary nature thereof as provided herein. End User agrees to take diligent action to fulfill its obligations hereunder by instruction or agreement with its employees or agents (whose confidentiality obligations shall survive termination of employment or agency) who are permitted access to the Software. Access shall only be given on a need-to-know basis. Except as set forth in this Agreement or as may be permitted in writing by FileNet, End User shall not use, provide or otherwise make available the Software or any part or copies thereof to any third party. End User shall not reverse engineer, decompile or disassemble the Software or any portion thereof, nor otherwise attempt to create or derive the source code. End User acknowledges that unauthorized reproduction, use, or disclosure of the Software or any part thereof may cause irreparable injury to FileNet and/or its licensors, who may therefore be entitled to injunctive relief to enforce these license restrictions, in addition to any other remedies available at law, in equity, or under this Agreement. Further, the trademarks are owned by the respective trademark holder.

   D. FileNet agrees that End User’s affiliates (business entities of which End User owns or controls more than fifty {50%} percent of the voting rights or the controlling body of the business entity) may use the Software; provided that prior to any affiliate’s use of the Software: (i) End User accepts responsibility for the acts or omissions of such affiliates as if they were End User’s acts or omissions; (ii) End User shall indemnify FileNet against losses or damages suffered by FileNet arising from breach of this Agreement by any such affiliate; and (iii) such use shall not constitute an unauthorized exportation of the Software or documentation under U.S. Government laws and regulations.

3. **Termination.** FileNet shall have the right to terminate End User's license if End User fails to pay any and all required license fees or otherwise fails to comply with these license terms and conditions. Upon expiration of the license term or upon notice of such termination, End User shall immediately return or destroy the Software and all portions and copies thereof as directed by FileNet and, if requested by FileNet, shall certify in writing as to the destruction or return of the same. All confidentiality and non-disclosure obligations herein shall survive termination.

4. **Limited Warranty**

   A. FileNet warrants that it has good and clear title to or has the right to sublicense the Software being licensed hereunder, free and clear of all liens and encumbrances.

   B. FileNet warrants for a period of one year from the Shipment Date, Software used in a manner for which it was designed will perform the functions described in the applicable FileNet documentation supplied at the time of delivery provided that, (i) Software is continuously subject to a FileNet Software Support contract, (ii) any substantial nonconformance is reproducible, and (iii) the substantial nonconformance is not caused by third party software or hardware not specified in FileNet's documentation or not expressly authorized in advance by FileNet. FileNet's sole obligation and liability hereunder shall be to use reasonable efforts to remedy any material non-conformance which is reported to FileNet in writing within the warranty period.

   C. End User accepts sole responsibility for, system configuration, design and requirements, selection of the software for the intended results, modifications, changes or alterations.

   D. THERE ARE NO OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT WITH RESPECT TO THIS AGREEMENT, THE AGREEMENT, OR THE SOFTWARE LICENSED HEREUNDER. FILENET DOES NOT WARRANT THAT THE OPERATION OF THE SOFTWARE WILL BE UNINTERRUPPTED, THAT THE SOFTWARE IS ERROR-FREE, OR THAT ALL ERRORS CAN BE CORRECTED.

5. **Export.** End User agrees that the Products purchased hereunder will not be exported directly or indirectly, separately or as part of any system, without first obtaining a license from the U.S. Department of Commerce or any other appropriate agency of the U.S. Government, as required.

6. **Restrictions on Use.** End User acknowledges that one of FileNet's Third Party Providers provides application integration software as part of the Software (the "Integration Software"). End User agrees to use the Software solely as follows: (i) FileNet's workflow or content management software will be the triggering source and/or the destination of the transaction managed by the Integration Software; (ii) FileNet's workflow or content management functionality must be a necessary part of the transaction managed by the Integration Software such that without such FileNet functionality, the transaction managed by the Integration Software could not process; (iii) the Software will not be used or configured in such a way as to only provide the functionality that the Integration Software provides; and (iv) all derivative works of the Software made by or for End User are subject to the foregoing restrictions.

7. **Choice of Law.** The laws of the State of California will govern the construction and operation of this Agreement without regard to the conflict of laws provisions thereof.

8. **U.S. Government Restricted Rights**. The Software is Commercial Software and the Software and Documentation are provided with Restricted Rights. Use, duplication or disclosure by the Government is subject to restrictions as set forth in paragraph (c) (1) (ii) of the Rights in Technical Data and Computer Software Clause at DFARS 252.227-7013 or subparagraphs (c) (1) and (2) of the Commercial Computer Software-Restricted Rights at 48 CFR 52.227-19, as applicable. Contractor/manufacturer is FileNet Corporation, 3565 Harbor Blvd., Costa Mesa, California 92626.

INDEX