

SAP ArchiveLink (BC-SRV-ARL)



HELP.BCSRVARL

Release 4.6C



Copyright

© Copyright 2001 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft®, WINDOWS®, NT®, EXCEL®, Word®, PowerPoint® and SQL Server® are registered trademarks of Microsoft Corporation.

IBM®, DB2®, OS/2®, DB2/6000®, Parallel Sysplex®, MVS/ESA®, RS/6000®, AIX®, S/390®, AS/400®, OS/390®, and OS/400® are registered trademarks of IBM Corporation.

ORACLE® is a registered trademark of ORACLE Corporation.

INFORMIX®-OnLine for SAP and Informix® Dynamic Server™ are registered trademarks of Informix Software Incorporated.

UNIX®, X/Open®, OSF/1®, and Motif® are registered trademarks of the Open Group.

HTML, DHTML, XML, XHTML are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

JAVA® is a registered trademark of Sun Microsystems, Inc.

JAVASCRIPT® is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

SAP, SAP Logo, R/2, RIVA, R/3, ABAP, SAP ArchiveLink, SAP Business Workflow, WebFlow, SAP EarlyWatch, BAPI, SAPPHIRE, Management Cockpit, mySAP.com Logo and mySAP.com are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other products mentioned are trademarks or registered trademarks of their respective companies.

Icons

| Icon | Meaning |
|---|----------------|
|  | Caution |
|  | Example |
|  | Note |
|  | Recommendation |
|  | Syntax |
|  | Tip |

Contents

| | |
|--|-----------|
| SAP ArchiveLink (BC-SRV-ARL) | 9 |
| Information for Everyone..... | 10 |
| Introduction to SAP ArchiveLink..... | 11 |
| Business Document Navigator | 13 |
| Displaying Hit Lists | 16 |
| Displaying Documents | 17 |
| Displaying Detailed Information on a Document..... | 18 |
| Storing Documents..... | 19 |
| Storing Documents Using Drag and Drop..... | 21 |
| Create New Documents | 22 |
| Storing and Displaying WWW Addresses..... | 24 |
| BDS Note | 25 |
| Editing Notes..... | 26 |
| Note Hit List..... | 27 |
| Note Display..... | 28 |
| Displaying Keywords..... | 29 |
| Displaying the Version String..... | 30 |
| Copying Documents | 31 |
| Deleting Documents..... | 32 |
| Editing Documents | 33 |
| Changing Attributes..... | 34 |
| Creating Annotations..... | 35 |
| Refreshing the Hit List..... | 36 |
| Navigation to BOR Objects | 37 |
| Exporting Documents..... | 38 |
| Transporting Documents..... | 39 |
| Terms..... | 40 |
| Documents | 42 |
| Document Types | 43 |
| Technical Document Classes..... | 45 |
| Object Types and Methods | 46 |
| Business Objects / Objects | 49 |
| Synchronous and Asynchronous Storing..... | 50 |
| Administration Concept..... | 51 |
| Content Servers | 53 |
| Links | 54 |
| Link Tables | 55 |
| Incoming Documents | 56 |
| Frontend Communication..... | 57 |
| Storage Strategies | 59 |
| Concept: Storing for Subsequent Entry | 62 |
| Procedure: Storing for Subsequent Entry..... | 64 |
| Concept: Storing and Entering..... | 66 |
| Process Flow: Storing and Entering | 67 |
| Concept: Storing for Subsequent Assignment..... | 68 |
| Procedure: Storing for Subsequent Assignment | 70 |

| | |
|---|------------|
| Concept: Storing and Assigning..... | 72 |
| Procedure: Storing and Assigning | 73 |
| Concept: Assigning then Storing | 74 |
| Process Flow: Assigning then Storing | 75 |
| Concept: Storing with Bar Codes..... | 76 |
| Process Flow: Early Storing with Bar Codes..... | 78 |
| Process Flow: Late Storing with Bar Codes | 79 |
| SAP ArchiveLink Workflow Wizard | 80 |
| SAP ArchiveLink document types Customizing Wizard..... | 81 |
| Bar codes | 82 |
| Outgoing documents | 84 |
| Print lists | 85 |
| Transfer to R/3 Application Components | 86 |
| Viewers | 87 |
| Connection to External Viewers | 88 |
| Connection to Generic Object Services..... | 89 |
| Information for End Users | 90 |
| Searching for Stored Documents | 91 |
| Maintaining User-Specific Storage Settings..... | 93 |
| Storing for Subsequent Entry | 94 |
| Early Storing with Bar Codes..... | 97 |
| Storing and Entering | 98 |
| Assigning then Storing..... | 100 |
| Late Storing with Bar Codes | 102 |
| Searching for, Displaying and Printing Print Lists | 104 |
| Transferring Stored Documents | 106 |
| Displaying Stored Documents Generically..... | 107 |
| Assigning Documents Generically | 108 |
| Storing Documents Generically..... | 109 |
| Assigning Bar Codes Generically..... | 110 |
| Information for Administrators | 111 |
| The Administration Tool Monitor..... | 112 |
| Monitor Screen..... | 113 |
| Monitor Environment..... | 116 |
| Prerequisites for Storage Requests | 118 |
| Archive Device and Spool | 119 |
| Storing Outgoing Documents | 121 |
| Asynchronous Storing of Print Lists and Archive Files | 123 |
| Displaying Asynchronous Requests | 126 |
| Retrieving Archive Files and Print Lists | 128 |
| Displaying Requests in Request Management..... | 130 |
| Storing with Bar Codes | 131 |
| Process Flow: Storing with Bar Codes | 134 |
| Queue Management: Error Handling | 136 |
| Administration of Stored Documents | 139 |
| Linking Business Objects with Stored Documents | 141 |
| Administration of Incoming and Outgoing Documents..... | 143 |

| | |
|---|------------|
| Searching for Stored Documents..... | 144 |
| Displaying Link Entries for Stored Documents | 146 |
| Creating an Incoming Document | 148 |
| Reassigning a Stored Document | 149 |
| Assigning a Stored Document Subsequently | 150 |
| Administration of Print Lists..... | 151 |
| Searching for Stored Print Lists | 152 |
| Displaying Link Entries for Stored Print Lists..... | 153 |
| Administration of Archive Files..... | 155 |
| Searching for Stored Archive Files | 156 |
| Displaying Link Entries for Stored Archive Files..... | 157 |
| Authorizations | 158 |
| Authorizations..... | 159 |
| Authorization Checks for SAP ArchiveLink | 161 |
| Roles for SAP ArchiveLink..... | 163 |
| Information for Persons Making Customizing and System Settings | 164 |
| Customizing | 165 |
| Basic Settings..... | 168 |
| Specific Customizing..... | 169 |
| Communication | 172 |
| Protocols | 173 |
| Protocols: Concept..... | 175 |
| Generating Protocols | 179 |
| Storing..... | 181 |
| Storing Using Internal Entry Dialog | 182 |
| Storing PC Files, and Documents Entered Using External Entry Dialogs..... | 183 |
| Starting an External Application | 187 |
| Display | 188 |
| Displaying Using Internal Viewers and OLE Automation | 189 |
| Displaying PC Files and Displaying Using External Viewers and OLE Automation..... | 190 |
| Displaying PC Files and Displaying Using External Viewers via EXECUTE..... | 193 |
| Displaying Outgoing Documents in PDF Format..... | 194 |
| Close Window..... | 195 |
| Applications | 196 |
| Syntax: OLE Automation..... | 198 |
| Calling Object Methods..... | 200 |
| Setting Object Attributes | 202 |
| Retrieval of Object Attributes | 203 |
| Example for Applications | 204 |
| Information for SAP Business Workflow | 205 |
| Default Settings for Storage Scenarios | 206 |
| Further Default Settings | 207 |
| Information for Developers | 210 |
| Storing Incoming Documents | 211 |
| Storing with Bar Codes | 212 |
| Storing Outgoing Documents | 214 |

| | |
|---|------------|
| Storing Outgoing Documents | 215 |
| Form Overlay..... | 217 |
| Storing Print Lists | 219 |
| Storing Print Lists Asynchronously..... | 220 |
| Indexing Print Lists..... | 221 |
| ARCHIVELINK_DEFINE_KEY..... | 222 |
| ARCHIVELINK_CREATE_INDEX..... | 224 |
| ARCHIVELINK_CREATE_DARC..... | 225 |
| Index Information and Hypertext Links | 227 |
| Information for Content Server Suppliers | 228 |
| SAP Content Server HTTP 4.5 Interface..... | 229 |
| Introduction..... | 230 |
| Definition of Terms..... | 231 |
| Implementation | 233 |
| Security | 234 |
| secKey | 235 |
| Protection / Right of Access | 237 |
| Syntax | 239 |
| General | 240 |
| Coding in the URL..... | 241 |
| Coding in the Response Body | 243 |
| Functions..... | 244 |
| Access Functions..... | 246 |
| info | 247 |
| get..... | 252 |
| docGet..... | 254 |
| create..... | 258 |
| HTTP-PUT..... | 260 |
| HTTP-POST multipart/form-data..... | 261 |
| mCreate | 263 |
| append..... | 266 |
| update..... | 268 |
| HTTP-PUT..... | 270 |
| HTTP-POST multipart/form-data..... | 271 |
| delete | 272 |
| search..... | 274 |
| attrSearch | 277 |
| Administration Functions | 283 |
| putCert..... | 284 |
| serverInfo..... | 286 |
| Error Codes..... | 289 |
| Appendix | 290 |
| Parameters and Key Words..... | 291 |
| Information on Migrating Existing Archives..... | 296 |
| SAP ArchiveLink 4.5 Bar Code BAPI..... | 299 |
| Communication | 301 |

| | |
|---|-----|
| Protocols | 302 |
| Protocols: Concept..... | 304 |
| Generating Protocols..... | 308 |
| Storing..... | 310 |
| Storing Using Internal Entry Dialog | 311 |
| Storing PC Files, and Documents Entered Using External Entry Dialogs..... | 312 |
| Starting an External Application | 316 |
| Display | 317 |
| Displaying Outgoing Documents in PDF Format..... | 318 |
| Displaying PC Files and Displaying Using External Viewers via EXECUTE..... | 319 |
| Displaying PC Files and Displaying Using External Viewers and OLE Automation..... | 320 |
| Displaying Using Internal Viewers and OLE Automation | 323 |
| Close Window..... | 324 |
| Applications | 325 |
| Syntax: OLE Automation..... | 327 |
| Calling Object Methods..... | 329 |
| Setting Object Attributes | 331 |
| Retrieval of Object Attributes | 332 |
| Example for Applications | 333 |

SAP ArchiveLink (BC-SRV-ARL)

The SAP ArchiveLink documentation is organized user-specifically.

The first section contains general information on SAP ArchiveLink, particularly on the various document types and the corresponding different storage strategies.

In the section for end users, you, as the person responsible, receive information on the procedure for the different storage strategies, on the search of stored documents, and so on.

The section for administrators contains information on the administration tool monitor provided by SAP ArchiveLink and on the administration tool for stored documents, as well as information on authorizations.

The section aimed at those configuring the system and Customizing provides information on possible Customizing settings and protocol and application maintenance.

The section on Business Workflow contains information on the close connection between SAP ArchiveLink and SAP Business Workflow, which is particularly evident in the different storage strategies.

Information on the integration of SAP ArchiveLink in the different R/3 applications can be found in the section Information for Developers.

The last main section, Information for Content Server Suppliers, contains the specification of the SAP ArchiveLink 4.5 interface, as well as information on the SAP ArchiveLink 4.5 Barcode BAPI.

Information for Everyone

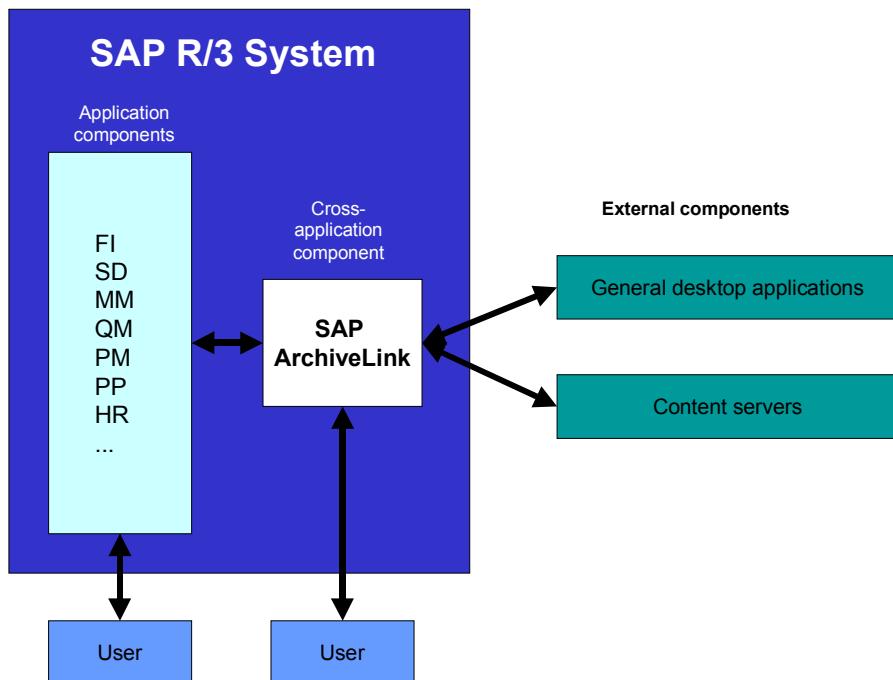
Information for Everyone

The following sections provide a general introduction and are intended for all those interested in the cross-application tool **SAP ArchiveLink®**. The general SAP ArchiveLink functions and concepts are explained.

Introduction to SAP ArchiveLink

SAP ArchiveLink® is a communication interface between R/3 application components and external components (content servers or general desktop applications). It is integrated in the basis component of the SAP R/3 System.

SAP ArchiveLink®



SAP ArchiveLink® has the following interfaces:

- User interface
For an explanation of how to use this interface, see [Information for End Users \[Page 90\]](#).
- Interface to the R/3 application components

This is made up of function modules, which must be integrated into the R/3 application components, if content server functions are to be used. Business objects from R/3 applications are linked to stored documents. In addition, object methods are available that enable the flexible integration of storing with SAP ArchiveLink®, into SAP Business Workflow®.

The interface to the R/3 application components is described in the following sections:

- [Information for Administrators \[Page 111\]](#)
- [Information for Persons Making Customizing and System Settings \[Page 164\]](#)
- [Information for SAP Business Workflow \[Page 205\]](#)
- [Information for Developers \[Page 210\]](#)

Introduction to SAP ArchiveLink

For more information about integrating storing using SAP ArchiveLink in R/3 application components, see the documentation [SAP ArchiveLink - Scenarios in Applications \[Ext.\]](#).

- **Interface to external components**

The interface to external content servers is described in the section [Information for Content Server Suppliers \[Page 228\] \[Page 228\]](#).



In the following documentation, the term **SAP ArchiveLink** will be used synonymously with the official product name **SAP ArchiveLink®**.

Likewise, the term **SAP Business Workflow** will be used synonymously with the official product name **SAP Business Workflow®**.

Business Document Navigator

Purpose

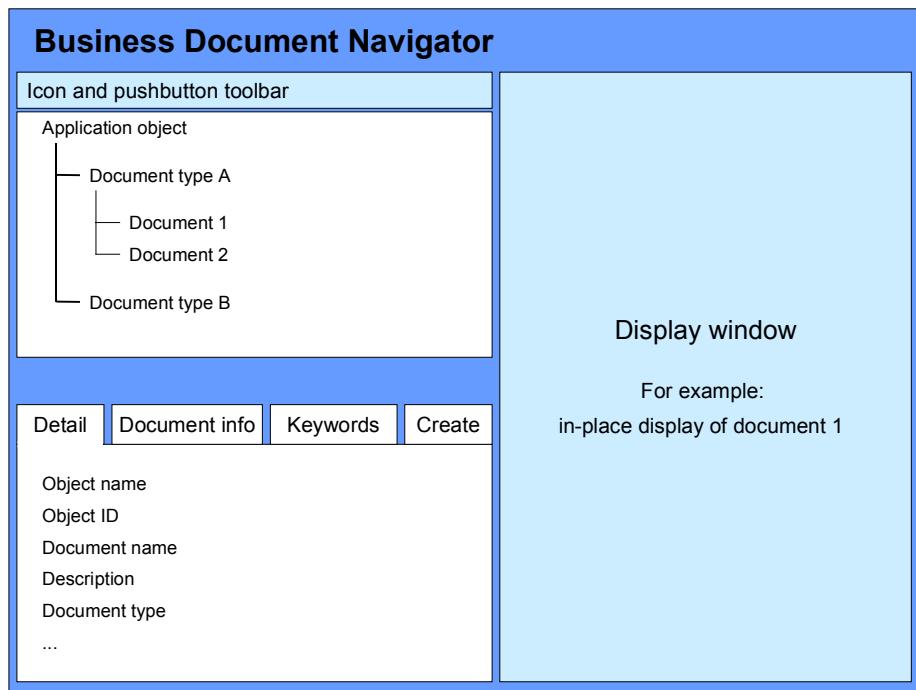
The **Business Document Navigator (BDN)** is the central tool for processing documents. You can use the BDN to execute standard document management functions on documents managed, for example, by SAP ArchiveLink.

To go to the Business Document Navigator, choose *Office* → *Business Documents* → *Documents* → *Find*. Alternatively, choose *Tools* → *Business Documents, Environment* → *Business Documents* in the administration menu for business documents.



You do not need to make any settings in Customizing (IMG).

Overview Graphic



In the upper part of the screen, the relevant documents, sorted by document type (the document types in turn belong to specified application objects) are displayed in the tree. The lower left part of the screen contains tab pages with the functions Detailed display, Document information (version string), Keywords and Storing. On the right-hand side of the screen, you can display a selected document in-place.

Features

The BDN provides the following functions:

- Display hit list

Business Document Navigator

You see the hit list in the form of a hierarchical list.

For more information, see [Displaying the Hit List \[Page 16\]](#).

- Display individual documents

You view the individual documents directly in the dynpro or in an external viewer.

For more information, see [Displaying Documents \[Page 17\]](#).

- Display detailed information on stored documents

For more information, see [Displaying Detailed Information on a Document \[Page 18\]](#).

- Store existing documents

You can display standard document types and object-specific document types, and store corresponding documents by double-clicking.

For more information, see [Storing Documents \[Page 19\]](#).

- Create new documents

You start applications that are supported by SAP Desktop Office Integration in-place and create new documents.

For more information, see [Creating New Documents \[Page 22\]](#).

- Create and edit notes

You can create, display, edit and delete general and private notes for documents and application objects.

For more information, see [BDS Note \[Page 25\]](#).

- Display keywords for a document

You can display all keywords that were stored for a document.

For more information, see [Displaying Keywords \[Page 29\]](#).

- Display version string

You can display all the versions and variants for a document.

For more information, see [Displaying the Version String \[Page 30\]](#).

- Copy documents

For more information, see [Copying Documents \[Page 31\]](#).

- Transport documents

For more information, see [Transporting Documents \[Page 39\]](#).

- Delete documents

For more information, see [Deleting Documents \[Page 32\]](#).

- Edit documents

You can edit DOI-compatible documents that are stored using upload. You can overwrite the current document, create a new document and create a new version or variant.

For more information, see [Editing Documents \[Page 33\]](#).

- Change existing keywords

Business Document Navigator

You can change the existing keywords and descriptive text for a document.

For more information, see [Changing Attributes \[Page 34\]](#).

- Refresh the hit list
 - For more information, see [Refreshing the Hit List \[Page 36\]](#).
- Annotations for BDS documents that are displayed using the EAI viewer
 - For more information, see [Creating Annotations \[Page 35\]](#).
- Display documents using HTML control
 - For more information, see [Displaying Documents \[Page 17\]](#).
- Store WWW addresses
 - For more information, see [Storing Documents \[Page 19\]](#) and [Storing and Displaying WWW addresses \[Page 24\]](#).
- Navigation to any WWW address
 - For more information, see [Storing and Displaying WWW Addresses \[Page 24\]](#).
- Export documents to frontend
 - For more information, see [Exporting Documents \[Page 38\]](#).
- Lock documents when changes are being made
 - For more information, see [Editing Documents \[Page 33\]](#).
- Create a transport request when documents are deleted
 - For more information, see [Transporting Documents \[Page 39\]](#).
- Store documents using drag and drop
 - For more information, see [Storing Documents Using Drag and Drop \[Page 21\]](#)
- Display hit list and documents in the **SAPGUI for HTML** and in the **SAPGUI for Java**
- Direct navigation to BOR objects from the BDN
- Display SAP ArchiveLink documents (faxes) in *Single Page Tiff Format*



You can call the functions *Display document*, *Display detailed information*, *Display document information*, *Display keywords*, *Change document*, *Change attributes*, *Copy document*, *Export document*, *Transport document*, *Delete document* and *Notes* from the display tree via the context menu.

Displaying Hit Lists

Use

This function generates a hierarchical list of documents that were stored for a certain application object.

Prerequisites

When you call the Business Document Navigator, you must also specify a class name (or BOR object type). If no object ID is specified, the documents for all object IDs are displayed (see also [Searching for Stored Documents \[Page 91\]](#)). The hit list is sorted by object ID and then, within each object ID, by document type.

Features

The hit list is displayed as a tree structure.

The root node of the tree is always the specified class. If there are several object IDs, the hit list is sorted by object ID and then, within each object ID, by document type. The documents come under the document types. Always the most up to date document is listed. The following attributes are displayed:

- Descriptive text
- File name
- Date created

If possible, the document class is displayed using an appropriate symbol.

You can call the following functions from the hit list using the context menu:

- Display document
- Change document
- Delete document
- Transport document
- Export document
- Create and process notes
- Copy document
- Display detailed information
- Display versions
- Display keywords
- Change attributes

Activities

By opening the individual nodes, you can see the documents for the object ID.

Displaying Documents

Use

This function displays individual documents.

Features

The document is generally displayed in the dynpro itself, that is, in place. The document is displayed either via SAP Desktop Office Integration (DOI, see also [BC - Desktop Office Integration \[Ext.\]](#)) or using viewer control (see also [Viewer for Displaying Original Files in Standard \[Ext.\]](#) and [SAP HTML Viewer \[Ext.\]](#)). If possible, documents of classes BMP, GIF, JPG, PCX, PS, TIF, TXT and STL are displayed in place using the viewer control first named, the EAI viewer. If, however, a DOI-compatible application is found, this is used to display the document. MS Office 97, Lotus SmartSuite 98 and the Acrobat Reader are currently supported, that is, document classes DOC, XLS, PPT, 123, PRZ, LWP and PDF. Documents of classes GIF, JPG, MOV, HTM and XML can be displayed in place using [HTML control \[Ext.\]](#).

Documents of document classes FAX and OTF are displayed according to the settings in the relevant SAP ArchiveLink protocol (see [Maintaining Protocols \[Ext.\]](#)).

If it is not possible to display the document in place, an appropriate viewer that is available on the PC is started and the document is displayed "out place".

Activities

To display, double-click on the relevant document in the tree.

Displaying Detailed Information on a Document

Displaying Detailed Information on a Document

Use

This function is used to display detailed information on individual documents.

Features

The following information is currently displayed on the tabstrip controls tab page:

- Class name
- Object ID
- Document name
- Descriptive text
- Document type
- Language
- Version number and variant number
- Created by
- Created on
- Changed by
- Changed on
- Document class
- MIME type
- Content repository
- Document ID
- Status



As of Release 4.6C, fixed attributes can be transferred when the BDN is called.
These fixed attributes are also displayed in the detailed information.

Activities

Select a document in the tree. The detailed information is displayed on the tabstrip controls tab page, in the lower left part of the screen.

Storing Documents

Use

This function is used to store existing documents.

Prerequisites

If several object keys, that is, object IDs, are displayed in the upper left part of the screen, before you store a new document you have to select an object key under which the document is to be stored.

Features

On the tab page *Create*, the following document types are displayed in the lower left part of the screen in tree form:

- Standard document types
 - Screen
 - Document class TIFF
 - Table template
 - Document class XLS
 - Presentation
 - Document class PPT
 - Text
 - Document class DOC
 - WWW address
 - Document class HTM
- Object-specific document types

The standard document types are defined in [Maintaining Document Types \[Ext.\]](#) and exist for each class. In document type maintenance, you must assign a document class to each document type. You can characterize a document type as a standard document type using the standard flag.

You must also create object-specific document types in document type maintenance. You assign document types to BOR object types in [link maintenance \[Ext.\]](#).

When you store a document, you can specify the following additional information in a dialog box:

- Descriptive text
- Document language
- Up to 5 keywords

Storing Documents



The search for documents via *Office → Business Documents → Documents → Find* can be limited using the descriptive text and keywords.

When you store a WWW address, the URL to be created is entered using a dialog box (see also [Storing and Displaying WWW Addresses \[Page 24\]](#)).



Up to Release 4.6B, the content of all documents that were created and stored using the BDN is stored physically in the R/3 database.

As of Release 4.6C, content categories can be assigned to the classes. The documents are then stored in the [content repository \[Ext.\]](#) that is assigned to the specific category. If no category is assigned to the classes, the document is stored in the R/3 database.

Activities

Choose the tab page *Create* in the lower left part of the screen: Double-clicking on the required document type calls the file selection box. You can then store an existing file. Alternatively, you can call the file selection box by choosing *Import file* in the context menu.

Storing Documents Using *Drag and Drop*

Use

This function enables you to store documents in the BDN using *drag and drop*. The documents can be at the following places:

- On the desktop
- In the Microsoft Windows Explorer

The document has to be moved to EAI control. It is displayed in the EAI viewer and linked to the object that the previous object also has a link to.

Prerequisites

A document is displayed in the BDN using the EAI viewer (see also [Viewer for Displaying Original Files in Standard \[Ext.\]](#)).

Activities

1. Display a document in the BDN using the EAI viewer.
2. Drag the other document from the desktop or from the MS Windows Explorer into the EAI viewer.
3. In the following dialog box, enter a descriptive text and the language.
You can also specify up to 5 keywords.
4. Confirm your entries.

Create New Documents

Create New Documents

Use

This function is used to create new documents.

Prerequisites

If several object keys, that is, object IDs, are displayed in the upper left part of the screen, before you store a new document you have to select an object key under which the document is to be stored.

Features

On the tab page *Create*, standard document types and object-specific document types are displayed in the lower left part of the screen in tree form (see also [Storing Documents \[Page 19\]](#)).

You can start applications that are supported by SAP Desktop Office Integration (DOI) (see also [BC - Desktop Office Integration \[Ext.\]](#)) in-place and create new documents. SAP DOI currently supports MS Office 97 and Lotus SmartSuite 98.

When you select an application, it is started on the frontend in the display area of the Business Document Navigator. You can then create the document.



Temporary storing of the document is only possible within the application: The document is then stored on the local hard disk.

Choose the icon *Store document* to store the document when you have finished editing, and exit the relevant application.

When you store a document, you can specify the following additional information in a dialog box:

- File name
- Descriptive text
- Document language
- Up to 5 keywords



The search for documents via *Office* → *Business Documents* → *Documents* → *Find* can be limited using the descriptive text and keywords.



The content of all documents that were created and stored using the BDN is stored physically in the R/3 database.

As of Release 4.6C, content categories can be assigned to the classes. The documents are then stored in the [content repository \[Ext.\]](#) that is assigned to the specific category. If no category is assigned to the classes, the document is stored in the R/3 database.

Create New Documents

Activities

1. Choose the tab page *Create* in the lower left part of the screen: Select the required document type and choose *Start application* in the context menu. Select the required application from the list of applications that are on the frontend and supported by DOI integration.
The application is started in-place in the display area in the BDN.
2. Create a document in the required application and store the document by choosing the icon *Store document*.

Storing and Displaying WWW Addresses

Storing and Displaying WWW Addresses

Use

This function is used to navigate to WWW addresses with the BDN using [HTML control \[Ext.\]](#). The pushbutton *Go to WWW address* is available for this in the BDN navigation bar, if HTML control is active.

WWW addresses can also be stored using the BDN. For more information, see [Storing Documents \[Page 19\]](#).

Activities

To navigate to a WWW site, proceed as follows:

1. Display the WWW site in HTML control and start the browser in-place.
2. Choose *Go to WWW address* and specify a URL.

To navigate to your personal homepage, choose the pushbutton *Homepage*.

3. Confirm your entry.

To store a WWW address for a specific object, proceed as follows:

1. In the navigation tree, position the cursor on the object for which the URL is to be stored.
2. Choose the document type *WWW address* by double-clicking on it in the list of standard document types.
3. In the subsequent dialog box, enter a description as well as the WWW address or copy the URL of the displayed site using Import.

BDS Note

Purpose

BDS Note enables you to use the general note functions for documents and application objects in the Business Document Navigator. This note function can be called by choosing the icon  in the BDS application toolbar. In the note dialog box, you can see at a glance which notes already exist and you can edit or display them or create new notes.

You can create both private and general notes.



When documents are deleted or transported from the BDN, the system checks whether notes exist for the relevant documents. After an appropriate query, these notes are then also deleted or transported.

Features

- You can display existing notes
- You can create and delete notes. Private notes can also be changed.
Private notes are marked with a specific flag.
- You can store notes.



BDS Note enables you to display and enhance earlier SAP ArchiveLink notes. The prerequisite for this is that notes from SAP ArchiveLink are stored in an HTTP storage system, that is, the notes have to be in an HTTP storage system.

Constraints

The note functions are currently only available for Microsoft platforms.

Editing Notes

Editing Notes

Use

You go to the tab page *Create/Edit* in order to:

- Create a new note, that is, a note does not yet exist
- Edit a private note
- Store an existing note as a file

To do this, choose the icon  in the text edit control.

- Store a note locally
- To do this, choose the icon  in the text edit control.

Prerequisites

1. You have selected a document or application object in the BDN navigation area or in the Document information for which you want to create or edit a note.
2. You have selected the icon .

Activities

To create a new note, proceed as follows:

1. Enter a title for your note.
2. Create the note yourself in the text edit control.
3. If the standard attribute *General note* is to be changed to *Private note*, select the flag *Private note*.



You can only change an existing note later if it is a private note.

4. Save your new note by choosing .

After you have saved, the system goes to the tab page *Display*.

To edit a private note, proceed as follows:

1. Select the private note in the hit list and choose .

You go to the tab page *Create/Edit*, on which your note is displayed.

2. Change your note.

You can extend the content of your note and/or add new content. Choose the relevant icons in the text edit control for general text editing functions.

3. Save your changes by choosing .

Note Hit List

Use

You go to the tab page *Hit list* in order to:

- obtain an overview of all existing notes
- delete notes, in accordance with your authorizations.

To delete a note, select the note that you want to delete and choose the icon .

Prerequisites

3. You have selected a document or application object in the BDN navigation area or in the document information for which a note already exists.
4. You have selected the icon .

Features

The following information for existing notes is displayed in the hit list:

- Type
 - General note or
 - Private note
- Description
 - Title
- Creator
 - Creator's user name
- Creation time
 - Date and time of creation
- Last changed by
 - User name of last person to make changes
- Change time
 - Date and time of last change

For more detailed information about a note, choose the icon .

Choose the relevant icons for further list functions in the hit list.



To display a note, select the relevant note and choose the tab page [Display \[Page 28\]](#).

Note Display

Note Display

Use

You go to the tab page *Display* in order to:

- Display an existing note
- Search for specific terms in the text of the note
 - To do this, choose the icon  in the text edit control.
- Store a note as a local file.
 - To do this, choose the icon  in the text edit control.

Prerequisites

- You have created a note.
- You have selected an existing note in the hit list and chosen the tab page *Display*.

Displaying Keywords

Use

This function is used to display keywords.

Features

On the tab page *Keywords* in tabstrip controls in the lower left part of the screen, the keywords that exist for a document are displayed:

- All keywords created when documents were created
See also [Creating New Documents \[Page 22\]](#).
- All keywords created when documents were stored
See also [Storing Documents \[Page 19\]](#).
- All keywords created when attributes of SAP ArchiveLink documents were created or changed
See also [Changing Attributes \[Page 34\]](#).

Activities

In the upper left part of the screen, select the document for which you want keywords to be displayed and choose the tab page *Keywords* in the lower left part of the screen.

Displaying the Version String

Displaying the Version String

Use

This function is used to display all the versions and variants for a document.

Features

On the tab page *Document information* in tabstrip controls in the lower left part of the screen, the versions (content versions) and variants (language variants) that exist for a document are displayed:



The same context menu is active for the documents in the version string as for the documents in the display tree.

The following information is displayed on the tab page:

- Descriptive text
- Date created
- Variant ID

Activities

In the upper left part of the screen, select the document for which you want the version string to be displayed and choose the tab page *Document information* in the lower left part of the screen.

Copying Documents

Use

This function enables you to copy documents from both the display tree and the version tree.



Documents stored using SAP ArchiveLink cannot currently be copied.

Features

When you copy a document, a new document, that is, a new version, is always created.

You can specify a new descriptive text and new keywords. Alternatively, you can keep those of the original document.

Activities

Select the document that you want to copy and choose *Copy document* in the symbol toolbar.

Deleting Documents

Deleting Documents

Use

This function enables you to delete one or more documents from the Business Object Navigator.



A document stored using SAP ArchiveLink is deleted according to the existing scenario for SAP ArchiveLink (see also [Displaying Link Entries for Stored Documents \[Page 146\]](#)).

If it is a document stored using SAP ArchiveLink that has been copied in the BDS, the entry in the BDS tables is deleted first. Then the comment that the document was copied in the BDS is deleted from the SAP ArchiveLink link table. The system then queries whether the document is also to be deleted from the SAP ArchiveLink link table.

Features

Documents stored using SAP ArchiveLink or the Business Document Navigator can be deleted.

To delete several documents at once, you can select the documents using the checkboxes in the display tree and on the tab page *Document information* (in tabstrip control).

If a document for which versions or variants exist is selected in the display tree for deletion, the system queries whether the versions and variants of the document are also to be deleted.



As of Release 4.6C, the system queries whether a transport request is to be created when documents are deleted (see also [Transporting Documents \[Page 39\]](#)).

Activities

Select the document(s) that you want to delete and choose *Delete document*.

Editing Documents

Use

This function is used to edit documents created and stored in the Business Document Navigator.



You cannot edit a pure SAP ArchiveLink document.

This function is used for documents that can be edited with Desktop Office Integration.

- You can overwrite the current document
- You can create a new document
- You can create a new version (content version) or variant (language variant)

Prerequisites

The MIME type of the document to be edited belongs to an application that is supported by SAP Desktop Office Integration. See also [BC - Desktop Office Integration \[Ext.\]](#).

Features

If the function *Edit document* is called for a selected document, the relevant application is started in the GUI and the document can be edited further. When you save the document, the following four functions are available:

- Store the edited document as a new document
- Overwrite the stored document with the current document
- Create a new content version for the current version
- Create a new language variant for the current version

In all four cases you can change the descriptive text and the keywords.



As of Release 4.6C, documents are locked when changes are being made. This prevents parallel editing of documents.

Activities

Select a document that you want to edit and choose *Edit document* in the symbol toolbar.

Changing Attributes

Changing Attributes

Use

This function is used to change the descriptive text, language and keywords for a document.

Features

You can change the following attributes:

- Descriptive text
- Language
- Keywords



You can use this function to change attributes of SAP ArchiveLink documents manually.

Activities

Select a document that you want to edit and choose *Change attributes* in the symbol toolbar. When the function is called, a dialog box appears in which you can change the relevant texts and/or language codes.

Creating Annotations

Use

This function is used to create annotations for documents that are displayed in the EAI viewer (see also [Viewer for Displaying Original Files in Standard \[Ext.\]](#)).

Prerequisites

The documents for which annotations are to be created are displayed using the EAI viewer.

Features

When the document is called, the system checks whether annotations already exist for this document. If annotations exist, the menu option *Layer* is displayed in the EAI viewer toolbar. All existing annotations are listed under this menu option and can be displayed.

Various tools for editing are available in editing mode (*Redlining*). New annotations can be created and existing annotations deleted in editing mode.

Activities

Select a document that you want to edit and display it by double-clicking on it.

- To display existing annotations, select the required annotation under the menu option *Layer* in the viewer.
 - The annotation is displayed in the original.
- To edit annotations, choose *Tools* → *Redlining* in the viewer.
 - A new toolbar with various editing tools is displayed.
 - You can edit the annotation.
 - You can delete the annotation.
 - You can create a new annotation.
- If there are no annotations for a document, the menu option *Layer* is not displayed when the document is called. To create a new annotation, choose *Tools* → *Redlining* in the viewer.
 - Before the toolbar is displayed, specify a name for the new annotation.
- After editing, save your annotations by choosing the icon  in the EAI viewer standard toolbar.

Refreshing the Hit List

Use

This function restructures the hit list of stored documents.

Features

The hit list is restructured when the function is called. If documents have been stored (by another user, for example) in the meantime, these are displayed afterwards in the display tree.



The display tree is displayed in compressed form after the function has been called.

Activities

Choose the icon *Refresh hit list* in the icon toolbar.

Navigation to BOR Objects

Use

This function is used for documents that are stored in the Business Object Repository for several instances of an object in order to navigate directly from the BDN hit list to the corresponding object in the BOR (BOR, see also [Business Object Repository \[Ext.\]](#)).

Prerequisites

The BOR objects have implemented the method `Display`.

Activities

In the navigation tree, choose the required document by double-clicking on it.

Exporting Documents

Exporting Documents

Use

This function is used to download documents and their components to the user's local frontend. The user can select the relevant target directory before the export.

Activities

Select the document that you want to export and choose the icon  in the standard toolbar.

Transporting Documents

Use

This function is used to transport documents created and stored in the Business Document Navigator.



Documents stored using SAP ArchiveLink cannot be transported.

Prerequisites

The system settings for the application class to which the documents belong provide for the transporting of documents.

Features

The transport connection exists for documents from both the display tree and the version tree.

To create a transport request for several documents at once, you can select the documents using the checkboxes in the display tree and on the tab page *Document information* (in tabstrip control). A transport request is created for all selected documents.

If a document for which versions and variants exist is selected from the display tree, the system queries whether the versions and variants of the document are also to be transported.



As of Release 4.6C, the system queries whether a transport request is to be created when documents are deleted (see also [Deleting Documents \[Page 32\]](#)).

Activities

Select the document that you want to transport and choose *Transport document* in the symbol toolbar.

Terms

Terms

In this section, terms central to the SAP ArchiveLink environment are explained:

- [Documents \[Page 42\]](#)
- [Document types \[Page 43\]](#)
- [Technical document classes \[Page 45\]](#)
- [Object types and methods \[Page 46\]](#)
- [Business objects / objects \[Page 49\]](#)
- [Synchronous and asynchronous storing \[Page 50\]](#)



The following terms, which have appeared in SAP ArchiveLink documentation for earlier releases are no longer used and have been replaced by the terms shown.

Terms

| New | Old |
|---|--|
| Store | Archive |
| Storing for Subsequent Entry | Early archiving using SAP Business Workflow |
| Storing and Entering | Simultaneous archiving |
| Assigning then Storing | Late archiving |
| Stored documents | Archive documents, archive objects, archived R/3 documents archived documents |
| Incoming documents | Incoming (original) R/3 documents, NCI documents |
| Outgoing documents | Outgoing (original) R/3 documents, CI documents |
| Business objects / objects | R/3 documents, SAP documents, Application documents, Application objects, SAP objects |
| Archive files | Reorganization data, reorganization files, archiving data, archiving files |
| External content server content server | Optical archive, archive system external archive archive |
| Link information | Link entry |
| Document ID | Archive document ID |

Documents

Documents

The term **document** is used in SAP ArchiveLink to mean a series of logically connected pages. This also applies to very long texts.

Documents can also be archive files and desktop files.



- All the pages of a letter together form a document.
- If a stored invoice consists of only one page, then that page is a document.
- A print list with 5,000 pages is also a document.
- A desktop file displayed on the frontend by an external viewer program is also a document.
- Archive files transferred to other R/3 application components, also count as documents.

Document Types

The term **document type** is used to classify more precisely documents to be stored.



Incoming invoices (which belong to document types group "incoming documents") can belong to the following document types:

- Incoming invoices without invoice verification (FIIINVOICE)
- Incoming credit memos without invoice verification (FIICREDIT)

The following groups of document types exist; the grouping takes the technical document class into account:

- **Incoming documents**

These documents (for example, incoming invoices) may be read by a scanner. The information is then stored in a raster image. The same applies to incoming faxes.

Incoming documents can also be in electronic form already.

- **Outgoing documents**

These documents (outgoing invoices for example) are generated in the R/3 System, usually printed and can be processed automatically.

- **Print lists**

These documents can be ABAP lists or screen lists. They are generated in the R/3 System and can be processed automatically.

- **Archive files**

These documents are database extracts, are generated in the R/3 System and can be processed automatically.

- **Binary files**

These documents consist of data originating in the R/3 System.

- **Desktop files**

These documents are application data, from MS Word or MS Excel, for example.

Storing Documents in the Content Server

When a print list is stored in a content server, it is supplemented by a description file provided by SAP. Access to sub-objects is possible and depends on concrete indexing.

The description file

- may be empty.
- may contain information that indexes documents, thereby enabling sub-documents to be accessed.
- may contain other index information.

Document Types



From a print list that contains the open item account balance audit trail, it is possible to access individual customers using appropriate indexing. See also [Indexing Print Lists \[Page 221\]](#).

Displaying Stored Documents

The following cases need to be distinguished when displaying stored documents:

- **Incoming and outgoing documents**

These documents are always displayed individually.

- **Print lists**

The entire list (using the search facility from the display window) as well as a sub-object from the list can be displayed.

- **Archive files**

These documents cannot be displayed using SAP ArchiveLink. The content server can make an archive file available as a whole or as an individual object.

For more information on the archiving of application data, see the documentation [Archiving and Deleting Application Data \[Ext.\] \[Ext.\]](#).

- **Binary files**

These documents are not interpreted by SAP ArchiveLink; they are transferred to another R/3 application component for processing.

- **Desktop files**

These documents are usually displayed by external viewer programs on the frontend.

Technical Document Classes

SAP ArchiveLink manages document types, to each of which is assigned a **technical document class**. The technical document class denotes the storage format for the document.

Different viewers interpret different storage formats, enabling stored documents to be displayed.



- In the standard system, ALF format is displayed directly in R/3.
- PDF format can be displayed using the Adobe Acrobat Reader.
- DOC format can be displayed using Microsoft Word.

Examples of technical document classes

| Document Types | Technical Document Class |
|--------------------|---------------------------------------|
| Incoming Documents | FAX (format for scanned documents) |
| Outgoing documents | PDF (Portable Document Format) |
| Print lists | ALF (Advanced List Format) |
| Archive files | REO |
| Binary files | BIN |
| Desktop files | DOC, XLS and so on. |

Object Types and Methods

Object types are generic descriptions of an object, created at the time of definition. Specific **methods** are defined for the object types. The methods determine the operations that can be executed on an object and usually refer to existing ABAP functions. They are called via a standard interface, which is determined essentially via the method name and the method parameters. The actual implementation of the methods is not taken into account, it is not visible and does not have to be known to the program that calls the method.

The definition of methods in the described manner makes it possible to access the same functions for different objects under the same method name and with the same parameter interface.

Document types are stored and processed in object type groups.



The object type BKPF (accounting document) is defined in the application component FI (Financial Accounting).

The following methods are specified for accounting documents:

- Parameter
SAP ArchiveLink parameters (inherited method)
- ArchivedDocsDisplay
Display stored documents (inherited method)
- BarcodeCapture
Assign object bar code (inherited method)
- Confirm
Confirm
- Create
Create
- Display
Display
- Edit
Edit
- ExistenceCheck
ExistenceCheck

The document types FIIINVOICE (incoming invoice without invoice verification) and FIILETTER (incoming invoice letter), among others, are assigned to this object type.

Object Types and Methods

Examples of Object Types

Business object types are object types chosen with regard to R/3 application components, while technical object types are terms chosen abstractly.

Business object types

- { }-• Documents Incoming invoices, purchase requisitions, applications, ...
- { }-• Master data Customer, material, supplier, ...
- { }-• Transaction data Request, quotation, ...

Technical object types

- { }-• ABAP program
- { }-• Object type
- { }-• Object type method

Object Type Definition

Object types are described and implemented in object type definition via specification of their components:

- Basic data
- Key fields
- Attributes
- Methods with parameters, results and exceptions
- Events with parameters
- Implementation in the program

The term **object type** is an SAP term.

A complete directory of all object types for all clients is provided by the **Business Object Repository (BOR)**, supplied by SAP in the standard system.

Business Object Repository

The **Business Object Repository** is a directory of all Workflow object types in a hierarchical arrangement. The object types are each assigned to an area. The hierarchical structure of these areas is based on the R/3 application hierarchy.

Object Types and Methods

To go to the BOR, choose

Tools → SAP Business Workflow → Development,

Definition tools → Business Object Builder

Utilities → Business Object Repository.

For more information on objects, object types and the corresponding methods and so on, see F1 Help and the application help in the BOR and in the documentation *SAP Business Workflow*.

Business Objects / Objects

A [business object \[Ext.\]](#) represents an actual object in the real business world. It describes an integral business context. This encapsulation leads to simplification, because the inner structuring of a business object remains hidden in additional nested objects. Business APIs as methods of business objects enable external systems to access R/3 System functions, via the Internet, for example.

Synchronous and Asynchronous Storing

Synchronous and Asynchronous Storing

During storing (that is, the transferal of a document to a content server) two cases are distinguished:

- **Synchronous storing**

In the case of documents that are stored synchronously, SAP ArchiveLink waits for the storing process to be completed.

- **Asynchronous storing**

Storing large documents may take some time:

If this happens, SAP ArchiveLink sends an asynchronous storage request to the content server. This request is confirmed immediately by the content server. Storing then takes place outside the R/3 System. After the storage request has been completed, the content server returns:

- the document ID, if storing was successful.
- an error message, if storing was unsuccessful.

Since SAP ArchiveLink does not wait for storing to finish, but instead continues immediately after the request confirmation from the content server, this is referred to as asynchronous storing. Documents stored asynchronously, such as print lists, are therefore not accessible immediately after the storage request is sent, but only once storing has been completed.

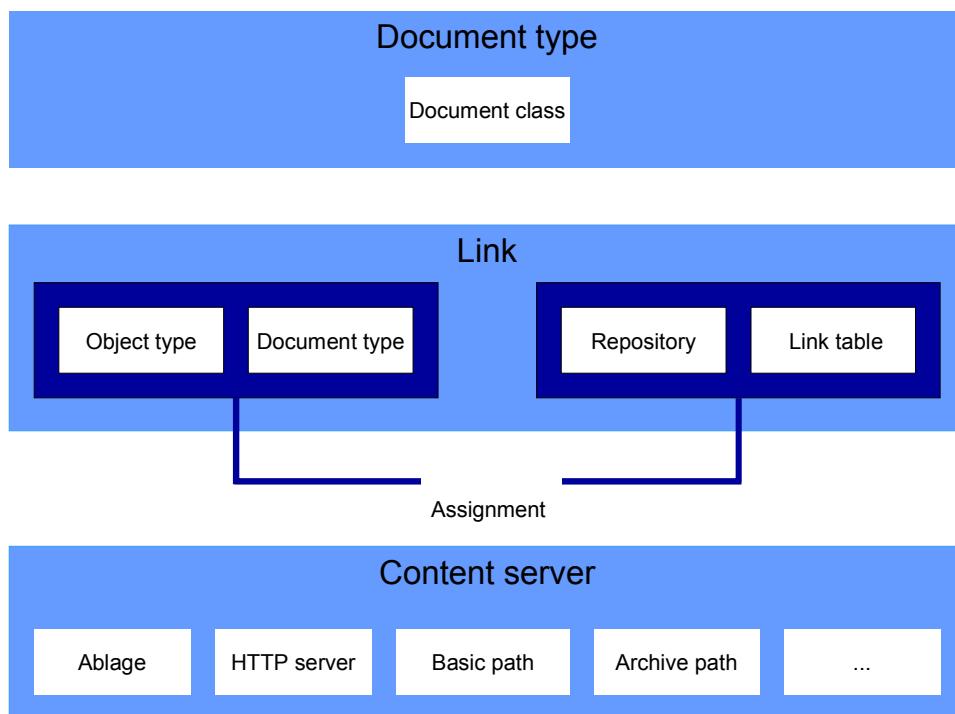
Administration Concept

Documents stored in a content server must have a unique reference to a business object generated in the R/3 application component.

To create the link between the business objects in the R/3 application components and the documents in the content server, the following administrative tasks need to be performed in the R/3 System:

- Definition of document types
- Definition of content servers
- Assignment of
 - document types and object types
 - to
 - content servers and link tables

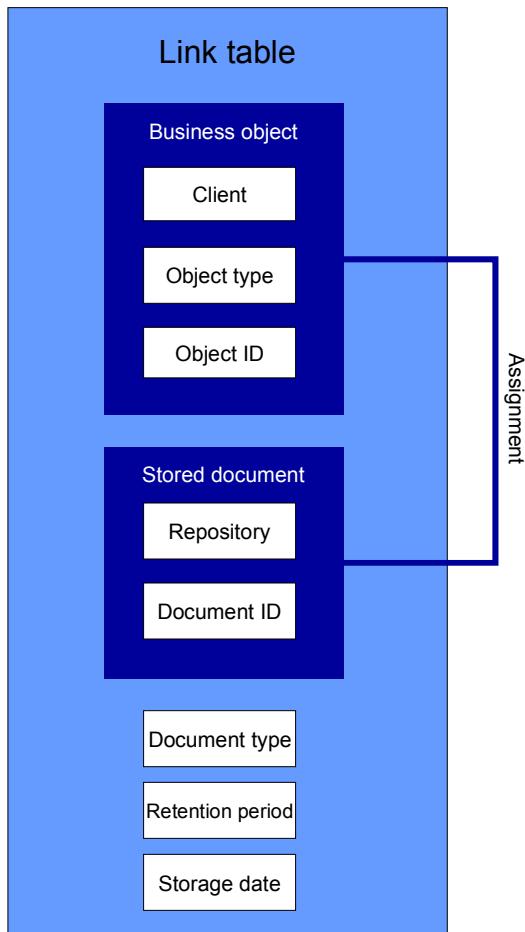
Administration tasks



The respective business object is assigned to the relevant stored document in the link table in the database at runtime.

Administration Concept

Database



There are various customizing functions available for performing definition, assignment, and linking. See the section [Customizing \[Page 165\]](#). The table tasks are explained in the following sections:

- [Content Servers \[Page 53\]](#)
- [Links \[Page 54\]](#)
- [Link Tables \[Page 55\]](#)

Content Servers

The external content servers connected to the R/3 System are specified in the table of content servers. Content servers are identified by a two-digit identification number.

Communication takes place via the specified communication protocol. The content server can access a directory within the R/3 System via the basic path. There is further information that you need to maintain in this table.

You can define content servers at the following places in the system:

- In the Implementation Guide (IMG), choose "R/3", "Basis", "Basis Services", "SAP ArchiveLink", "Basic Settings", "Maintain content servers"
or
- Choose *Tools → Business documents, Basic Settings → Content servers*

For information on maintenance, see the online help in the IMG.

Links

Links

The following link entries are specified in the link table (entries should be made in accordance with the business object /object and the document type):

- the content server in which the document is to be stored
- the link table in which the reference between the (business) object and the corresponding stored document is managed

The active external content server is also specified. Maintenance of this table is a prerequisite for using the link tables.

You can maintain links in the Implementation Guide (IMG):

Choose "R/3", "Basis", "Basis Services", "SAP ArchiveLink", "Basic settings", "Specify links".

For information on maintenance, see the online help in the IMG.

For information about the link tables supplied in the standard system, see the section [Link Tables \[Page 55\]](#).



The term object type describes the general class, while business objects / objects are concrete specific instances.

Link Tables

The link tables are the administration tables in which references between business objects and stored documents are created at runtime. The reference consists of a client-specific SAP object identification and a document identification (see [Administration Concept \[Page 51\]](#)).

The **object identification** of the business object consists of:

- Client
- Object type
- Object ID

The **document identification** of the stored document assigned to the business object consists of:

- Content server (unique ID of content server)
- Document ID (unique ID that the content server assigns to a stored document)

The concept of **link tables** enables R/3 application components to store documents using their own object ID and to access the stored documents again via this object ID. This means that the application components do not have to manage the reference to the stored document themselves. This task is handled by SAP ArchiveLink.

This concept also allows multiple references, such as the assignment of a stored document to several business objects.

Four link tables are supplied in the standard SAP system.

- **TOA01**
- **TOA02**
- **TOA03**
- **TOAHR**

This table is reserved for documents from the R/3 application components PA (personnel administration) and PY (payroll).

- **TOADL**

This table cannot be defined. It is always used for managing print lists.



The various link tables TOA01 to TOA03, supplied in the standard system, optimize access times.

You can define additional link tables in the R/3 System. These must have the dictionary structure TOAV0.

Incoming Documents

Incoming Documents

This section describes the functions for incoming documents. An overview of frontend communication is given and the various storage strategies and the use of bar codes are described.

Information is available on the following topics:

- [Frontend Communication \[Page 57\]](#)
- [Storage Strategies \[Page 59\]](#)
- [Bar Codes \[Page 82\]](#)

Incoming and outgoing documents can be displayed and searched for in the following ways:

- Displaying and searching from the R/3 application component via the appropriate business object (see [Searching for Incoming Documents \[Page 91\]](#)).
- Displaying and searching directly in SAP ArchiveLink Administration (see [Searching for Incoming and Outgoing Documents \[Page 144\]](#)).

Frontend Communication

Frontend components can be integrated into the R/3 System in various ways. Since SAP ArchiveLink serves as an open interface to external components, and so also to frontend components, it is in principle possible to integrate frontend applications into the R/3 System. For example:

- Scanning components
- Content server viewers
- Desktop programs

A file generated at a work center can be stored, and displayed again subsequently.

Incoming documents can be:

- Stored regardless of their form
 - Scanned documents
 - Incoming faxes
 - Files from work centers, that is, desktop files
- Displayed and edited, according to their type
 - Scanned documents, using viewers for the specific document format
 - Desktop files, using the appropriate application

Faxes and Business Workplace Documents

Business Workplace Documents

Business Workplace documents of document class ARC can be stored using SAP ArchiveLink.



When creating a document of document class ARC in the Business Workplace, you must specify a document type belonging to object type SOOD. As of R/3 Release 3.1G, only those document types maintained in Customizing are proposed.

For more information on documents in the Business Workplace, see [Business Workplace \[Ext.\]](#).

Faxes

Incoming faxes can be displayed, edited in the Business Workplace and processed/stored.

- A link to an incoming fax is created in the Business Workplace by the SAPcomm or SAPconnect interface and sent to the inbox.
- When the fax object is processed in the inbox, the fax file is stored, and further processing performed as in the storage scenario "storing for subsequent entry".



The fax board/box supplier must support the SAP incoming fax interface.

Frontend Communication

Faxes can arrive in two ways:

- **Via SAPcomm**

The advantage of this is that the naming conventions for faxes are supported: several files exist for a multi-page fax.

The disadvantage is that the files are only stored in the file system.

- **Via SAPconnect**

The advantage of this is that the files can be stored in the file system or in the database. In addition, not only documents of class FAX or TIF (not Multipage-TIFF, however), but those of any document class are supported.

For further information on FAX documents in the Business Workplace, see [Business Workplace \[Ext.\]](#).

Communication Protocols

Frontend applications are addressed via various communication protocols, depending on platform and capability:

- OLE Automation 2.0, on Windows NT, for storing or displaying a document and the associated functions
- Execute, on Windows NT, for displaying a document

For further information about communication, that is protocols and applications, see section [Communication \[Page 301\]](#).

Storage Strategies

This section explains the various procedures that can be used when scanning and storing incoming documents. There is also a discussion of the advantages and disadvantages of the various storage strategies.

The following strategies for storing incoming documents are distinguished:

- Storing for Subsequent Entry
- Storing and Entering
- Storing for Subsequent Assignment
- Storing and Assigning
- Assigning then Storing
- [Early Storing with Bar Codes \[Page 78\]](#)

At the start of processing in early storing with bar codes, the incoming document is assigned a bar code, scanned and stored. The document is then entered in the R/3 System.

This means that the incoming document is stored **before** the business object is entered.

- [Late Storing with Bar Codes \[Page 79\]](#)

At the start of processing in late storing with bar codes, the incoming document is assigned a bar code. The document is not scanned and stored until after it is entered in the R/3 System.

This means that the incoming document is stored **after** the business object is entered.

SAP ArchiveLink also lets you transfer documents already stored in other business application components to your own R/3 application. You do this from the display function and it is described in the section [Transfer to R/3 Application Components \[Page 86\]](#).

Advantages and Disadvantages of the Storage Strategies

This section compares the advantages and disadvantages of the various storage strategies.

Storing for Subsequent Entry

Advantages:

- The ability to process documents centrally and hence efficiently and economically, using SAP Business Workflow. Documents in paper form are therefore removed from the process flow very early on. This affords time as well as cost advantages, since conventional mail distribution within the company is not required and faster processing is therefore possible.
- Documents enter an electronic process, where the electronic document can undergo numerous processing steps controlled by the computer.

Disadvantages:

- The initial high costs associated with the necessary purchase of high-resolution screens in the application departments and possible reorganization of the mailroom.

Storage Strategies

Storing and Entering

Advantages:

- When processing specially protected documents, for example in Human Resource Management, the person responsible can process and store the documents at their work center simultaneously.
- This type of storing is useful for technical documents where attributes are assigned immediately (such as in the Document Management System) or when making corrections to, or re-entering documents.

Disadvantages:

- When a large number of documents is processed, the constant switch between document storing and document processing (entering the business object) interferes with the workflow and therefore reduces efficiency.
- This type of storing is generally not suitable for storing en masse.

Storing for Subsequent Assignment/Storing and Assigning

Advantages:

- The ability to process documents centrally and hence efficiently and economically, using SAP Business Workflow.
- Documents enter an electronic process, where the electronic document can undergo numerous processing steps controlled by the computer.

Assigning then Storing

The **advantages** are mainly organizational, and are as follows:

- The process flow and paper flow occur in the conventional way. There are considerably fewer organizational changes, with the resulting time and psychological problems, than is the case with the other storage strategies.
- Paper documents, such as invoices, can be checked and countersigned before they are entered in the system. The countersignatures are stored along with the document.

Disadvantages:

- The optimization of business processes using SAP Business Workflow is not possible. Automatic agent assignment is therefore not possible either.

Storing with Bar Codes

The **advantages** of early and late storing with bar codes are as follows:

- The process flow and paper flow occur in the conventional way. There are considerably fewer organizational changes, with the resulting time and psychological problems, than is the case with the other storage strategies.
- The purchase of high-resolution screens in the application departments is not necessary.
- Classification according to document type is not required.

The **disadvantages** of storing with bar codes are as follows:

Storage Strategies

- The optimization of business processes using SAP Business Workflow is not possible. Automatic agent assignment is therefore not possible either.
- Incoming documents are circulated in paper form.

Concept: Storing for Subsequent Entry

The scenario **storing for subsequent entry** is used if you want to separate (organizationally and in terms of time) the storing of incoming documents from the actual processing, that is, the entering of the business object.



The document is removed from the normal process flow in the mailroom or the PC fax inbox, that is, at a very early stage. Initial costs are incurred as a result of the purchase of high-resolution screens in the application departments and possible reorganization of the mailroom,

The early removal of documents in paper form from the process flow provides the basis for significant cost-saving and optimization measures.

Incoming documents are processed using **SAP Business Workflow**.

A **task** can be used to maintain Workflow document types. The following **standard tasks** are available for storing for subsequent entry:

- TS30001128
- TS00007869

Standard tasks are single-step tasks provided by SAP covering elementary business activities from an organizational viewpoint. A single-step task refers to a **single** object method (technical link to R/3 functionality) and is associated with those persons who can process the object.



You can also incorporate workflow templates that you have defined yourself.

You can define these workflow templates using the SAP ArchiveLink Workflow Wizard as of R/3 Release 4.0A (see [SAP ArchiveLink Workflow Wizard \[Page 80\]](#)).

For further information, see the documentation *SAP Business Workflow*, section [Tasks and Task Groups \[Ext.\]](#).

Default Settings

You need to define default settings in order to define the various storage scenarios. Using default settings, you can combine several document types into groups with user-defined names.



The same document type can occur in several default settings and if different recipients are defined each time, the same document type can occur several times within one default setting.

Recipients are maintained for the document types. A recipient consists of an **object type** (for example, user, position, work center or organizational unit) and an **organizational object** (for example, the person responsible for processing of the relevant document type). The assignment of an organizational object and object type to a document type ensures that only those agents who are actually responsible are selected from among the possible agents. If the recipient **us Mustermann** is specified, then the work item is only displayed in his or her inbox. In complex business processes, is it advisable to define an item, a work center, or an organizational unit

Concept: Storing for Subsequent Entry

(instead of a user) as the recipient. You define the recipients by specifying the object type and organizational object in the **organizational structure** of the company.

The assignment of the recipient to a document type is dependent on the role resolution for the respective task: The object type and organizational object are transferred to the role container.

Standard Task TS30001128



The standard task TS30001128 is available as of R/3 Release 4.5A.

With standard task TS30001128, input data is specified for all mandatory elements of the task container at start up. (The image, the recipient, the document type, the document class and the long description.) During processing, the object IDs of the business objects generated are put into the container. TS30001128 is therefore very similar to the old TS00007869. The standard task functions have, however, been enhanced. In the new storage scenarios for R/3 Release 4.5A, it is possible to switch between the different scenarios, for example, from entering to assigning.



As of R/3 Release 4.5A, it is recommended that you use the new standard task TS30001128, instead of TS00007869 (see below). In the long term, standard task TS30001128 will replace standard task TS00007869.

Standard Task TS00007869

With standard task TS00007869, input data is specified for all mandatory elements of the task container at start up.



This standard task covers the same SAP Business Workflow 3.0 functions as the previous standard process type NCI_INPUT.

For R/3 Release 3.0, task TS00007869 is configured in the delivered customer system, but deactivated.

For R/3 Release 3.1G, this task is delivered activated.

As of Release 4.5A, SAP recommends the use of the new standard task TS30001128 (see above) instead of TS00007869.

Procedure: Storing for Subsequent Entry

SAP recommends that you use SAP Business Workflow 3.0/4.0. The following information is specific to SAP Business Workflow 3.0/4.0.

You can also use a workflow template that you have designed yourself, for storing for subsequent entry. You can generate this workflow template using the SAP ArchiveLink Workflow Wizard.

The SAP ArchiveLink Workflow Wizard is in the system under
Tools → *Business documents*,
Basic Settings → *Customizing* → *Workflow Wizard*.

For more information, see the online documentation.

The workflow and information flow for *storing for subsequent entry* and the standard task TS30001128 is as follows:

1. The incoming document can be in paper or electronic form.
 - If the incoming document is in paper form, the paper pages are grouped into documents and scanned by an employee in the mailroom. Documents are then checked for quality and completeness. The scanned documents are displayed in the scan dialog window. The scan entry can be replaced by a FAX entry in the Business Workplace.
 - If the incoming document is in electronic form, it may be an MS Word document in the system.
2. The employee responsible for assignment selects a setting in the R/3 window (FI documents, for example) via the store function in the *Office* menu. For presorted documents, this only needs to be done once per document stack or document type.
3. The person responsible then assigns a document type to the displayed incoming document using active 3.0 task (TS00007869). Several document types (incoming invoice without invoice verification, incoming credit memo without invoice verification, for example) are defined for each input area (for example, financial accounting). This allows several different document types to be stored and precisely assigned.
4. The person responsible confirms the assignment. This starts the workflow task and generates a work item, which is put into the worklist of the person responsible within the application.
If more than one person are responsible for this document type, they will all see the work item in their worklist. Only one of them has to process the work item. The persons responsible are determined by agent assignment for the task, role resolution (standard role 126) in SAP Business Workflow and the assignment of an organizational object to a document type in the default setting.

For more information on agent assignment and role resolution, see the documentation *SAP Business Workflow* in sections [Maintaining Agent Assignment \[Ext.\]](#) and [role resolution](#).

Procedure: Storing for Subsequent Entry

5. The person responsible within the application executes the work item. He or she can then choose between the following functions:

- *Process*

This entails displaying the image in the relevant viewer and calling an R/3 application component.

- *Change*

This entails changing the workflow document type because of an incorrect previous classification. The existing old work item is deleted and a new one generated. The new work item is sent to the persons responsible.

- *Create*

This entails starting a task with an additional workflow document type, that is, the existing work item continues to exist and an additional work item is generated.



Multiple processing of documents when storing for subsequent entry should also be mentioned:

Several different document types can be assigned to a document during subsequent processing, thereby providing for additional processing steps in the workflow. This allows situations to be handled in which various information (and the necessary processing steps) is available in one incoming document. For example, an incoming invoice may also contain a dunning notice or notice of an address change.

- *Cancel*

Execution of the work item is canceled.

6. When the person responsible processes the work item, the application method (for example, call of an application transaction) provided for this document type in Workflow is called automatically. The person responsible creates a business object.
7. After the business object has been created, the business object and stored document are assigned to one another in a link table. The work item is then deleted from the workflow inbox of the person responsible within the application.

See also:

Procedure: [Storing for Subsequent Entry \[Page 94\]](#)

Concept: Storing and Entering

Concept: Storing and Entering

In the storage strategy [storing and entering \[Ext.\]](#), storing is carried out where the incoming documents are processed, that is, the person responsible carries out the storing himself/herself, verifies the image quality, edits the image, if necessary and finally enters the document, classifies it and stores it. If the agent is a permitted agent in SAP Business Workflow, the work item created is executed **immediately**.

Otherwise, *storing and entering* is the same as [storing for subsequent entry \[Page 68\] \[Page 62\]](#).



Storing and entering requires a content server whose stored documents can be accessed **immediately after they have been stored**.

Incoming documents are processed using **SAP Business Workflow**.

The following **standard tasks** are available for storing and entering:

- TS30001128
- TS00007869

Standard tasks are single-step tasks provided by SAP covering elementary business activities from an organizational viewpoint. A single-step task refers to a **single** object method (technical link to R/3 functionality) and is associated with those persons who can process the object.



You can also incorporate workflow templates that you have defined yourself.

You can define these workflow templates yourself as of R/3 Release 4.0A using the SAP ArchiveLink Workflow Wizard (see [SAP ArchiveLink Workflow Wizard \[Page 80\]](#)).

For further information, see the documentation *SAP Business Workflow*, section [Tasks and Task Groups \[Ext.\]](#).

For more information on default settings, organizational plans and the standard tasks TS30001128 and TS00007869, see: [Concept: Storing for Subsequent Entry \[Page 62\]](#).

Process Flow: Storing and Entering

The workflow and information flow in the storage scenario [storing and entering \[Ext.\]](#) is very similar to that in the storage scenario [storing for subsequent entry \[Page 64\]](#), but with the following differences:

- Documents are entered by the agent himself or herself and not by an employee from the mailroom. The agent therefore carries out all processing steps from scanning (if necessary) via classifying, to storing and entering.
- If the agent is a permitted agent in SAP Business Workflow, the work item created is executed **immediately**.



You can also use a user-defined workflow template for storing and entering. You can generate the workflow template using the SAP ArchiveLink Workflow Wizard.

The SAP ArchiveLink Workflow Wizard is in the system under
Tools → *Business documents*,
Basic Settings → *Customizing* → *Workflow Wizard*.

For more information, see the online documentation.

See also:

Procedure: [Storing and Entering \[Page 98\]](#).

Concept: Storing for Subsequent Assignment

Concept: Storing for Subsequent Assignment



The storage scenario **storing for subsequent assignment** is available as of R/3 Release 4.5A.

In *storing for subsequent assignment*, the incoming document relates to an existing business object. The incoming document is scanned, stored and forwarded directly, using SAP Business Workflow, to the employee responsible.

In contrast to [storing for subsequent entry \[Page 62\]](#), the business object already exists and only assignment needs to be performed.

Storing for subsequent assignment is similar to *assigning then storing* with the difference that SAP Business Workflow is involved.

The following **standard task** is available for *storing for subsequent assignment*:

- TS30001117

Standard tasks are single-step tasks provided by SAP covering elementary business activities from an organizational viewpoint. A single-step task refers to a **single** object method (technical link to R/3 functionality) and is associated with those persons who can process the object.



You can also incorporate workflow templates that you have defined yourself.

You can define these workflow templates using the SAP ArchiveLink Workflow Wizard as of R/3 Release 4.0A (see [SAP ArchiveLink Workflow Wizard \[Page 80\]](#)).

For further information, see the documentation *SAP Business Workflow*, section [Tasks and Task Groups \[Ext.\]](#).

Default Settings

You need to define default settings in order to define the various storage scenarios. Using default settings, you can combine several document types into groups with user-defined names.



The same document type can occur in several default settings and if different recipients are defined each time, the same document type can occur several times within one default setting.

Recipients are maintained for the document types. A recipient consists of an **object type** (for example, user, position, work center or organizational unit) and an **organizational object** (for example, the person responsible for processing of the relevant document type). The assignment of an organizational object and object type to a document type ensures that only those agents who are actually responsible are selected from among the possible agents. If the recipient **US Mustermann** is specified, then the work item is only displayed in his or her inbox. In complex business processes, is it advisable to define an item, a work center, or an organizational unit (instead of a user) as the recipient. You define the recipients by specifying the object type and organizational object in the **organizational structure** of the company.

Concept: Storing for Subsequent Assignment

The assignment of the recipient to a document type is dependent on the role resolution for the respective task: The object type and organizational object are transferred to the role container.

Standard Task TS30001117

The standard task TS30001117 is available as of R/3 Release 4.5A.

It enables storing for subsequent assignment for practically all BOR object types.

In the new storage scenarios for R/3 Release 4.5A, it is possible to switch between the different scenarios, for example, from entering to assigning.

Procedure: Storing for Subsequent Assignment

You can use a user-defined workflow template when storing for subsequent assignment. You can generate this workflow template using the SAP ArchiveLink Workflow Wizard.

The SAP ArchiveLink Workflow Wizard is in the system under *Tools* → *Business documents*, *Basic Settings* → *Customizing* → *Workflow Wizard*.

For more information, see the online documentation.

The workflow and information flow for *storing for subsequent assignment* and the standard task TS30001117 is as follows:

1. An incoming document relates to an existing business object in an R/3 application component.
2. The incoming document is forwarded to the storage location.
3. In the storage location, the employee responsible scans the document. Documents are then checked for quality and completeness. The scanned documents are displayed in the scan dialog window.

If the incoming document is in electronic form, as an MS Excel document for example, the scan-dialog step is not necessary.

It is also possible to use the same document as during the previous session. If you want to do this, multiple assignment must be activated in Customizing for [basic settings \[Page 168\]](#).

Several document types can be assigned to a document during processing, thereby providing for additional processing steps in the workflow. This allows situations to be handled in which various information (and the necessary processing steps) is available in one incoming document. For example, an incoming invoice may also contain a dunning notice or notice of an address change.

4. The employee responsible for assignment selects a setting in the R/3 window (FI documents, for example) via the store function in the *Office* menu. For presorted documents, this only needs to be done once per document stack or document type.
5. The person responsible then assigns a document type to the displayed incoming document using active task (TS30001117). Several document types (incoming invoice without invoice verification, incoming credit memo without invoice verification, for example) are defined for each input area (for example, financial accounting). This allows several different document types to be stored and precisely assigned.
6. The person responsible confirms the assignment. This starts the workflow task and generates a work item, which is put into the worklist of the person responsible within the application.
If more than one person are responsible for this document type, they will all see the work item in their worklist. Only one of them has to process the work item. The persons responsible are determined by agent assignment for the task, role resolution (standard role

Procedure: Storing for Subsequent Assignment

126) in SAP Business Workflow and the assignment of an organizational object to a document type in the default setting.

For more information on agent assignment and role resolution, see the *SAP Business Workflow* documentation, section [Maintaining Agent Assignment \[Ext.\]](#).

7. The person responsible within the application executes the work item. He or she can then choose between the following functions:

- *Continue*

This entails displaying the image in the relevant viewer and calling an R/3 application component. The object ID for the business object is entered.

- *Change*

This entails changing the workflow document type because of an incorrect previous classification. The existing old work item is deleted and a new one generated. The new work item is distributed to the persons responsible.

- *Create*

This entails starting a task with an additional workflow document type, that is, the existing work item continues to exist and an additional work item is generated.

- *Cancel*

Execution of the work item is canceled.

8. When the person responsible processes the work item, the business object and stored document are assigned to one another in a link table. The work item is then deleted from the workflow inbox of the person responsible within the application.

Concept: Storing and Assigning

The storage scenario **storing and assigning** is available as of R/3 Release 4.5A.

In *storing and assigning*, storing and assigning of incoming documents takes place at a work center, that is, the person responsible carries out the storing himself/herself, verifies the image quality, edits the image (if necessary), enters the document, classifies it and assigns it to a business object. If the agent is a permitted agent in SAP Business Workflow, the work item created is executed **immediately**.

Otherwise, *storing and assigning* is the same as [storing for subsequent assignment \[Page 68\]](#).



Storing and assigning requires a content server whose stored documents can be accessed **immediately after they have been stored**.

Incoming documents are processed using **SAP Business Workflow**.

A **task** can be used to maintain Workflow document types. The following **standard task** is available for *storing and assigning*:

- TS30001117

Standard tasks are single-step tasks provided by SAP covering elementary business activities from an organizational viewpoint. A single-step task refers to a **single** object method (technical link to R/3 functionality) and is associated with those persons who can process the object.



You can also incorporate workflow templates that you have defined yourself.

You can define these workflow templates using the SAP ArchiveLink Workflow Wizard as of R/3 Release 4.0A (see [SAP ArchiveLink Workflow Wizard \[Page 80\]](#)).

For further information, see the documentation *SAP Business Workflow*, section [Tasks and Task Groups \[Ext.\]](#).

Procedure: Storing and Assigning

The workflow and information flow in the storage scenario *storing and assigning* is very similar to that in the storage scenario ["storing for subsequent assignment \[Page 70\]"](#), with the following differences:

- Documents are assigned by the person responsible himself or herself and not by another employee. The person responsible therefore carries out all processing steps from scanning (if necessary), via classifying, to storing, entering and assigning.
- If the agent is a permitted agent in SAP Business Workflow, the work item created is executed **immediately**.



You can use a user-defined workflow template when storing and assigning. You can generate the workflow template using the SAP ArchiveLink Workflow Wizard.

The SAP ArchiveLink Workflow Wizard is in the system under
Tools → *Business Documents*,
Basic Settings → *Customizing* → *Workflow Wizard*.

For more information, see the online documentation.

See also:

- [Concept: Storing and Assigning \[Page 72\]](#)

Concept: Assigning then Storing

Concept: Assigning then Storing

In the storage scenario *assigning then storing*, the incoming document relates to an existing business object. Example: the incoming document is scanned and then stored.



No change from the usual processing is necessary for this storage strategy and the incoming document travels around the firm in paper form.

The workflow and organization flow for *assigning then storing* is analogous to that for [storing for subsequent assignment \[Page 68\]](#), except that SAP Business Workflow is **not** involved in [assigning then storing \[Ext.\]](#).

Process Flow: Assigning then Storing

Process Flow: Assigning then Storing

The workflow and information flow for [assigning then storing \[Ext.\]](#) is as follows:

1. An incoming document relates to an existing business object in an R/3 application component.
2. The incoming document is forwarded to the storage location.
3. In the storage location, the employee responsible scans the document. Documents are then checked for quality and completeness. The scanned documents are displayed in the scan dialog window.
If the incoming document is in electronic form, as an MS Excel document for example, the scan dialog step is not necessary.
4. The employee selects a setting in the R/3 window using the store function from the *Office* menu and assigns a document type to the incoming document.
5. He or she then enters the object ID for the relevant business object.
6. In the check window containing the most important data about the business object, the employee checks that the business object entries agree with those of the displayed incoming document, and confirms the assignment.
7. This confirmation assigns the business object and the incoming document to each other in a link table.

See also

- [Concept: Assigning then Storing \[Page 74\]](#)
- [Procedure for assigning then storing \[Page 100\]](#)

Concept: Storing with Bar Codes

Storing with bar codes can be performed in two different ways:

- **Early Storing with Bar Codes**

This storage strategy is useful for separating the storing of incoming documents (in terms of time and organization) from the actual processing which is performed, that is the entering of the business object.

- **Late Storing with Bar Codes**

In this storage strategy, storing takes place after the business object has been entered.

No change from the usual procedure is necessary and the incoming document travels around the firm in paper form.

Early Storing with Bar Codes

In early storing with bar codes, a bar code must be applied to the first page of an incoming document. In early storing, this occurs at the beginning of processing.

Documents are scanned and stored as required. When a document is scanned, the input software (of the content server) automatically recognizes the bar code and transfers it to SAP ArchiveLink together with the document ID.

The documents are now put into circulation in paper form and are therefore updated in the relevant R/3 application transaction also from paper form.

When the paper document is entered and the relevant business object generated, the person responsible has the following options:

- To enter the bar code using a bar code pen or gun. The bar code is then transferred automatically to a dialog box.
- To read the bar code and enter it manually in the dialog box.

SAP ArchiveLink now makes the assignments to the business object.

Late Storing with Bar Codes

For late storing with bar codes, a bar code must be applied to the first page of an incoming document. As with early storing, this occurs at the beginning of processing.

The documents are now put into circulation in paper form and are therefore entered in the relevant R/3 application transaction also from paper form. When the paper document is entered and the relevant business object generated, the person responsible has the following options:

- To enter the bar code using a bar code pen or gun. The bar code is then transferred automatically to a dialog box.
- To read the bar code and enter it manually in the dialog box.

The incoming document is then forwarded to the storage location, in contrast to early storing with bar codes, where storing takes place right at the beginning. When a document is scanned, the input software (of the content server) automatically recognizes the bar code and transfers it to SAP ArchiveLink together with the document ID.

SAP ArchiveLink now makes the assignments to the business object.

Concept: Storing with Bar Codes

This procedure has one step less than [assigning then storing \[Page 74\]](#), thereby reducing the possibility of error.

Process Flow: Early Storing with Bar Codes

The workflow and information flow in **early storing with bar codes** is as follows:

1. The incoming document is assigned a bar code in the mailroom or entry location.
The paper pages are scanned at the entry location. Documents are then checked for quality and completeness.
2. The document is stored. The scanning component recognizes the bar code and reports it back to the R/3 System together with the document ID.
3. The incoming document is processed in paper form in an R/3 application component. For example, a posting procedure is performed and a business object thereby generated. In every posting procedure, the bar code, if it already exists, must also be transferred.
4. SAP ArchiveLink assigns the stored document to the appropriate business object in a link table.

See also

- [Concept: Storing with Bar Codes \[Page 76\]](#)
- [Procedure: Early Storing with Bar Codes \[Page 97\]](#)

Process Flow: Late Storing with Bar Codes

In late storing with bar codes, the process and information flow occurs as follows:

1. The incoming document is assigned a bar code in the mailroom.
2. The incoming document is processed in an R/3 application component, for example, a posting procedure is performed and a business object thus generated. For every posting procedure, the bar code must also be transferred.
3. The incoming document is forwarded to the input location.
4. In the input location, the paper pages are grouped into documents and scanned. Documents are then checked for quality and completeness.
5. The scanning component recognizes the bar code and reports it back to the R/3 System together with the document ID.
6. SAP ArchiveLink assigns the incoming document to the appropriate business object in a link table.

See also

- [Concept: Storing with Bar Codes \[Page 76\]](#)
- Procedure: [Late Storing with Bar Codes \[Page 102\]](#)

SAP ArchiveLink Workflow Wizard

SAP ArchiveLink Workflow Wizard

The SAP ArchiveLink Workflow Wizard leads the user step-by-step through the process for generating a workflow template. At the end of the wizard process, the template is generated using the entries that the user has made during the wizard process. It can be implemented for all storage scenarios that use SAP Business Workflow.

If the SAP ArchiveLink Workflow Wizard is used, the workflow loses its fixed form with regard to storing with SAP ArchiveLink (for example, the standard task TS00007869), and receives instead various optional parts that can be defined individually by the respective user.

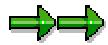


As of R/3 Release 4.5A, this wizard covers the new storage scenarios.

Each screen step has detailed online help.

SAP ArchiveLink document types Customizing Wizard

The SAP ArchiveLink Document Types Customizing Wizard leads the user step-by-step through the process for generating a document type. You can specify document types to be used to generate default values for the Customizing settings. At the end of the wizard process, actual generation is performed using the entries that the user makes during the wizard process. It can be implemented for all storage scenarios.



As of R/3 Release 4.5A, there is a connection to the Change & Transport System.

Each screen step has detailed online help.

Bar codes

This section explains the prerequisites for using bar codes, and various bar code types.

For further information on storing bar codes, see [Concept: Storing with Bar Codes \[Page 76\]](#) [\[Page 76\]](#).

Prerequisites for Use

Bar codes can be used for early and late storing. The prerequisites are:

- The entry component of the content server must support bar codes.
- The R/3 application component must:
 - support bar codes for processing (posting a business object, for example) (see also information on the relevant scenarios in the documentation [SAP ArchiveLink - Scenarios in Applications \[Ext.\] \[Ext.\]](#) or
 - support generic object services
(see also [Object Services \[Page 89\]](#) and [Assigning Bar Codes Generically \[Page 110\]](#)).

Once a document has been scanned, the bar code read correctly, the document stored and an assignment to the business object generated, the bar code had served its purpose, but it remains on the stored document. The bar code can be reused later if required.

Bar code types

Ask your content server supplier which bar codes are supported by the entry software.

The R/3 System accepts a wide variety of bar code types. The standard system provides an automatic check for the following five bar code types:

- **2/5 interleaved**

This is a numeric bar code widely used in industrial applications. It consists of an even number of digits.

- **EAN 13**

This is a bar code that encodes 13 characters: the first two are country codes, followed by 10 data characters and a checksum. This type of bar code is used mainly in the retail sector.

- **EAN 8**

This is a shortened version of the EAN 13 bar code. It includes a two-digit country code, five data characters and a checksum character.

- **UPC A**

This type of bar code is used mainly in the retail sector. The manufacturer and specific product are encoded in the bar code so that cash register systems can automatically find the product price. In the USA, the UPC-A bar code and the assignment of manufacturer IDs is monitored by the Uniform Code Council.

- **UPC E**

Bar codes

This is an altered UPC A bar code based on the number system 0. Since there are no zeros, this bar code takes up very little space, meaning it can be used on very small products.



A check procedure is used that is valid for all five bar code types delivered.

In the check routines delivered, an integrated check sum check procedure is used, which relates only to the **check digits** and not the type.

It is also possible to create a new bar code yourself and maintain a check bar code correspondingly. The check bar code must have the same interface as the standard check function module. For information on the bar code type with the corresponding long description and function module, choose

Tools → Business documents,

Basic settings → Bar Code → Bar code types

to reach the screen "Display View: Bar Code Types: Overview

A check for other bar code types can be easily integrated on a user-specific basis. For more information, see [Early Storing with Bar Codes \[Page 212\]](#) [\[Page 212\]](#)..

Outgoing documents

Outgoing documents

Outgoing documents are documents which are created in R/3 application components, such as invoices, dunning notices and delivery notes. They are normally printed and sent out.

Outgoing documents can be stored in content servers, if

- if they are generated with SAPscript and
- the archiving is supported by the appropriate business application components, (see the documentation [SAP ArchiveLink - Scenarios in Applications \[Ext.\] \[Ext.\]](#)).

Output

You can choose between the following types of storing:

- *Print* (on printer only)
- *Store* (in a content server only) or
- *Print and store* (on printer and in a content server)



As of R/3 Release 4.0B, only document type PDF is used for outbound documents.

Display and Search

The outgoing documents can be accessed directly from the generating R/3 application component.

Incoming and outgoing documents can be displayed and searched for in the following ways:

- Displaying and searching from the R/3 application component via the appropriate business object (see [Searching for Stored Documents \[Page 91\]](#)).
- Displaying and searching directly in SAP ArchiveLink Administration (see [Searching for Incoming and Outgoing Documents \[Page 144\]](#)).

Print lists

The results of a report in the R/3 System are often print lists. Examples of such print lists are:

- Accumulated balance audit trails
- Batch where-used lists
- Project-related service specifications
- Balance sheet valuations
- Cost center reports

If you also want to be able to search for subobjects (for example, individual customers) in the print list (for example, open item account balance audit trail), it must contain **index information**. See the documentation [SAP ArchiveLink - Scenarios in Applications \[Ext.\] \[Ext.\]](#) for which reports and/or print lists are output with index information in the various R/3 application components.

The storing of print lists in a content server is described in detail and with examples for the developer in the section [Overview: Storing Print Lists \[Page 219\]](#).

Links to other stored documents are possible within a print list in the form of **Hyperlinks** (see [Inserting Index Information and Hypertext Links \[Page 227\] \[Page 227\]](#)).



As of R/3 Release 4.5A, you can use hyperlinks to insert object references to any BOR object types, within stored print lists (see [Storing Print Lists \[Page 219\] \[Page 219\]](#)).

Transfer to R/3 Application Components

Transfer to R/3 Application Components

In some cases, a document that has already been stored is to be transferred to another R/3 application component. You can do this from the following places in SAP ArchiveLink:

- from SAP ArchiveLink administration.

This is reserved for the administrator. The procedure is described in [Reassigning Stored Documents \[Page 149\]](#).

- from an R/3 application component.

This is only possible if the R/3 application component supports the direct transfer of image documents. The procedure is described in [Transferring Stored Document \[Page 106\]](#). There is also a description of the transfer to the possible R/3 application components in the documentation [SAP ArchiveLink - Scenarios in Applications \[Ext.\]](#).



See also [Generic Object Services \[Page 89\]](#)

Viewers

Stored documents can be displayed using various viewers. The following viewers are possible:

- Web browser as of R/3 Release 4.5A
- R/3 System (for displaying print lists)
- Document-specific viewers (for example, MS Word or MS Excel)
- Viewers from content server suppliers (that is, external viewers)



You must always use the appropriate viewer for the document class.

Display

Displaying of stored documents is generally activated from the R/3 application component. The viewer called displays the stored document in the display window. The user therefore has two windows on the screen:

- the R/3 window
- the viewer window

You can position the windows individually on the screen, resize them within certain limits, and close them. The display component functions are a little different in UNIX systems and MS Windows systems.

Hyperlinks

Black-underlined or framed sections in sub-objects of a print list indicate hyperlinks. Hyperlinks are references to other documents. You can display the referenced stored document by double-clicking on these hyperlinks. If several stored documents are referenced, double-clicking generates a list from which you can select the required stored document.

Connection to External Viewers

Connection to External Viewers

PC programs that can be started with the transfer of a file which is to be opened, can be accessed from the R/3 System via SAP ArchiveLink. As a result, you can integrate many standard Windows programs.

Windows applications which can be started using OLE 2.0 Automation and which can be transferred to a file for opening via OLE 2.0 Automation can be accessed from the R/3 System via SAP ArchiveLink. The OLE method syntax expected by the application depends on the application and is stored in the SAP ArchiveLink settings. With regard to restrictions, it should be noted that export and import parameters cannot be transferred at the same time with a method call.



If the file name is specified, this file must be accessible locally on the frontend. If a stored document is only referenced via content server and document ID, the application must have access to the content server.

Connection to Generic Object Services

From the business objects implemented in the R/3 applications, four SAP ArchiveLink functions can be executed on a **cross-application** and an **application-independent** basis:

- [Displaying Stored Documents Generically \[Page 107\]](#)

You can display a list of the stored documents for which a link with the corresponding R/3 application business object already exists. From this hit list, you can display the stored document in your viewer or display the relevant link entries.

- [Assigning Documents Generically \[Page 108\]](#)

You can display a list of the stored documents most recently displayed. You can then link the stored documents with more business objects.

- [Storing Documents Generically \[Page 109\]](#)

You can store documents in dialog. Documents (depending on document type) can be linked with business objects in the following ways:

- from a scan dialog
- from an external application
- from the desktop

- [Assigning Bar Codes Generically \[Page 110\]](#)

You can create a bar code for a business object: You can enter a bar code for a document type, which will be used later when the business object is linked to the stored document.



To go to generic object services, choose *System → Links* from the main menu.

These functions are integrated into **all** R/3 applications via generic object services and object links; special function modules do not have to be integrated into the respective applications. See also [BC - Generic Object Services \[Ext.\]](#)



Prerequisites for the smooth operation of these functions are object orientation of the respective R/3 application and support of generic object services and object links.

Information for End Users

Information for End Users

This section is intended for end users of SAP ArchiveLink. Procedures are described that enable end users to use SAP ArchiveLink functions in their own areas.

Information is available on the following topics:

- Working with incoming documents
- Working with the Business Document Navigator
- Working with outgoing documents
- Working with print lists
- Working with business context facilities

Searching for Stored Documents

Displaying and Searching from R/3 Application Components

To view the stored documents from your R/3 application business object, choose the following in the appropriate R/3 application component:

Environment → *Object links* → *Stored documents*



From the R/3 application, a hit list of the stored documents is displayed in an amodal window.

Displaying and Searching Directly in the Content Server



In R/3 Release 4.6A, you can use the **Business Document Navigator** to search for and display stored documents.

To search for stored documents directly in the content server, proceed as follows:

1. Choose *Office* → *Business Documents* → *Documents* → *Find*.
2. Enter the following:
 - *Class name*
Name of object (of the business object, for example)
 - *Class type*
 - i. BO: Business Object Repository object
 - ii. CL: Class Browser object
 - iii. OT: Other objects
 - *Object ID*

To further limit the search, you can specify the following information:

- *Description*
- *Document type*
- *Created by*
Creator
- *Last changed by*

You can also specify up to 5 keywords.

3. Choose *Program* → *Execute*.

A list of all documents that fulfil the search criteria is displayed in the **Business Document Navigator**.

For more information on the Business Document Navigator, see the section [BDN \[Page 13\]](#).

Searching for Stored Documents



As an **administrator** you can also search for and display stored incoming and outgoing documents via SAP ArchiveLink administration.
The procedure is described in [Searching for Incoming and Outgoing Documents \[Page 144\]](#).

Maintaining User-Specific Storage Settings



The user-specific settings are available as of R/3 Release 4.5A.

Use

You can maintain various storage settings for the various storage scenarios within SAP ArchiveLink and define these as your initial screen.

Features

You can make the following settings:

- Direct selection of storage scenario
 - In the default settings, icons are displayed in front of the document types for the various storage scenarios, enabling direct storing.
Direct selection is the default.
- Display key instead of long text
 - In the default settings, the key, instead of the long texts, is specified for the default setting, document type and user.
- Compressed output
 - The default settings display takes up as little space as possible. This is useful for small monitors.
- Individual processing
 - This default setting is useful if different documents are processed at one work center. After storing and, if applicable, processing the work item, you go directly back to the hierarchy.
- Restrict view to
 - You can restrict the view to a specific default setting.

Procedure

1. Choose *Office* → *Business Documents* → *Documents* → *Move*.
A list of all relevant default settings is displayed.
2. Choose *Settings* → *User settings*.
A dialog box showing the settings options is displayed.
3. Activate the required settings.
4. If you want to save the settings, choose *Settings* → *Define initial screen*.

Storing for Subsequent Entry

Storing for Subsequent Entry

The following describes in detail the procedure for "Storing for subsequent entry", which consists of assigning the document type and processing the work item.

Prerequisite

Default settings for the storage scenarios are defined.

Procedure: Assigning a Document Type

Once the presorted documents have been scanned, they are available in the scan dialog display window as document stacks. The documents must now be transferred to the R/3 System and processed there.

As a person responsible for assignment, you should proceed as follows to assign the document class:

1. In the R/3 window, choose:

Office → Business documents → Documents → Store

A list of all defined default settings is displayed.

You can make [user-specific settings \[Page 93\]](#).

2. Select a default setting.

Each default setting contains a list of the relevant document types and people responsible.



For more information on an individual (selected) entry, choose *Utilities → Information*.

3. For the required document type, choose *Document → Storing for subsequent entry*.

You receive a dialog box, in which you can, if necessary, change the note that is displayed with the long text for the document type.

4. You can continue in three ways. You can confirm the assignment, cancel processing or execute mass storing.

- a) *OK*

Confirms assignment

The document is then stored in the content server and a work item created in the integrated inbox work list.

You can then deal with the remaining documents in the document stack as described in steps 1 to 4.

- b) *Cancel*

Cancels processing

- c) *Mass storing*

Storing for Subsequent Entry

This function is useful if the documents in the document stack have already been checked and sorted and all documents are to be processed in the same way (that is, they are all of the same document type).

If you select *Mass storing*, a confirmation prompt is shown in which you must confirm your choice.

The documents are then removed automatically one after the other from the document stack. The same actions, as described under *Ok*, are performed for each document.

5. Continue with the processing of the **work item**.

Procedure: Processing a Work Item

As a person responsible for an application (e.g. an accountant), you have work items in your integrated inbox. To process a work item, proceed as follows:

1. Go to your inbox (*Office* → *Inbox*) and display your work items for *Workflow*.
2. Select the work item you want to process and choose *Work item* → *Execute*.

If you are using the standard task TS30001128, the stored document is displayed in the viewer window. A dialog box is also displayed with the following options:

– *Process*

You branch to the relevant application transaction, in which you create the business object. When the business object is created, the incoming document is simultaneously linked to the business object.

When the relevant application transaction has been completed, you return to the workflow inbox.

Now, using the usual R/3 methods, you can search the business object, view the posting data and display the incoming document that is linked with this business object. There can be several incoming documents for a business object. For example: quotation, purchase order, order acknowledgment, delivery note, invoice.

– *Change*

If the wrong document type was assigned or the wrong processing was chosen (enter, assign), choose *change* and then the correct document type or storage scenario. You can also change the recipient and the note.

In the standard system, the work item is stored in the inbox when the document type is changed and can be further processed from there later. The work item is processed immediately if the scenario "storing for subsequent entry" is activated in the relevant default setting. You can maintain the storage scenarios according to your own requirements by defining default settings.

– *Create*

If additional document types need to be assigned (they may be noted, for instance, on the incoming document), choose *Create*.

Select the appropriate document type. If not all the elements required for the workflow are available, the dialog box "Edit container instance" is displayed. Make the necessary entries, then choose *Goto* → *Back*.

Storing for Subsequent Entry

A new workflow is started which is placed in the inboxes of the agents responsible for this document type. The old work item remains in the workflow inbox.

You cannot select the same document type and processing several times.

- *Cancel*

Processing is canceled.

Further information:

- [Concept: Storing for Subsequent Entry \[Page 62\]](#)
- [Procedure: Storing for Subsequent Entry \[Page 64\]](#)

Early Storing with Bar Codes

The procedure for early storing with bar codes is almost the same as the procedure for late storing with bar codes, the only difference being that in early storing an incoming document with a bar code is entered directly in the mailroom and the R/3 System is then updated.

See also:

[Late Storing with Bar Codes \[Page 102\]](#)

Further information:

- [Concept: Storing with Bar Codes \[Page 76\]](#)
- [Process Flow: Early Storing with Bar Codes \[Page 78\]](#)

Storing and Entering

Storing and Entering

Prerequisite

Default settings for the storage scenarios are defined.

Procedure

To enter the business object for the incoming document, proceed as follows:

1. Choose *Office* → *Business Documents* → *Documents* → *Store*.
A list of all defined default settings is displayed.
You can make [user-specific settings \[Page 93\]](#).
2. Select a default setting.
Each default setting contains a list of the relevant document types and people responsible.

For more information on document types, choose *Utilities* → *Information*.
3. For the required document type, choose *Document* → *Store and enter*.
You receive a dialog box, in which you can, if necessary, change the note that is displayed with the long text for the document type.
4. You can continue in three ways. You can either confirm the assignment with *Continue* or choose *Mass storing* or cancel processing.
 - *OK*
You branch to the respective application transaction. If there are more incoming documents in the document stack after processing, you must confirm the document type assignment for each document.
 - *Mass storing*
This function is useful if the documents in the document stack have already been checked and sorted and all documents are to be processed in the same way (that is, they all have the same document type).
If you select *Mass storing*, you branch directly to the application transaction for each document in the document stack, without being asked to confirm the assignment again.
 - *Cancel*
This cancels processing.
5. Continue with the processing of the **work item**. You have the following processing options:
 - *process*
You branch to the relevant application transaction, in which you create the business object. When the business object is created, the incoming document is simultaneously linked to the business object.
Now, using the usual R/3 methods, you can search the business object, view the posting data and display the incoming document that is linked with this business

Storing and Entering

object. There can be several incoming documents for a business object. For example: quotation, purchase order, order acknowledgment, delivery note, invoice.

– *change*

If the wrong document type was assigned or the wrong processing was chosen (enter, assign), choose *change* and then the correct document type or storage scenario. You can also change the recipient and the note.

In the standard system, the work item is stored in the inbox when the document type is changed and can be further processed from there later. The work item is processed immediately if the scenario "storing and entering" is activated in the relevant default setting. You can maintain the storage scenarios according to your own requirements by defining default settings.

– *Create*

If additional document types need to be assigned (they may be noted, for instance, on the incoming document), choose *Create*.

Select the appropriate document type. If not all the elements required for the workflow are available, the dialog box "Edit container instance" is displayed. Make the necessary entries, then choose *Goto → Back*.

A new workflow is started which is placed in the inboxes of the agents responsible for this document type. The old work item remains in the workflow inbox.

You cannot select the same document type and processing several times.

– *cancel*

Processing is canceled.

Further information:

- [Concept: Storing and Entering \[Page 66\]](#)
- [Process Flow: Storing and Entering \[Page 67\]](#)

Assigning then Storing

Assigning then Storing

Prerequisite

Default settings for the storage scenarios are defined.

Procedure

1. The business object to which the incoming document relates, already exists in the R/3 System.

2. As a person responsible for data entry, you choose *Office* → *Business Documents* → *Documents* → *Store*.

A list of all defined default settings is displayed.

You can make [user-specific settings \[Page 93\]](#).

3. Select a default setting.

Each default setting contains a list of the relevant document types and people responsible.



For more information on document types, choose *Utilities* → *Information*.

4. For the required document type, choose *Document* → *Assigning and storing*.

5. In the following dialog box, enter the business object number.

You can also display the business document.

6. Choose *OK*.

- If *Confirm* is implemented as object type method, a check window appears with the most important data for the business object, for example, invoice number, invoice creator, etc. Check whether the information on the business object agrees with that of the displayed incoming document.

- If *Existence Check* is implemented as object type method, there is no check window, only an existence check.

7. Confirm the assignment.

If you confirm:

- the incoming document is stored
- the business object is assigned to the stored document.

Further information:

- [Concept: Assigning then Storing \[Page 74\]](#)
- [Process Flow: Assigning then Storing \[Page 75\]](#)

Late Storing with Bar Codes

Late Storing with Bar Codes

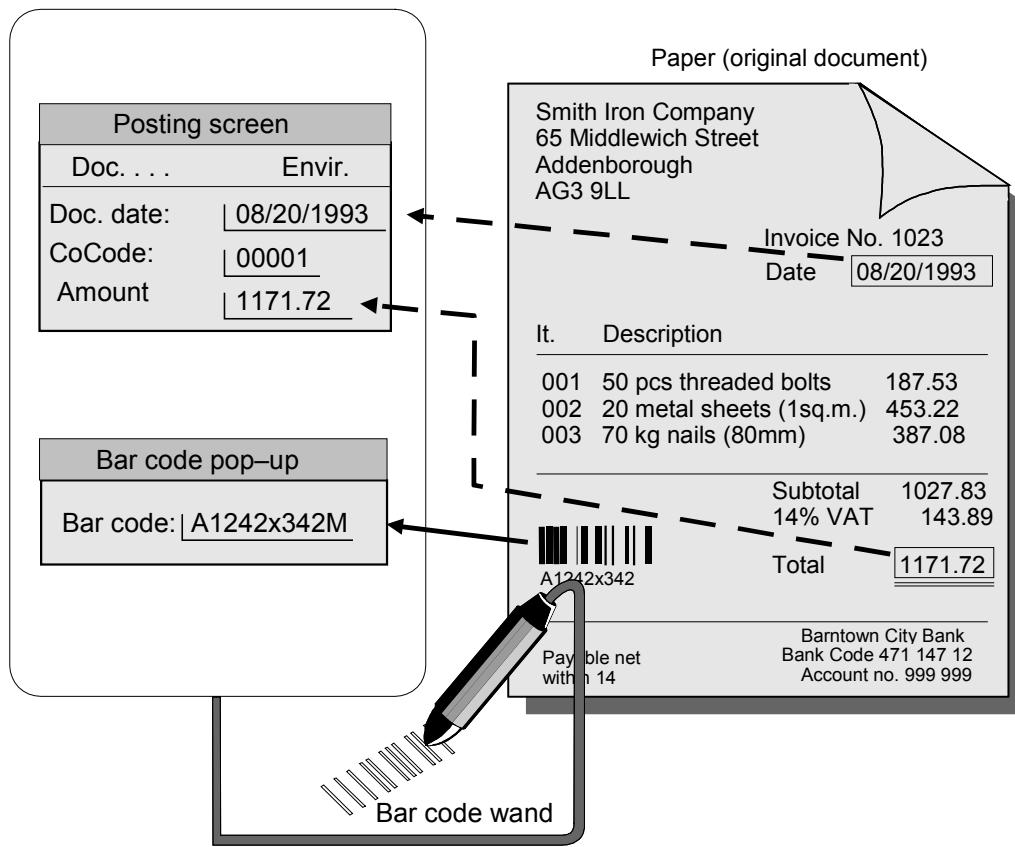
Procedure

To perform late storing with bar codes on an incoming document, proceed as follows:

1. Enter the business object for the incoming document.

The incoming document already has a bar code. This can be carried out in the mail depot for example, by affixing ready-made bar code labels, or you can do it yourself. Only the first (or last) page of a multi-page document is given a bar code label.

Screen with R/3 application (e.g. FI)



→ manually or using bar code wand
→ manual transfer

2. A dialog box where you enter the bar code (using a bar code reader or bar code gun, or manually) is displayed during each posting procedure in your R/3 application.
3. The incoming document is then forwarded as paper to the entry location (generally a central location), where it is scanned and checked on the screen for quality and completeness. The procedure from this point is carried out automatically as long as no problems occur:
 - The scanning component recognizes the bar code and reports it back to the R/3 system together with the document ID.

Late Storing with Bar Codes

- SAP ArchiveLink assigns the stored document to the corresponding business object.
The business object can then be accessed.

Further information:

- [Concept: Storing with Bar Codes \[Page 76\]](#)
- [Process Flow: Late Storing with Bar Codes \[Page 79\]](#)

Searching for, Displaying and Printing Print Lists

Searching for, Displaying and Printing Print Lists

Procedure

To search for stored print lists in the content server and to display and print these lists, proceed as follows:

1. From the main menu, choose:

Office → Business Documents → Find lists.

You branch to the screen “Find Stored Print Lists”.

2. Enter (if known):

- *Report name*

The report that generated the print lists.

- *Info*

If the report name is ambiguous, you must make an entry here.

- *Storage date from...to*

If you enter dates, the system only searches for print lists which were stored within the specified period.

- *Business object*

- *Document type*

- *User*

- *Short text*

If you want to search for **all** existing stored print lists, leave the fields empty and continue with step 3.

3. Choose *Program → Execute*.

A list is displayed of the stored print lists that fulfil the search criteria.



As of R/3 Release 4.0A, the list display corresponds to the SAP standard layout, which means that all standard filter and sort functions can be used. For more information, see [ABAP List Viewer \[Ext.\]](#).



You can display a print list using the icon *Display from content server*.

You can display the status for each entry in the hit list using the icon *Status query*.

4. To view the link entries for a stored document, select the relevant document and choose *Select detail*.

The link entries for the relevant print list are displayed.

5. You can now display the following information about the print lists:

- *Display*

Searching for, Displaying and Printing Print Lists

To display the relevant print list, choose *Print list* → *Display from content server*.



As of R/3 Release 4.5B, print lists are displayed, as standard, directly in the actual dynpro.

For further information, see [Displaying Link Entries for Stored Print Lists \[Page 153\]](#).

- Print

To print the print list, choose *Print list* → *Print from content server*.



You can only print the print list if you have the appropriate authorization (see also [Displaying Link Entries for Stored Print Lists \[Page 153\]](#)).

You can also use the print icon in the system toolbar to print the link entry. You do not need additional authorization to do this.



As an **administrator** you can also search for and display stored print lists via SAP ArchiveLink administration. The procedure is described in [Searching for Stored Print Lists \[Page 152\]](#) and [Displaying Link Entries for Stored Print Lists \[Page 153\]](#).

Further information:

- Concept: [Print Lists \[Page 85\]](#)
- [SAP ArchiveLink - Scenarios in Applications \[Ext.\]](#) documentation.

Transferring Stored Documents

Transferring Stored Documents

To transfer documents that are already stored, to other R/3 application components, proceed as follows :

1. Activate the display of the document in the application where the stored document already exists.
2. Select a document from the hit list.
3. Transfer the document to the relevant R/3 application.



See also [Business Context Facilities \[Page 89\]](#). For more information, see [Assigning Documents Generically \[Page 108\]](#).

Further information:

- Concept: [Transfer to R/3 Application Components \[Page 86\]](#)
- [SAP ArchiveLink - Scenarios in Applications \[Ext.\]](#) documentation.

Displaying Stored Documents Generically

From your application transaction, you can display the stored documents belonging to the relevant business object via the system menu.

For more information, see [BC – Business Context Facilities and Object Relationship Services \[Ext.\]](#).

Prerequisites

The R/3 application publishes its objects.

Procedure

1. From the relevant transaction or the your business object, choose *System → Links*.

A list of the stored documents belonging to your business object is displayed.

2. Position the cursor on an entry and choose *Linked object*.

The relevant linked document is displayed.

3. You can use a variety of other functions. For example:

- Store document

See also [Storing Documents Generically \[Page 109\]](#).

- Create link to stored document

See also [Assigning Documents Generically \[Page 108\]](#).

- Link bar code

See also [Assigning Bar Codes Generically \[Page 110\]](#).

Assigning Documents Generically

Assigning Documents Generically

From your application transaction, you can subsequently assign the stored documents belonging to the relevant business object using the system menu.

Prerequisites

The R/3 application supports the business context facilities.

Procedure

1. From the relevant transaction or business object, choose *System → Links*.
2. Choose *Create link to stored document*.
3. Select a document and processing type.
 - *Select*
Continue with step 3.
 - *Check selection*
If a document is already selected in the viewer, this document is automatically used and can be assigned.
 - *Display document*
The stored document is displayed from the content server in your viewer.
 - *Cancel*
3. Confirm or change the document type.

Result

The document has been assigned subsequently.

Storing Documents Generically

From your application transaction, you can store the documents belonging to the relevant business object, using the system menu.

Prerequisites

The R/3 application supports the business context facilities.

Procedure

1. From the relevant transaction or business object, choose *System → Links*.
2. Choose *Store document*.

A dialog box is displayed showing the document types defined for the relevant object type.

3. If the default is set, that is, if you have not activated the option "External program for document type selection" in your basic settings, an ABAP dialog box appears, which helps you to perform the following steps.

Depending on content server Customizing, you continue in one of the following ways:

- Double-clicking on the required document type displays a dialog box, in which you can specify the path of the file to be stored.
or
- An appropriately maintained application is started via OLE. Now you can enter a new document.

Continue with step 3.

3. Position the cursor in the dialog box on the document and choose *OK*.

Result

The document has been stored in dialog and appears in the list of existing links.

Assigning Bar Codes Generically

Assigning Bar Codes Generically

From your application transaction, you can assign a bar code to the relevant business object.

Prerequisites

- The R/3 application supports the business context facilities.
- Storing with bar codes is activated for your object type.
- An assignment to a content server is maintained for the object type in Customizing.

Procedure

1. From the relevant transaction, choose *System → Links*.
2. Choose *Enter bar code*.
3. Enter a bar code directly or using F4 help.
4. Select a suitable document type.

Using F4 help, you can display the document types maintained in Customizing.

5. Choose *Continue*.

Result

You have assigned a bar code to your business object. You can now store a document assigned to your business object automatically using bar code recognition.



You can view the result of your assignment in the monitor in the list of open bar codes.

Information for Administrators

The following sections are intended for administrators.

- The administration tool *Monitor* is explained.
- The administration of stored documents is explained.
- The authorizations and profiles for SAP ArchiveLink are explained.

The Administration Tool Monitor

The Administration Tool Monitor

The administration function *Monitor* allows you to edit or check all the components that are necessary for SAP ArchiveLink (spool, queues, background processing and so on) from a central position. This function therefore provides a general tool for monitoring.

The following sections concern the operation of SAP ArchiveLink. The following topics are described:

- Structure of the monitor screen
- Prerequisites for storage requests
- Process flow, error situations and administration options for asynchronous storing, retrieving, and late and early storing with bar codes.



As of R/3 Release 4.0A, the SAP ArchiveLink Monitor contains only monitoring functions.

The Customizing functions are available in the Implementation Guide (IMG) as of R/3 Release 4.0A.

Positioning in the R/3 System

To go to the SAP ArchiveLink Monitor, choose

Tools → *Business Documents*,

Environment → *ArchiveLink Monitor*

Monitor Screen

To go to the initial monitor screen, choose
Tools → *Business documents*,
Environment → *ArchiveLink Monitor*.

The screen “SAP ArchiveLink Monitor” provides you with an overview of the current status of the following areas:

Monitor screen

| Group Heading | Description |
|------------------------------|--|
| Asynchronous requests | <p>Open The list of open jobs is displayed for all asynchronous storage requests. This includes, for example, storing of archive files and print lists and printing of stored lists.</p> <p> The list of jobs with errors is displayed for all asynchronous storage requests.</p> |
| Requests via tRFC | <p></p> Display of all failed calls of the function module ARCHIV_CREATE_TABLE when storing outgoing documents. |
| Queues | <p>The number of entries in the different queues is specified under this group heading.</p> <p><i>Storage</i> CARA queue: contains requests for asynchronous storing.</p> <p><i>Confirmation</i> CFBC queue: contains confirmations for asynchronous storing and bar code entries.</p> <p><i>Retrieve</i> CFBA queue: contains requests for asynchronous retrieval.</p> <p>For each queue, there is an error queue (CARA_E, CFBC_E, CFBA_E). The error queue contains entries if errors occur when the CARA, CFBC or CFBA queues are processed.</p> |
| Connection test | Under this group heading, you use the pushbutton <i>Display connection test</i> , to check whether individual content servers can be addressed. The status bar shows progress made. You can call a list of connection errors using the pushbutton  . |
| Background processing | <p><i>Scheduling</i> <i>Storing</i> <i>Confirmation</i> <i>Retrieval</i></p> The information displayed under this group heading indicates whether a job is scheduled for background processing and whether this job has started the jobs that process the various queues. To see jobs with errors, use the pushbutton  . |

Monitor Screen

| | |
|------------------|---|
| Bar codes | This group header concerns storing incoming documents with bar codes. Open, internal and external bar codes are displayed. If a bar code has been entered incorrectly, you can change the bar code value. You can also compare bar codes you change manually, with any existing bar code confirmations. |
| Logging | <p><i>Entries</i></p> <p>The number of errors and other entries (such as user and command) contained in the logging table are displayed. Errors can occur when sending commands to content servers or when attempting to create link entries, for example. From the detail display for a logging entry you can display the error message and/or delete the entry.</p> |

As of R/3 Release 4.0A, the list display corresponds to the SAP standard layout, which means that all standard filter and sort functions can be used. For more information, see [ABAP List Viewer \[Ext.\]](#).

The following functions are offered as of R/3 Release 4.5A:

- You can refresh the display of the individual areas by choosing the relevant *Refresh* pushbutton. You can use the upper most *Refresh* pushbutton to refresh all areas; you can also select and deselect areas to be refreshed.
- The status bar shows the progress of the refresh.
- You can save a type of update display for each user under *Settings* → *Define initial screen*. For more information, see online help.

You can perform the following actions on the monitor screen:

- You can branch directly to the respective component (background processing, for example) using the pushbuttons. From there, you can call the usual functions (such as jobs).
- You can reset the display to the Alert Monitor, under *Edit*:

If problems occur in the SAP ArchiveLink environment, an error (“alert”) is triggered in the Computing Center Management System (CCMS) Alert Monitor.



To go to the CCMS Alert Monitor, choose
Tools → *CCMS*,
Control/Monitoring → *Alert Monitor*.

The errors are classified as yellow (critical) or red (very critical) alert.

The function *Reset Alert* in the SAP ArchiveLink Monitor updates the display on the Alert Monitor, that is, the red or yellow traffic light is reset to green. This is needed if an error has been rectified, for example.

To update the Alert, choose
Edit → *Reset alert* in the SAP ArchiveLink Monitor.

- There are several processing options under *Environment* (see [Monitor Environment \[Page 116\]](#)).

Monitor Environment

Monitor Environment

There are more processing options in the Monitor under the menu option *Environment*. You can branch to the following transactions:

- Transactional RFC
- Spool Administration
- Application log
- Administration of stored documents (see also [Administration of Stored Documents \[Page 139\]](#))
- Business Document Navigator (see also [Business Document Navigator \[Page 13\]](#))

Transactional RFC



The monitoring function for transactional RFCs is available as of R/3 Release 4.0A.

From the Monitor, choose

Environment → Transactional RFC

You go to the monitor for failed transactional RFC calls. An overview of users who have called SAP ArchiveLink function modules is displayed. You can also call this overview for the user currently logged on, if necessary.

Failed transactional RFC calls are logged here. If an error occurs when generating an outgoing document, for example, the relevant function module (ARCHIV_CREATE_TABLE) can be restarted here.

An interface command is sent to the connected content server, which transfers the known RFC information used for communication with external systems. This informs the content server of the RFC parameters via which the current R/3 System can be reached.

Spool Administration

From the monitor, choose

Environment → Spool administration

You go to spool administration, where you can specify the output device.

See also: [Archive Device and Spool \[Page 119\]](#).

Displaying the Application Log



This function enables you to display an overview of the errors that occurred during Business Document Service bar code entry and is available as of R/3 Release 4.6A.

1. From the Monitor, choose
Environment → Application log
2. Enter **SBDS** as the object.

3. Enter one of the following as the sub-object:
 - a. **BAR CODE**
 - Application log for bar code processing
 - b. **CONNECTION**
 - Link entry
 - c. **CORE**
 - Business Document Service: Core and API
4. Choose *Execute*.

For more information, see [Displaying Logs \[Ext.\]](#).

Prerequisites for Storage Requests

Before storage requests can be started, the following tasks must have been completed:

- Customizing settings for SAP ArchiveLink must have been made (see [Basic Settings \[Page 168\]](#)).
- All queues must have been created.



You must not create queues implicitly, since this can lead to errors.

- The job ARCHIVELINK must have been scheduled.



You execute these activities via the Implementation Guide (IMG).

In the IMG, choose

“R/3”, “Basis”, “Basis Services”, “SAP ArchiveLink”. For information on the individual activities, refer to the online help in the IMG.

Archive Device and Spool

The archive device is the logical printer to which the spool sends storage requests.

For storing print lists, an archive device must be defined in spool administration. In addition, a particular device type must be defined for the relevant archive device in spool administration. If no device type is defined, the message MISSING appears next to *Archive device* and you must create an archive device.

From the Monitor, you can go straight to spool administration via *Environment → Spool administration*. You can display the output devices and the corresponding device types.

In the standard R/3 System, an archive device **ARCH** of type **ARCHIXOS** is the default for storing with SAP ArchiveLink. This archive device is defined via the profile parameter **rspo/default_archiver**.



You can create a new archive device and a new device type. Note the settings specified via the archive device **ARCH** and the device type **ARCHIXOS**.

For this, choose

Tools → Business Documents,
Basic Settings → Maintain archive device

or

In the Implementation Guide (IMG), choose
 “R/3”, “Basis”, “Basis Services”, “SAP ArchiveLink”, “Basic Settings”, “Maintain content servers”

You can find the procedure for creating an archive device and a device type in the BC - Printer Manual documentation in the section [Defining and Modifying Device Types \[Ext.\]](#).

Profile Parameter

Profile parameters are configured outside the R/3 System. For storing, the following two profile parameters must be configured as shown:

| Profile Parameter | Value | Details |
|------------------------------|-------------|--|
| rspo/archive_format | 2 | Print list format with color information and/or hyperlinks |
| rspo/archive_archiver | ARCH | Archive device |



You can determine the current profile parameter settings using the report RSPARAM.

Data formats

There are two data formats that are configured using the profile parameter **rspo/archive_format**:

Archive Device and Spool

-{}-• Format 1 Contains ASCII characters and line feed and form feed as control characters

-{}-• Format 2 Contains print controls, columns, comments and DARC index lines



When creating print lists that contain color information and/or hyperlinks, the profile parameter **rspo/archive_format** must have the value '2' (see above).

Storing Outgoing Documents



As of R/3 Release 4.0B document type PDF exclusively is used for outbound documents. No more outgoing documents are created in OTF Format.

Outgoing documents are stored synchronously in PDF format via the function module ARCHIV_CREATE_TABLE, using the **transactional remote function call (tRFC)**. Storing, therefore, **no longer** takes place via:

- CFBC command
- Spool

See also: Information for developers: [Overview: Storing Outgoing Documents \[Page 214\]](#)

Storing

The R/3 application generates an outgoing SAPscript document:

- **directly** or
- **indirectly** via message control

A function module is called from SAPscript that receives transferred SAPscript data. This data is stored directly via the CART command (table, see [CART \[Ext.\]](#)), so that no more files are generated in the file system.



Before R/3 Release 3.0 outgoing documents were always generated and stored via the spool.

As of R/3 Release 3.0 outgoing documents in PDF format can also be stored without the spool.

As of R/3 Release 3.1 G outgoing documents in OTF and PDF formats were stored without the spool.

Errors

- **R/3 Release < 3.1G**

If an **error** occurs during storing with an R/3 Release earlier than 3.1G, the outgoing documents must:

- either be generated from the application again
- or the whole spool list must be stored anew, even if there is only one storage error.

- **R/3 Release >= 3.1G**

If an **error** occurs during storing **as of 3.1G**, an overview is displayed with those documents for which an error occurred. The function module (ARCHIV_CREATE_TABLE) is called by a **transactional RFC**. Via the restart mechanism associated with the transactional RFC, a restart is again possible. In the Monitor, choose:

Storing Outgoing Documents

Environment → Transactional RFC.

To display the requests with errors, use the icon *Display requests with errors via tRFC* under the group heading *Requests via tRFC*.

For further information on the transactional RFC, see the *RFC Programming in ABAP* documentation, section "[Transactional Remote Function Calls \[Ext.\]](#)".

Confirmation

The content server reports successful storing back to SAP ArchiveLink in the RART message (see [RART \[Ext.\]](#)). The message contains the content server and the document ID. SAP ArchiveLink then generates the link between the stored document and the business object from the R/3 application component.

Asynchronous Storing of Print Lists and Archive Files

Asynchronous Storing of Print Lists and Archive Files

In SAP ArchiveLink, print lists and archive files are stored **asynchronously**.

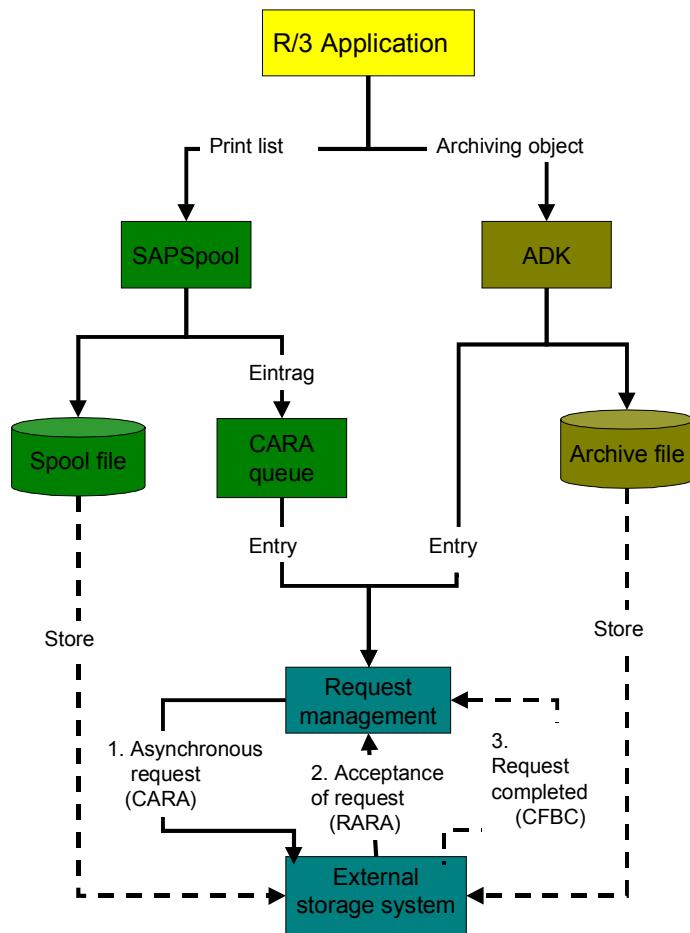


If you use SAP ArchiveLink via HTTP instead of RFC, there are no asynchronous requests for storing and retrieving. Therefore asynchronous requests are handled synchronously. This means that wait times can be long when processing requests in the Monitor or storing and retrieving from data archiving. The following solution is provided:

The asynchronous requests are mapped to background jobs, which in turn are mapped synchronously. The names of the background jobs are made up of the prefix **CARA_** (for storing) and **CFBC_** (for retrieving) and the request number. The SAP ArchiveLink Monitor has been enhanced for monitoring jobs: This type of requests appear together with the CARA and CFBC jobs under the group header background processing. If errors occur in synchronous request processing via jobs, the requests can be repeated using request management in the SAP ArchiveLink Monitor (see also: [Queue Management: Error Handling \[Page 136\]](#)).

The process flow for asynchronous storing is shown schematically in the diagram below:

Asynchronous Storing of Print Lists and Archive Files



Print lists are managed in SAP ArchiveLink. A DMS control record can also be generated. You can maintain the settings necessary for this in the Implementation Guide (IMG) under
 “R/3”, “Basis”, “Basis Services”, “SAP ArchiveLink”, “Basic Settings”, “Make basic settings”.

The procedure is described in more detail below. The position on the monitor screen where information can be called on the step involved is shown in parentheses.

1. The R/3 application generates:
 - a) A print list
 - b) An archiving object
2. The R/3 application transfers:
 - a) The print list to the spool
 - b) The archiving object to the Archive Development Kit (ADK)

Asynchronous Storing of Print Lists and Archive Files

3.
 - a) The spool generates the following:
 - A spool file to be stored in an content server
 - An entry in the CARA queue,
(group header "Queues", pushbutton *Storing*)
via which an entry in request management is then written
(group header "Asynchronous requests", pushbutton *Open*)
 - b) The ADK generates the following:
 - A archive file to be stored in a content server
 - An entry in request management
(group header "Asynchronous requests", pushbutton *Open*)
4. Request management transfers the asynchronous storage request to the content server (CARA).
5. The content server confirms the acceptance of the storage request to request management (RARA).
6. Storing is either successful or unsuccessful:
 - If storing was successful, the content server reports completion of the request to request management (CFBC).
 - If storing was not successful, an entry is made in the CFBC_E queue.
(Group heading "Queues", pushbutton *Error* (confirmation))



Incorrect storage requests can be retrieved from the error queue CFBC_E via *Transfer* and processed again.

Displaying Asynchronous Requests

Displaying Asynchronous Requests

Open Requests

You can display the open requests in request management by selecting *Open* under the group heading “Asynchronous requests”. *Open* contains requests transferred for storing/retrieval, but not yet finally processed.



As of R/3 Release 4.0A, the list display corresponds to the SAP standard layout, which means that all standard filter and sort functions can be used. For more information, see [ABAP List Viewer \[Ext.\]](#).

As of R/3 Release 4.0A, management of asynchronous requests is exclusively via request management.

The list of open requests shows the output statuses.

You can display more information by double-clicking.

Output Status

| Activity | Status | Status CARA queue | Status CFBC queue |
|--------------------------------|--------------------------------|-------------------------|-------------------------|
| Print list being prepared | Waiting for output preparation | Blank | Blank |
| Spool generates CARA entry | Sent to host spool | Request written in CARA | Blank |
| Request sent to content server | Sent to host spool | Blank | Blank |
| Storage confirmed | Done / incorrect | Blank | Blank |

The file is written to the directory that is specified as the basic path in content server customizing.

To change the basic path for an open request, display the request and choose the *Repeat* icon.



As of R/3 Release 4.0A, failed requests are repeated without generating new files.

You can query the current status using the *Status* icon.

You can delete the storage request using the *Delete* icon.

You can call the application function module using *Application*. This is only possible if the content server has already sent a confirmation.

Displaying Asynchronous Requests

Requests Containing Errors

Unsuccessful requests in request management are listed under the *Error* icon (Group heading "Asynchronous requests"). They have output status "Error". To find a description of the error situation:

Display an individual request via the group heading "Asynchronous requests" or via *Monitor* → *Asynchronous requests* → *Requests with errors*. You can perform the request again using the *Repeat selection* icon in the detail display.

Retrieving Archive Files and Print Lists

Retrieving Archive Files and Print Lists

Stored archive files and stored print lists are retrieved **asynchronously**.

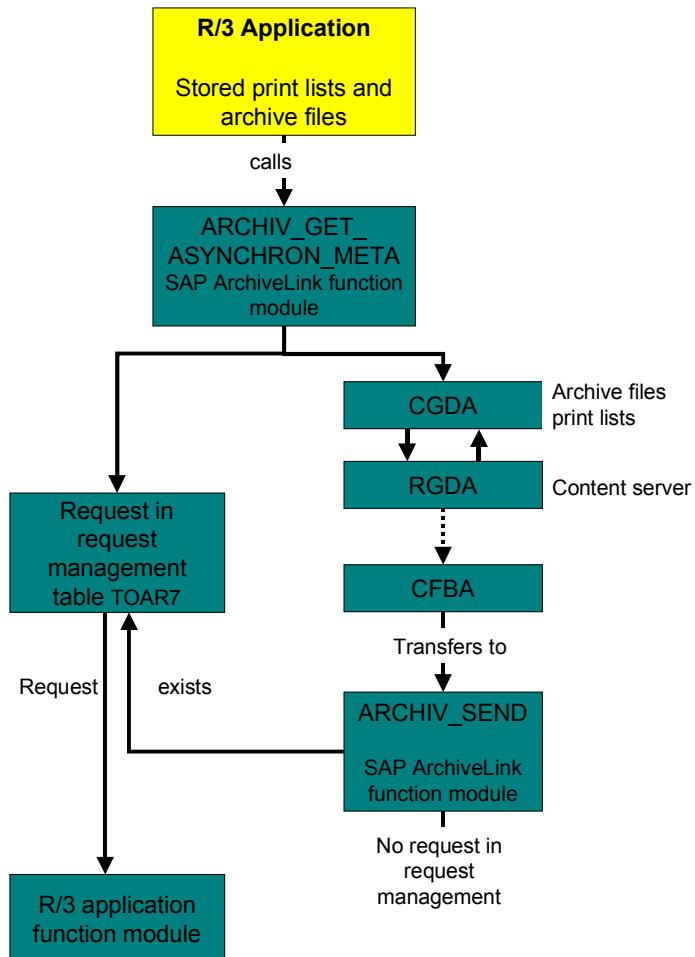


If you use SAP ArchiveLink via HTTP instead of RFC, there are no asynchronous requests for storing and retrieving. Therefore asynchronous requests are handled synchronously. This means that wait times can be long when processing requests in the Monitor or storing and retrieving from data archiving. The following solution is provided:

The asynchronous requests are mapped to background jobs, which in turn are mapped synchronously. The names of the background jobs are made up of the prefix **CARA_** (for storing) and **CFBC_** (for retrieving) and the request number. The SAP ArchiveLink Monitor has been enhanced for monitoring jobs: This type of requests appear together with the CARA and CFBC jobs under the group header background processing. If errors occur in synchronous request processing via jobs, the requests can be repeated using request management in the SAP ArchiveLink Monitor (see also: [Queue Management: Error Handling \[Page 136\]](#)).

The procedure used in asynchronous retrieval is shown schematically in the diagram below. The asynchronous retrieval procedure is analogous to that of asynchronous storing (see [Storing Print Lists and Archive Files Asynchronously \[Page 123\]](#))

Retrieving Archive Files and Print Lists



The procedure is described in more detail below.

1. The application wants stored archive files or print lists to be retrieved. It calls the SAP ArchiveLink function module for asynchronous retrieval.
 2. The function module does the following:
 - It makes an entry in SAP ArchiveLink request management.
 - It sends a retrieval request to the content server.
 3. The content server reports successful storing directly to the R/3 System using the function module **ARCHIV_SEND**.
- For unsuccessful storing, an entry is made in the **CFBA_E** error queue (group header “Queues”, pushbutton *Display CFBA_E queue (retrieval)*).

Displaying Requests in Request Management

Displaying Requests in Request Management

You can display the successful storing and retrieval requests by selecting *Open* under the group heading “Asynchronous requests”.



As of R/3 Release 4.0A, the list display corresponds to the SAP standard layout, which means that all standard filter and sort functions can be used. For more information, see [ABAP List Viewer \[Ext.\]](#).

Select *Error* to display unsuccessful requests.

If you display an individual request by double-clicking, you can:

- View the request status using the *Status* icon
- Delete the request using the *Delete* icon
- Call the application function module using the *Application* icon.
(This is only possible if the content server has already sent a confirmation.)
- Repeat the request by choosing the *Repeat* icon
This places a new request in request management; the original request remains.

Storing with Bar Codes

Use

Storing with bar codes is possible for incoming documents.

The general procedure for storing with bar codes is described in the sections [Early Storing with Bar Codes \[Page 78\]](#) and [Late Storing with Bar Codes \[Page 79\]](#).

Prerequisites

In order to store with bar codes, storing with bar codes must be integrated in the relevant R/3 application component. For example, in Financial Accounting (FI), storing with bar codes can, in principle, be called.

You can carry out Customizing for storing with bar codes via the **Implementation Guide (IMG)**. In the IMG, choose

“R/3”, “Basis”, “Basis Services”, “SAP ArchiveLink”, “Administration Settings”, [Activate Storing with Bar Codes \[Ext.\]](#).

For information on the procedure, refer to the IMG online help.

It may be necessary to maintain the bar codes in some R/3 application components. The extent to which this is necessary for individual R/3 application components is explained in the documentation [SAP ArchiveLink - Scenarios in Applications \[Ext.\]](#), in particular the sections about Financial Accounting (FI) and Sales and Distribution (SD).

Features

Using **generic object services** (see [Generic Object Services \[Page 89\]](#)), available as of R/3 Release 4.0A, a bar code can be assigned to a business object (for example) from any R/3 application that supports business context facilities (see [Assigning Bar Codes Generically \[Page 110\]](#)) and stored (see [Storing Documents Generically \[Page 109\]](#)).



At the uppermost level, only the link entry is deleted when an entry is deleted from the list of stored documents. The stored document itself is not deleted. When deleting from the detail display, you can send a delete request to the relevant content server.

When deleting an entry from the list of relevant business objects, the entry is logically deleted from both the uppermost level of the list display and the low level of the detail display. The link is deleted.

Under the group heading "Bar Codes", you can:

- Refresh the list of all open bar codes
- Compare open bar codes
- Display **all** open bar codes
- Display open internal bar codes

Internal bar codes are the bar codes that are generated in the R/3 System.

Storing with Bar Codes



You can limit the hit list by specifying additional criteria here: You can specify the storage period and the bar code creator.

- Display open external bar codes

External bar codes are the bar codes that come into the R/3 System from an external system.



You can limit the hit list by specifying additional criteria here: You can specify the storage period and the bar code creator.

From the lists displayed, you can (amongst other things):

- Delete the bar code



When open bar code entries are deleted, an authorization check is performed.

- Go to the detail display



From the lists displayed, you can use the menu option *Goto* to display an overview of all bar codes:

- You can branch to the display of **all** bar codes
- You can branch to the display of all open internal bar codes
- You can branch to the display of all open external bar codes
- You can branch to the display of all bar codes with a stop flag
- You can branch to the display of all internal bar codes with a stop flag
- You can branch to the display of all external bar codes with a stop flag



You can maintain the stop flag in bar code entry maintenance. By setting the flag, you specify that the entry concerned in the table is not deleted when it has been processed. All bar codes that contain such a flag can be displayed via the last two points under *Goto*. You can delete bar codes in the lists displayed, if necessary.

From the detailed display, you can:

- Change the bar code value
- Delete the bar code
- Display the relevant business object or document



As of R/3 Release 4.0A, the list display corresponds to the SAP standard layout, which means that all standard filter and sort functions can be used. For more information, see [ABAP List Viewer \[Ext.\]](#).

Comparing Bar Codes

You can compare bar codes using the *Compare bar codes* icon under the group heading “Bar Codes”.

If the wrong bar code is entered in storing with bar codes, no assignment can be made. In such cases, you can change the bar code manually. Changing the bar code has no effect however, because the table entries are changed but not compared. Comparing checks whether there are corresponding entries. If this is the case, a link entry is made.

Process Flow: Storing with Bar Codes

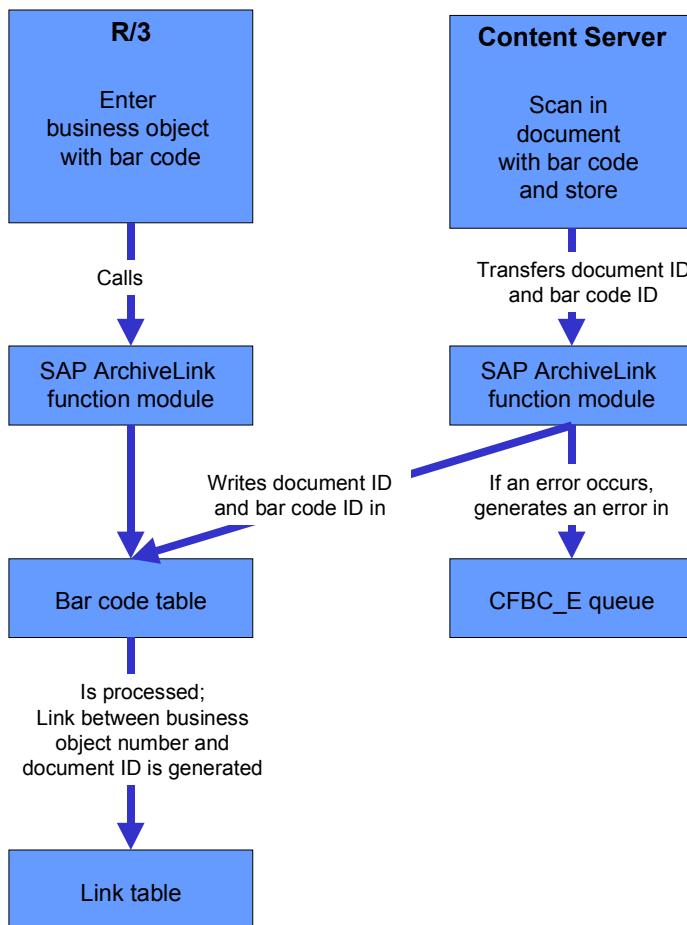
Process Flow: Storing with Bar Codes

Purpose

Incoming documents can be stored with bar codes.

Process Flow

The procedure used in storing with bar codes is shown schematically in the following diagram:



The procedure is described in more detail below. The position on the monitor screen where information can be called on the step involved is shown in parentheses.

Late Storing with Bar Codes

1. In the R/3 application, the business object is entered and the bar code entered in a dialog box.
2. SAP ArchiveLink enters the number of the business object and the bar code ID in the internal bar code table.

Process Flow: Storing with Bar Codes

3. In the content server, the incoming document in paper form is scanned with a bar code and stored.
4. The content server or the scan component transfers the document ID for the incoming document and the bar code ID to the R/3 System.
(Group heading "Queues", pushbutton *Confirmation*)
5. The internal bar code table is processed and a link record generated with the number of the business object and document ID.
6. The entry in the link table contains the link between business object and stored document.

Early Storing with Bar Codes

1. In the content server, the incoming document is scanned with a bar code and stored.
2. The content server transfers the document ID for the stored document and the bar code ID, to the R/3 System.
(Group heading "Queues", pushbutton *Confirmation*)
3. SAP ArchiveLink enters the document ID and the bar code ID in the external bar code table.
4. In the R/3 application, the business object is entered and the bar code entered in a dialog box.
5. The bar code table is processed and a link record generated with the number of the business object and document ID.
6. The entry contains the link between business object and stored document.

Queue Management: Error Handling

Queue Management: Error Handling

The following gives further details on the possible error situations for the queues below:

- **CARA / CARA_E** (contains asynchronous storage requests)
- **CFBC_E** (contains asynchronous confirmations and bar codes from the content server)
- **CFBC_E** (contains asynchronous confirmations for asynchronous retrieval requests)



As of R/3 Release 4.0A, the list display corresponds for all queues to the SAP standard layout, that is, all standard filter and sort functions can be used. For more information, see [ABAP List Viewer \[Ext.\]](#).

CARA- / CARA_E Queue

Generated asynchronous storage requests are located in the CARA queue. You can access the queue management via *Storing* under the group heading "Queues".

Different error situations can arise when the queue is processed. Entries containing errors are written to the CARA_E queue. To go to the monitor for queue management, choose *Monitor* → *Queues* → *Store* or *Monitor* → *Queues* → *Errors* → *Store* or use the pushbuttons under the group heading "Queues".

You can obtain detailed information on each queue entry by choosing the icon *Select detail*. You can:

- process the request
- delete the request or
- transfer the request

CARA Queue: Error Situations

| Error type | Action | Solution |
|----------------------|--|--|
| Content server error | Message to spool | Display the request using the <i>Display asynchronous requests with errors</i> icon (asynchronous requests) and repeat the request. |
| Kernel timeout | Message to spool | Display the request using the <i>Display asynchronous requests with errors</i> icon (asynchronous requests) and repeat the request. |
| Customizing error | Message to spool and administrator and transfer of the request to the error queue (CARA_E) | Display the request by choosing <i>Display CARA_E queue</i> (storing), branch to the individual display by double-clicking and then choose <i>Transfer</i> . |

Queue Management: Error Handling

| | | |
|------------------------------|--|--|
| Spool writes to invalid path | Message to spool and administrator and transfer of the request to the error queue (CARA_E) | Display the request by choosing <i>Display CARA_E queue</i> (storing), branch to the individual display by double-clicking and then choose <i>Transfer</i> . |
|------------------------------|--|--|



If you choose *Transfer*, an entry is always written in the relevant error queue, in the above case from the CARA queue into the CARA_E queue.

In the case of errors from the content server and kernel timeouts, the request is deleted from the CARA queue.

CFBC_E queue

Defective content server confirmations for asynchronous storage entries are in the CFBC_E-Queue. To go to the monitor for queue management, choose *Monitor* → *Queues* → *Errors* → *Confirmation* or use the pushbuttons under the group heading "Queues".

You can obtain detailed information on each queue entry by choosing the icon *Detailed information*. You can:

- process the request
- delete the request or
- display the request or
- transfer the request

CFBC Queue: Error Situations

| Error type | Action | Solution |
|---|--|---|
| Wrong message | Message to administrator and transfer of the request to the error queue (CFBC_E) | Display the request by choosing <i>Display CFBC_E queue</i> (confirmation), branch to the individual display by double-clicking and then choose <i>Transfer</i> . |
| No request exists for the archive file | Message to administrator and transfer of the request to the error queue (CFBC_E) | Display the request by choosing <i>Display CFBC_E queue</i> (confirmation), branch to the individual display by double-clicking and then choose <i>Transfer</i> . |
| Content server error for bar code confirmations | Message to administrator and transfer of the request to the error queue (CFBC_E) | Display the request by choosing <i>Display CFBC_E queue</i> (confirmation), branch to the individual display by double-clicking and then choose <i>Transfer</i> . |

Queue Management: Error Handling

| | | |
|---|--|---|
| No entry can be made in the link table for outgoing documents/bar codes | Message to administrator and transfer of the request to the error queue (CFBC_E) | Display the request by choosing <i>Display CFBC_E queue</i> (confirmation), branch to the individual display by double-clicking and then choose <i>Transfer</i> . |
| Content server error for outgoing documents/print lists | Message to spool | Display the request by selecting <i>Display CFBC_E queue</i> (confirmation) and repeat the request. |

CFBA_E queue

Defective content server confirmations for asynchronous retrieval entries are in the CFBA_E queue. To go to the monitor for queue management, choose *Monitor* → *Queues* → *Retrieve* or *Monitor* → *Queues* → *Errors* → *Retrieve* or use the pushbuttons under the group heading "Queues".

You can obtain detailed information on each queue entry by choosing the icon *Detailed information*. You can:

- process the request
- delete the request or
- transfer the request

The following error situations are possible:

- No request exists

In the case of a request for asynchronous retrieval, a request must first be generated in the asynchronous entries administration. This request management contains an overview of the current requests, their status, and the name of a function module to be called.

- Content server error confirmation

In this case, the function module to be called is informed and the status in request management set to "Archive error confirmation". It is then possible to repeat this request again from request management.

Administration of Stored Documents

To go to Administration of stored documents, choose:

Tools → Business Documents,
Miscellaneous → Stored documents

From Administration of stored documents, you can branch to the following areas:

- **Archive Development Kit (ADK) administration**
Choose *Goto → Archive administration (ADK)*.
- **SAP ArchiveLink Monitor**
Choose *Environment → ArchiveLink Monitor*.
For more information, see [Administration Tool Monitor \[Page 112\]](#).
- **Business Document Navigator**
Choose *Environment → Business documents*.
For more information, see [Business Document Navigator \[Page 13\]](#).

Administrative Functions

The following administrative functions are available:

- You can link an existing business object to stored documents.
- You are shown how many stored documents exist per document type in the content server (incoming and outgoing documents, print lists, archive files).
- You can search for specific stored documents and display images and link entries for these documents.

The following additional functions are available for incoming and outgoing documents:

- You can store incoming documents (store and enter).
- You can reassign and subsequently assign stored documents.
- You can display administration information, a list of link entries and the corresponding business objects.
- You can change the document type.
- You can delete the link between business object and stored document.
- Delete document type (where the last link is concerned)

Overview

In addition, in an overview the number of the following stored documents is displayed:

- Incoming and outgoing documents
- Print lists
- Archive files

Administration of Stored Documents

To get a current overview of the number of stored documents, choose *Stored Documents* → *Refresh number* in administration of stored documents *Stored documents* → *Refresh number*. The dates and times in the overview change correspondingly.

Linking Business Objects with Stored Documents

Procedure

To link an existing business object with stored documents, proceed as follows:

1. In administration of stored documents, choose
Stored documents → *Links*.
2. Choose a stored document from the following dialog box.



The document that was last displayed is first in the dialog box.

You can continue in the following ways:

- a) If you have selected a stored document in the viewer, you can query the selection.
To do this, choose *Query selection*.
 - b) You can display the image of the stored document.
To do this, choose the *Display document* icon.
 - c) You can cancel processing.
To do this, choose *Cancel*.
3. Choose *Select*.

The hit list of stored documents is displayed.



As of R/3 Release 4.0A, the list display corresponds to the SAP standard layout, which means that all standard filter and sort functions can be used.

4. Select one or more stored documents.
5. You can continue in the following ways:
 - a) You can display the image of the stored document by choosing *Display from content server*.
 - b) You can display the link entries in detail. To do this, choose *Display detail*.
 - c) You can display the business object, which belongs to the stored document. Choose *Display business object*.
 - d) To display the current status, choose the *Status query* icon.
 - e) To delete a stored document, choose *Stored documents* → *Delete*.



A confirmation prompt is displayed in which you must confirm the deletion process. If you confirm, the relevant stored document is displayed for security reasons. The link is then deleted.

Linking Business Objects with Stored Documents

When the last link is deleted, the stored document can no longer be accessed, irrespective of whether it is deleted from the content server itself.

Once the stored document can no longer be accessed, it can be deleted from the content server. If you answer the second confirmation prompt in the negative (whether the document in the content server should be deleted) the document can no longer be accessed, but can continue to exist.



You can check via an authorization check whether you have the authorization to delete. If you do not have this authorization, the delete icon is hidden.

6. In the detail display, you can continue in the following ways:
 - a) You can display the image of the stored document by choosing *Display from content server*.
 - b) You can display the business object, which belongs to the stored document. Choose *Display object*.
 - c) You can display the list of links. Choose *Stored documents* → *Links*.
 - d) To start a workflow task, choose *Stored documents* → *Start task*.
 - e) You can change the document type by choosing *Stored documents* → *Change document type*.
 - Enter the new document type and - if required - a retention time.
 - Choose *Stored documents* → *Change document type* to make the change.
The current specifications are displayed under the group header "Link entry".
 - f) To delete a stored document, choose *Stored documents* → *Delete*.

Administration of Incoming and Outgoing Documents

Administration of Incoming and Outgoing Documents

The administrative functions for stored incoming and outgoing documents are explained in this section. You can:

- Search for stored documents

([Searching for Incoming and Outgoing Documents \[Page 144\]](#))

This function enables you to search for stored incoming and outgoing documents in the content server and to display them in the viewer.

- Display stored documents

This function enables you to display stored documents via link entries

([Displaying Link Entries for Stored Documents \[Page 146\]](#))



If there is a large number of stored documents, this function may take a long time.

- Create an incoming document

([Creating an Incoming Document \[Page 148\]](#)),

This function contains the storage strategy "Storing and entering" an incoming document.

- Assign an stored document to another business object

([Reassigning an Stored Document \[Page 149\]](#))

This function enables a stored document to be reassigned to another business object.

The link between stored document and the original business object is deleted.

- Assign the stored document to a business object subsequently ([Assigning an Stored Document Subsequently \[Page 150\]](#))

This function enables an incoming or outgoing document that has already been stored to be assigned to a business object subsequently.

- and display the administration information for a stored document.

From the administration information, you can:

- Display link entries for stored documents,
- Display business objects for stored documents
- Change the document type of an stored document and
- Delete the link between stored document and business object.
- Start a workflow task.

Searching for Stored Documents

Searching for Stored Documents

Procedure

To search for incoming and outgoing stored documents in the content server using SAP ArchiveLink administration, proceed as follows:

1. From administration of stored documents, choose
Stored documents ? *Documents Find documents*.
2. Enter:

- *Business object*

To further limit the search, you can specify the following information:

- *Client*

Default is the client, in which you logged on in the R/3 System.

- *Content server*

- *Document ID*

- *Document type*

- *Document class*

- *Storage date from ...*

If a date is entered here, the system will only search for stored documents for which an entry was made in the link table on this date.

- *Storage date to ...*

- *Deletion date*

3. Choose *Program Execute*.

A dialog box is displayed asking you to enter key fields for a particular business object.

4. Leave these fields blank or specify the relevant business object and/or its number and choose *Continue*.



When searching you can use part entries and the wildcards * and ?.

Since the system searches for a summarized key, the result may be a larger list than expected.

The object key is created from the key fields and entered in the relevant field.

Result

A list is displayed of all documents which fulfil the search criteria.



Searching for Stored Documents

As of R/3 Release 4.0A, the list display corresponds to the SAP standard layout, which means that all standard filter and sort functions can be used. For more information, see [ABAP List Viewer \[Ext.\]](#).

You can display a stored document using the icon *Display from content server*.

You can display the relevant business object using the icon *Display business object*.

You can display the status for each entry in the hit list using the icon *Status query*.

You can display the link entries for stored document (see [Displaying Link Entries for Stored Documents \[Page 146\]](#)).

Displaying Link Entries for Stored Documents

Displaying Link Entries for Stored Documents

Procedure

To view link entries for stored incoming and outgoing documents using SAP ArchiveLink administration, proceed as follows:

1. From administration of stored documents, choose *Stored documents* → *Documents* → *Display documents*.

A list of all stored documents is displayed.



If many documents are stored, you should choose *Search for documents* instead of *Display documents* so that you can limit the list via the search criteria.

As of R/3 Release 4.0A, the list display corresponds to the SAP standard layout, which means that all standard filter and sort functions can be used.

To display a stored document, choose the *Display from content server* icon.

You can display the relevant business object using the icon *Display business object*.

2. Select an entry by double-clicking.

The link entries for the stored document are displayed:



For R/3 Release 4.0A, the link entries also include the status of the content server (online/offline) as well as the archiving date and time.

3. You can continue in the following ways:

- a) You can display the image of the stored document by choosing *Stored documents* → *Display document*.
- b) You can display the list of all business objects linked to one stored document. Choose *Stored documents* → *Display links*.
The fields that are displayed for each link entry can be customized.
- c) You can display the business object, which belongs to the stored document. Choose *Stored documents* → *Display object*.
- d) You can change the document type by choosing *Edit stored* → *Change document type*.
 - Enter the new document type and - if required - a retention time.
 - Choose *Continue*.The current specifications are displayed under the group header "Link entry".
- e) To start a workflow task, choose *Stored documents* → *Start task*.
- f) To delete the stored document, choose *Stored documents* → *Delete*.



Displaying Link Entries for Stored Documents

A confirmation prompt is displayed in which you must confirm the deletion process. If you confirm, the relevant stored document is displayed for security reasons. The link is then deleted.

When the last link is deleted, the stored document can no longer be accessed, irrespective of whether it is deleted from the content server itself.

Once the stored document can no longer be accessed, it can be deleted from the content server. If you answer the second confirmation prompt in the negative (whether the document in the content server should be deleted) the document can no longer be accessed, but can continue to exist.



You can check via an authorization check whether you have the authorization to delete. If you do not have this authorization, the delete icon is hidden.

- g) You can query the current status using the icon *Status query*.

Creating an Incoming Document

Procedure

To store an incoming document in a content server, using the storage scenario "Storing and entering", proceed as follows:

1. From administration of stored documents, choose
Stored documents ? Documents Create
2. Enter the business object and document type.
3. Choose *Stored documents Create*

A dialog box is displayed asking you to enter key fields. The object key is created from the key fields and entered in the relevant field. Enter the object key specific to the object type.

See also: [Storing and Entering \[Page 98\]](#)



As of R/3 Release 4.0A, you can store documents generically using object services (see [Generic Object Services \[Page 89\]](#) and [Storing Documents Generically \[Page 109\]](#)).

Reassigning a Stored Document

Procedure

To reassign a stored document, proceed as follows:

1. From the initial screen of administration of stored documents, choose *Stored documents* → *Documents* → *Reassign*.
2. Enter the following:
 - Business object for the initial object
 - Business object for the target object
 - Document type for the target object
 - (Optional) the retention period for the target object

The **initial object** is the business object to which the stored document is still assigned.

The **target object** is the business object to which the stored document is to be assigned.

4. Choose *Stored documents* → *Reassign*.
A dialog box is displayed.
5. Enter the key fields for the initial object in the dialog box and confirm.
The initial object is displayed for you to check.
6. Confirm the business object.
A dialog box is displayed.
7. Enter the key fields for the target object and choose *Continue* to confirm.
The target object is displayed for you to check. Once it has been confirmed, the appropriate link entry is created.

Assigning a Stored Document Subsequently

Assigning a Stored Document Subsequently

Procedure

To assign a stored document to a business object subsequently, proceed as follows:

1. From the initial screen of administration of stored documents, choose *Stored documents* → *Documents* → *Assign subsequently*.
2. Make entries for the target object (business object) regarding:
 - Business object
 - Document type
 - (Optional) retention time
4. Choose *Stored documents* → *Assign subsequently*.

A dialog box is displayed.

5. Enter the key fields for the business object and choose *Continue* to confirm.

The business object is displayed for you to check. Once it has been confirmed, the appropriate link entry is created.



See also [Generic Object Services \[Page 89\]](#) and [Assigning Documents Generically \[Page 108\]](#).

Administration of Print Lists

This section describes the administrative functions for stored print lists. You can:

- Search for stored print lists ([Searching for Stored Print Lists \[Page 152\]](#)) and
- Display the relevant link entries and print out the print lists ([Displaying Link Entries for Stored Print Lists \[Page 153\]](#)).



As of R/3 Release 4.5A, it is possible to process notes for stored print lists, in the R/3 System. You can display, create and enhance notes. To be able to do this, you must be using the SAP Content Server HTTP interface 4.5 and display the print list in the R/3 System.

Storing

A print list is generated in an R/3 application component (see the [SAP ArchiveLink - Scenarios in Applications \[Ext.\]](#) documentation). The print list is transferred to the spool for printing and storing. The spool generates a data file for the report output to be stored and a description file for any accompanying index information and instructs SAP ArchiveLink to store. SAP ArchiveLink transfers the storage request to the relevant content server.

Confirmation

The content server reports successful storing to SAP ArchiveLink. SAP ArchiveLink then enters a list object in SAP ArchiveLink list administration.



For more information on print lists and index information and hyperlinks, see [Storing Print Lists \[Page 219\]](#).

Searching for Stored Print Lists

Searching for Stored Print Lists

Procedure

To search for stored print lists in a content server via administration, proceed as follows:

1. From administration of stored documents, choose
Stored documents → *Print lists* → *Find print lists*.
2. Enter (if known):
 - *Client*
Default is the client, in which you logged on in the R/3 System.
 - *Report name*
The report that generated the print lists.
 - *Info*
If the report name is not unique, an entry can be made here.
 - *Storage date*
If you enter dates, the system only searches for print lists which were stored within the specified period.
 - *Business object*
 - *Document type*
 - *User*
 - *Short text*
 - *Content server*
 - *Document ID*
- If you want to search for all existing stored print lists, leave the fields blank and choose *Program* → *Execute* directly.
3. Choose *Program* → *Execute*.

Result

The stored print lists that fulfill the search criteria are displayed.



As of R/3 Release 4.0A, the list display corresponds to the SAP standard layout, which means that all standard filter and sort functions can be used. For more information, see [ABAP List Viewer \[Ext.\]](#).

You can display a print list using the icon *Display from content server*. You can display the status for each entry in the hit list using the icon *Status query*.

You can display the link entries for stored print list (see [Displaying Link Entries for Stored Print Lists \[Page 153\]](#)).

Displaying Link Entries for Stored Print Lists

Procedure

To display link entries for stored print lists, proceed as follows:

1. From administration of stored documents, choose *Stored documents* → *Print lists* → *Display*.

A list of all stored print lists is displayed.



As of R/3 Release 4.0A, the list display corresponds to the SAP standard layout, which means that all standard filter and sort functions can be used. For more information, see [ABAP List Viewer \[Ext.\]](#).

You can display the print list using the icon *Display from content server*.

You can display the status using the icon *Status query*.

2. Select an entry by double-clicking.

The link entry for the stored print list is displayed:



For R/3 Release 4.0A, the link entries also include the status of the content server (online/offline) as well as the archiving date and time.

3. You can continue in the following ways:

- a) You can display the **image** of the archived print list by choosing *Print lists* → *Display from content server*.



As of R/3 Release 4.5B, print lists are displayed, as standard, directly in the actual dynpro.

When displaying in the R/3 System, **hyperlinks** and **colors** can be displayed.

The following **restrictions** apply, but do not, however, relate to the HTTP Content Server interface 4.5 (see below):

- An attribute search is not possible.
- A free search of the whole list is not possible.
- The scroll bar can only be used to position in the currently displayed area. The scroll keys in the menu bar, however, relate to the whole list.



When displaying in the R/3 System, free search and attribute search (for indexed print lists) are possible, if the content server is addressed using the SAP Content Server HTTP interface 4.5 (see also [4.5 HTTP Content Server Interface \[Page 229\]](#)). Likewise, you can scroll using the scroll keys in the menu.

Displaying Link Entries for Stored Print Lists

When displaying in the R/3 System, processing (displaying, creating and enhancing) notes is possible, if the content server is addressed via the SAP Content Server HTTP interface 4.5.

You can also navigate to the start or end of the stored list by choosing *Goto → Start of list* or *End of list*.

- b) You can delete the print list by choosing *Print lists → Delete*.

Two confirmation prompts appear:

- Confirm that you wish to delete the list.
The links are deleted. The list can no longer be accessed.
- Confirm that you wish to delete the list in the content server.
Once the list can no longer be accessed, it can be deleted from the content server. If you answer the second conformation prompt in the negative, the list can no longer be accessed, but continues to exist.



You can check via an authorization check whether you have the authorization to delete. If you do not have this authorization, the delete icon is hidden.

- c) You can print the relevant print list by choosing *Print lists → Print from content server*.



As of R/3 Release 4.0A, there is an authorization check for printing a print list. If you do not have authorization to print, the print icon is hidden.

It is possible to print again, either all or part of a print list that is already stored.

A display frame appears inquiring whether the whole or part of the list is to be printed. If you choose to print a page range, a further dialog box appears in which you specify the page range.

When an option has been chosen, a screen appears in which you enter the print parameters.

The chosen parameters are evaluated internally and processed further and the stored print list is retrieved from the content server again. The report output is restarted in the format in which it was stored.

- d) You can display the current status using the icon *Status query*.

Administration of Archive Files

This section describes the administrative functions for stored archive files from the **Archive Development Kit (ADK)**. You can:

- Search for stored archive files and
- Display the relevant link entries ([Searching for Stored Archive Files \[Page 156\]](#)).



Information on deleting and archiving application data via the ADK can be found in the [Archiving and Deleting Application Data \[Ext.\]](#) documentation.

Storing

An archive file is generated in an R/3 application component via the **Archive Development Kit (ADK)** (see the [SAP ArchiveLink - Scenarios in Applications \[Ext.\]](#) documentation). With appropriate Customizing, the ADK instructs SAP ArchiveLink to store. SAP ArchiveLink transfers the storage request to a content server.

Confirmation

The content server reports successful storing to SAP ArchiveLink. SAP ArchiveLink then informs the ADK of the successful storing and generates the link between the ADK run and the stored archive files.

You can access sub-objects of a large archive file via the CGDB call (see [CGDB \[Ext.\]](#)). A prerequisite for this is that the archive file be indexed via the relevant R/3 application component.

High-performing single access is thus possible.

Searching for Stored Archive Files

Searching for Stored Archive Files

Procedure

1. From administration of stored documents, choose
Stored documents → *Archive files* → *Find archive files*.
2. Enter the following, if known:
 - *Client*
The default is the client with which you logged on to your R/3 System.
 - *Document type*
The document type for archive files is **ARCHIVE**.
 - *Document class*
 - *Object ID*
3. Confirm with *Program* → *Execute*.

Result

The stored archive files that fulfill the search criteria are displayed.



As of R/3 Release 4.0A, the list display corresponds to the SAP standard layout, which means that all standard filter and sort functions can be used. For more information, see [ABAP List Viewer \[Ext.\]](#).



You can display an archive file using the icon *Display from content server*.

You can display the relevant business object using the icon *Display business object*.

You can display the link entries for stored archive file (see [Displaying Link Entries for Stored Archive Files \[Page 157\]](#)).

Displaying Link Entries for Stored Archive Files

Displaying Link Entries for Stored Archive Files

Procedure

1. From administration of stored documents, choose *Stored documents* → *Archive files* → *Display*.

A list of all stored archive files is displayed.



As of R/3 Release 4.0A, the list display corresponds to the SAP standard layout, which means that all standard filter and sort functions can be used. For more information, see [ABAP List Viewer \[Ext.\]](#).

You can display the archive file using the icon *Display from content server*.

You can display the relevant business object using the icon *Display business object*.

2. Select one or more entries and choose *Display detail*.

The link entry for the stored archive file is displayed.



For R/3 Release 4.0A, the link entries also include the status of the content server (online/offline) as well as the archiving date and time.

You can query the current status using the icon *Status query*.

3. You can change the document type by choosing *Edit* → *Change document type*.

- Enter the new document type and - if required - a retention time.
- Choose *Continue*.

Authorizations

Authorizations

Roles are the basis for defining user authorizations. Roles generate profiles and authorizations for users. For more information on roles and authorizations, see [Authorizations, Profiles and the Profile Generator \[Ext.\]](#).

Authorizations enable users to perform particular functions in the SAP System. Each authorization relates to an **authorization object**. Authorizations are entered in the user master record, enabling the respective users to work productively with the system.

Authorization objects make possible complex checks, linked to several conditions, of an authorization that allows a user to execute an action.

Authorization profiles encompass a list of authorizations which control user access to various areas of the system. The authorization profiles are defined in the user master record for the respective user when the SAP System is installed.

Authorizations

The R/3 System checks accesses to SAP ArchiveLink functions with the following Basis authorization objects:

- **S_WFAR_OBJ**
- **S_WFAR_PRI**
- **S_BDS_DS**

The SAP ArchiveLink administrator requires certain authorizations for table maintenance:

- **S_TABU_DIS**

To display the authorization objects for SAP ArchiveLink, choose *Development → Other tools → Authorization objects → Objects* in the ABAP Workbench.

The authorization objects for SAP ArchiveLink can be found under “BC_Z” “Basis - Central Functions” and “BC_A” “Basis - Administration”.



For information on maintaining authorizations and authorization objects, see the documentation [BC - Users and Roles \[Ext.\]](#).

S_WFAR_OBJ

For further information, see the authorization object online help.



Of the R/3 authorization objects provided by SAP, only the activity (ACTVT) is already defined. The other fields can be configured by the system administration as required.

S_WFAR_PRI



This authorization object, used to access stored print lists, is available as of R/3 Release 4.5A.

For further information, see the authorization object online help.

S_BDS_DS

This authorization object is used in the following cases:

- Calling the Business Document Navigator (see also [Business Document Navigator \[Page 131\]](#))
- Storing documents using the Business Document Navigator
- Accessing stored documents in the Business Document Navigator

Authorizations

For further information, see the authorization object online help.

S_TABU_DIS

The authorization object S_TABU_DIS is used by SAP ArchiveLink in standard view maintenance. It controls accesses made across Basis via standard table maintenance, extended table maintenance, or the Data Browser, including accesses carried out in the Customizing system and via the Customizing Wizard.

The authorization object S_TABU_DIS consists of the following two fields:

- **DICBERCLS**

Authorization group

This field is used to assign the authorization for tables by authorization class according to table TDDAT.

- **ACTVT**

Activity

This field is used to define permitted operations. The following activities are possible:

- **02**

Add, change, or delete table entries

- **03**

Display table contents



The authorization group that should be checked in the case of maintenance views is SAOP.

The administrator requires S_TABU_DIS with SAOP.

Authorization Checks for SAP ArchiveLink

Function Modules

The following SAP ArchiveLink function modules are subject to authorization checks:

- **ARCHIV_SEND**
Function module for content server messages to the SAP System
- **ARCHIV_BARCODE_INSERT_RFC**
Function module for bar code returns to the SAP System

These function modules are RFC-capable. The system checks whether the relevant RFC connection user has the authorization object S_WFAR_OBJ with the following values in his or her user master record:

- 70 for ACTVT
- * for OAARCHIV
- * for OAOBJEKTE
- * for OADOKUMENT

Transactions

An authorization check on activity 70 is performed (70 for ACTVT) for the **administration transactions** (*Tools → Business Documents*).

For **transactions**

- Find and display stored documents
(*Office → Business Documents → Documents → Find*
see [Searching for Incoming Documents \[Page 91\]](#))
- Find and display stored print lists
(*Office → Business Documents → Find print lists,*
see [Searching for, Displaying and Printing Print Lists \[Page 104\]](#))

an authorization check is performed on the activity 03 (03 for ACTVT).

For the **transaction** for the storage strategies

- *Office → Business Documents → Documents → Store*

an authorization check is performed on the activity 01 (01 for ACTVT).

Functions

For the function **Print stored print list** (see [Displaying Link Entries for Stored Print Lists \[Page 153\]](#)) an authorization check is performed for activity 04 (04 for ACTVT).

For the functions

Authorization Checks for SAP ArchiveLink

- **Delete stored documents**
(see [Displaying Link Entries for Stored Documents \[Page 146\]](#)) and
- **Delete stored print lists**
(see [Displaying Link Entries for Stored Print Lists \[Page 153\]](#))

an authorization check is performed on the activity 06 (06 for ACTVT).

Roles for SAP ArchiveLink

The following role is available for SAP ArchiveLink:

- **SAP_BC_SRV_ARL_ADMIN**

Role for SAP ArchiveLink administrators

Authorizations for SAP ArchiveLink are also contained in the following Basis role:

- **SAP_BC_SRV_USER**

Role containing, amongst other things, normal user authorizations for SAP ArchiveLink.



For more information on SAP ArchiveLink roles, see the Implementation Guide (IMG) under
“R/3”, “Basis”, “Basis Services”, “SAP ArchiveLink”, “Administration Settings”,
“Authorizations” in the section [Working with Roles \[Ext.\]](#).

For more information on roles, see [Role Resolution \[Ext.\]](#).

Information for Persons Making Customizing and System Settings

Information for Persons Making Customizing and System Settings

This section is intended for those making Customizing and system settings.

- Customizing is described.
- The frontend communication possibilities are explained.

Customizing

The following gives an overview of the various Customizing options available in SAP ArchiveLink.



As of R/3 Release 4.0A, all Customizing for SAP ArchiveLink can be carried out via the IMG.

In the IMG, choose
“R/3”, “Basis Components”, “Basis Services”, “SAP ArchiveLink”.

The Customizing activities for SAP ArchiveLink are divided into the following groups in the IMG:

- Basic Settings
- Administration Settings
- System Settings
 - SAP Business Workflow Settings
 - Content Server Settings

Customizing outside Basic Settings should be performed when certain functionality is to be used or standard settings are to be changed or enhanced.

Basic Settings

For storing with SAP ArchiveLink, the maintenance of

- Content servers and
- Links

is necessary. When an HTTP content server is specified, an R/3 certificate must be sent to the content server.

In addition, the following Customizing settings should be made:

- Basic Settings
- Queues
- Jobs
- Number ranges

The following functions are then available for checking:

- Link check
- Result of link check
- Customizing check



For basic Customizing, choose the following in the IMG: “R/3”, “Basis Components”, “Basis Services”, “SAP ArchiveLink”, “Basic Settings”.

For information on maintaining individual activities, refer to the online help in the IMG.

Customizing

Administration Settings

In Customizing for administration-specific settings, the following can be maintained for each storage strategy, in addition to the basic settings:

- Protocols
- Applications
- Authorizations
- Barcode storage

The following function can be used for checking:

- Customizing check



For administration-specific Customizing, choose the following in the IMG: "R/3", "Basis Components", "Basis Services", "SAP ArchiveLink", "Administration Settings".

For information on maintaining individual activities, refer to the online help in the IMG.

System Settings

In Customizing for system-specific settings, the following can be maintained for each storage strategy, in addition to the basic settings:

- Link Tables
- Bar code types
- Document classes
- Document Types

The following function can be used for checking:

- Customizing check



For SAP Business Workflow, choose the following in the IMG: "R/3", "Basis Components", "Basis Services", "SAP ArchiveLink", "System Settings".

For information on maintaining individual activities, refer to the online help in the IMG.

SAP Business Workflow Settings

In Customizing for SAP Business Workflow, the following can be maintained for each storage strategy, in addition to the basic settings:

- Document Types
- Default values for storage strategies
- Workflow basic settings
- Workflow document types
- Workflow parameters

- User content server

The following function can be used for checking:

- Customizing check



For SAP Business Workflow Customizing, choose “R/3”, “Basis”, “Basis Services”, “SAP ArchiveLink”, “SAP Business Workflow Settings” in the IMG.

For information on maintaining individual activities, refer to the online help in the IMG.

See also: [Information for SAP Business Workflow \[Page 205\]](#)

Content Server Settings

In Customizing for content servers, the following can be maintained for each storage strategy in addition to the basic settings:

- Protocols
- Applications



For content server-specific Customizing, choose the following in the IMG: “R/3”, “Basis”, “Basis Services”, “SAP ArchiveLink”, “Content server settings”.

For information on maintaining individual activities, refer to the online help in the IMG.

Basic Settings

Basic Settings

The following must be maintained in Basic Settings for SAP ArchiveLink.

- **Content Servers**
- **Links**

The settings are made via view-controlled tables.



For a description of view-controlled tables, see the *BC - System Services* documentation, section [Extended Table Maintenance \[Ext.\]](#).

When content servers and links are determined, the assignment of the business object generated in the R/3 application to the stored document assigned, takes place in the link table itself during operation.

A single business object can be assigned to several stored documents. In this case, the object key is identical.

For every stored document, there must be a document ID, that is, the document ID must be unique.

For more information, see [Administration Concept \[Page 51\]](#).



If an HTTP content server was specified in content server maintenance, an R/3 certificate must be sent to the content server. Choose *Tools* → *Business Documents, Environment* → *Knowledge Provider*, *Edit* → *Send certificates*.

Default Settings

- **Object types** are provided in the standard version.
- **Document types** are provided in the standard version and can be modified and extended by the customer as required within the customer name ranges.
- **Document classes** are provided in the standard version and can be modified and extended by the customer as required within the customer name ranges.
- **Content servers** and **protocols** are not supplied.
- **Links** are only provided as examples in customer client 000. Depending on requirements, customers can copy these links or modify them individually.

Specific Customizing

The maintenance of the settings that are not in Basic Settings depends on which functions you want to use within SAP ArchiveLink.

These settings can be found at the following places in the system:

- In the Implementation Guide (IMG)
- In SAP ArchiveLink administration

To go to SAP ArchiveLink Administration, choose:
Tools → Business Documents

Document classes

Document classes control the storing and display of documents of the appropriate document type. This control is exerted via protocol maintenance.

You can create document classes in addition to those already existing.

You can create document classes in the following ways:

- In the IMG

For this, choose the following in the IMG
“R/3”, “Basis”, “Basis Services”, “SAP ArchiveLink”, “System Settings”, “[Maintain Document Classes \[Ext.\]](#)”.

- In SAP ArchiveLink administration

Choose
Basic Settings → Customizing → Document classes

For information on activity maintenance, refer to the online help in the IMG.

Link Tables

If you want to create further link tables in addition to those already defined, the new ones must have the same structure as the standard ones (see [Link Tables \[Page 55\]](#)).

You can create link tables in the following ways:

- In the IMG

For this, choose the following in the IMG
“R/3”, “Basis”, “Basis Services”, “SAP ArchiveLink”, “System Settings”, “[Maintain Link Tables \[Ext.\]](#)”.

- In SAP ArchiveLink administration

Choose
Basic Settings → Customizing → Link tables

For information on activity maintenance, refer to the online help in the IMG.

Storing with Bar Codes

For storing with bar codes, certain administrative settings are necessary.

Specific Customizing

You can activate storing with bar codes in the following ways:

- In the IMG

In the IMG, choose
“R/3”, “Basis”, “Basis Services”, “SAP ArchiveLink”, “Administration Settings”, [Activate Storing with Bar Codes \[Ext.\]](#).

For information on activity maintenance, refer to the online help in the IMG.

- In SAP ArchiveLink administration

For this, choose
Basic Settings → *Bar codes* → *Bar code entry*



For technical information on integrating storing with bar codes into your R/3 application component, see [Storing with Bar Codes \[Page 212\]](#).

Bar code types

The bar code type is the type name of the current bar code. The following bar code types are supplied as standard (see [Bar Codes \[Page 82\]](#)):

- CODE_2_5_I
- EAN_13
- EAN_8
- UPC_A
- UPC_E

The list of these bar codes supplied as standard can be extended as required.

You have the following options for creating bar code types:

- In the IMG

For this, choose the following in the IMG
“R/3”, “Basis”, “Basis Services”, “SAP ArchiveLink”, “System Settings”, [Defining Bar Code Types \[Ext.\]](#).

- In SAP ArchiveLink administration

For this, choose: *Basic Settings* → *Bar codes* → *Bar code types*

For information on activity maintenance, refer to the online help in the IMG.



Specifying bar code types is necessary with regard to automatic checking.

Document Types

For all R/3 applications where SAP ArchiveLink is integrated, standard document types are supplied.

Specific Customizing

You require your own document types in the following cases, for example:

- If you store print lists and want to structure according to particular document types
- If you want to store new document types



The [SAP ArchiveLink Document Types Customizing Wizard \[Page 81\]](#) is available as of R/3 Release 4.0A.

The SAP ArchiveLink document types Customizing Wizard leads the user step-by-step through the maintenance of all settings required to create new document types.

For further information, refer to the Wizard online help.

You can create document types in the following ways:

- In the IMG

For this, choose the following in the IMG
“R/3”, “Basis”, “Basis Services”, “SAP ArchiveLink”, “System Settings”, “[Maintain Document Types \[Ext.\]](#)”.

or

“R/3”, “Basis”, “Basis Services”, “SAP ArchiveLink”, “Business Workflow Settings”, “Call document types Customizing Wizard”.

- In SAP ArchiveLink administration

For this, choose
Document types → Global doc. types

or

Document types → Customizing Wizard

For information on activity maintenance, refer to the online help.

See also:

[Further Default Settings \[Page 207\]](#)

Communication

Communication

Communication includes both protocols and application maintenance.

Using **protocol maintenance**, you can define how certain SAP ArchiveLink functions, such as Display, Archive, and Retrieve, can be handled for individual document classes.

Using **application maintenance**, you can define according to function, the sequence of the calls relocated by the R/3 System to the specified partner application, in order to handle a particular function.

Protocols

Protocols are defined in order to control communication according to the document class. For the individual document classes, you can determine how (that is, using which communication protocol) every SAP ArchiveLink function, for example, Display, Archive and Retrieve should be handled.

By assigning

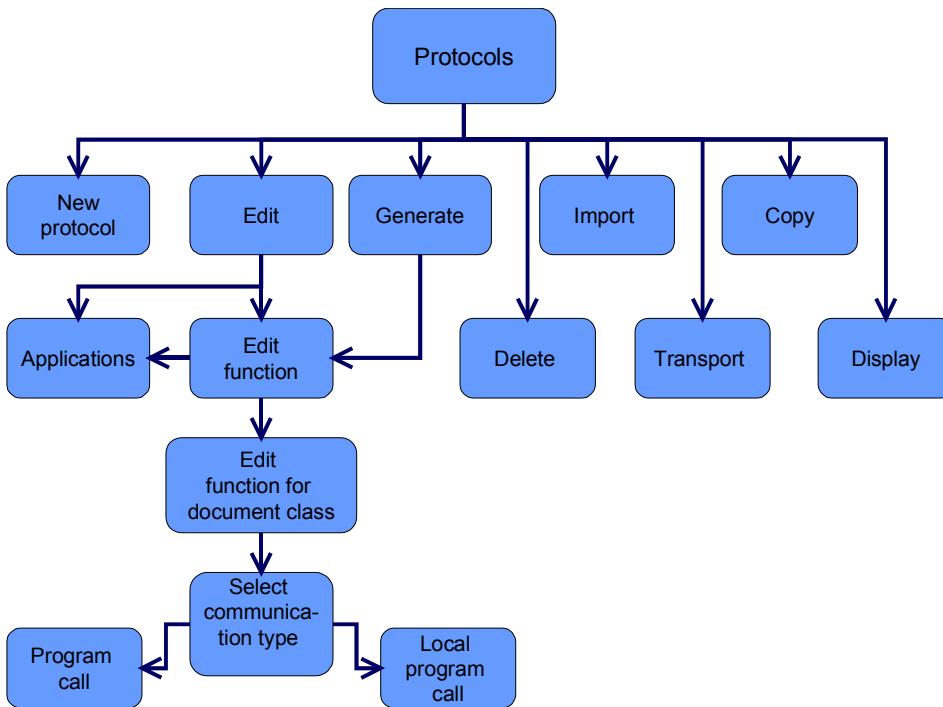
- document classes to document types and
- document types to business objects from R/3 applications.

you establish the relationship between the protocol definition and the business objects.



Protocol maintenance always refers to document classes.

Process Flow



You can maintain protocols in the following ways:

- In the IMG

For this, choose the following in the IMG
"R/3", "Basis", "Basis Services", "SAP ArchiveLink", "Administration Settings" or "Content server settings", "Maintain protocols".
- In the system

Protocols

For this, choose
Tools → *Business documents*,
Basic Settings → *Communication* → *Protocols*

For information on activity maintenance, refer to the online help in the IMG.

Protocols: Concept

The R/3 server communicates with content servers via RPC (Remote Procedure Call) or RFC (Remote Function Call) or HTTP (Hypertext Transfer Protocol) for file access.



As of R/3 Release 4.0A, the function ***Start external application*** is available (see [Start External Application \[Page 316\]](#)).

As of R/3 Release 4.6A, protocol maintenance is simplified: The backend functions are therefore no longer available for protocol maintenance:

- Synchronous Archiving Server
- Synchronous Archiving Table
- Asynchronous Archiving Server
- Synchronous Retrieval Server
- Synchronous Retrieval Table
- Synchronous retrieval byte stream
- Asynchronous Retrieval Server
- Status query
- Deleting a stored document
- Format stored document

Example: Protocol and possible functions

Protocols: Concept

| Example | Interface version 0045 | Example Protocol |
|----------------------------|--|------------------|
| Functions | | |
| Display stored document | | |
| | → Retrieve for frontend → Display local file | |
| Store from frontend | | |
| | → Store file on frontend → Store file from frontend | |
| Close window | | |
| Start external application | | |



As of R/3 Release 4.0A, it is possible for protocols to be generated by the system (see [Generating Protocols \[Page 308\]](#)).

The functions for storing or displaying can:

- be carried out directly using one function (or in one step) (for example, ICC) or
- be carried out using a combination of two functions (for example, a file to be stored can be retrieved via OLE to the frontend and then stored in the content server by the R/3 server via RFC).

The protocols are assigned to a content server in content server maintenance.



As of R/3 Release 4.5A, there is a **connection to the Change & Transport System** for protocols (*Protocol → Transport*).

You can also copy protocol definitions from other systems that can be accessed via RFC (*Protocol → Import*). This enables **remote Customizing** of the protocols.

So that users do not hinder each other during protocol maintenance, there is a **block against multiple access** for the protocols as of Release 4.5A. If, for example, a protocol is already being processed and another user wants to process this protocol, the second user receives an error message informing that the protocol is currently being processed by the first user.

As of R/3 Release 4.6A, there is only one configuration for each protocol. The platform to be used is Windows 32. This means that the frontend functions are no longer frontend-dependent.

Standard communication

The following standard communication is defined for the various functions for Windows 32:

| Function | Standard communication |
|--------------------------|--|
| Display stored document | ICC (for ALF, FAX and OTF), otherwise Archivelnk |
| Retrieval for frontend | Archivelnk |
| Display local file | ICC (for FAX), otherwise Archivelnk |
| Archive from frontend | ICC |
| Store file on frontend | Archivelnk |
| Store file from frontend | Archivelnk |
| Close Window | ICC |



Generally, you must enter the communication type **HTTP** for an HTTP content server.

Generally, you must enter the communication type **Archivelnk** for a file content server.

Elements

The central element of a protocol is protocol maintenance. You can maintain protocols in the following ways:

- In the IMG

For this, choose the following in the IMG
 "R/3", "Basis", "Basis Services", "SAP ArchiveLink", "Administration Settings" or "Content server settings", "[Maintain protocols \[Ext.\]](#)"

- In the system

For this, choose
Tools → *Business documents*,
Basic Settings → *Communication* → *Protocols*

For information on activity maintenance, refer to the IMG online help.

Application Scenarios

The following examples are application scenarios for which protocols must be maintained:

- Storing and displaying PC files (for example in MS Word, MS Excel, MS PowerPoint, etc.)
- Using external entry dialogs (a part of the content server which is not integrated)
- Using external viewers (a part of the content server which is not integrated)

Protocols: Concept

- Using external scan dialogs, if these are called via OLE Automation 2.0
- Using external viewers, if these are called via OLE Automation 2.0.

Non-application scenarios

You do not have to maintain protocols for the following application scenarios:

- You can only store incoming documents by storing with bar codes.
- Using internal scan dialogs which you can call via ICC
- Using internal viewers which you can call via ICC

Generating Protocols



Generating protocols is available as of R/3 Release 4.0A. Currently, only entries for the display of documents are generated.

Purpose

You can create a **standard protocol**. In this case, entries for document classes DOC, PDF, TIF, HTM and XLS are generated. The application registered for the relevant document class is started when you display.

You can also generate **individual entries** for the following applications and the corresponding document classes:

| Application | Document classes |
|---------------------------------------|-----------------------------------|
| MS Word | DOC |
| MS Excel | XLS |
| Wang Imaging for Windows NT | TIF, BMP |
| Graphic Converter for Apple Macintosh | TIF |
| MS Internet Explorer | TIF, DOC, XLS, PDF, HTM, GIF, JPG |

Prerequisites

You must create a protocol before you can start generating protocol entries.

Process Flow

Individual Entries

1. You must make specifications regarding:
 - Application and/or
 - Document class
2. You can then make specifications regarding:
 - Display local document
 - Execute application
 - Via EXECUTE or
 - Via OLE
3. Default values are suggested by the system for the document class and the display function.
4. You can change these default values or have the system make the respective entries via *Generate*.

Generating Protocols

Result

The system has generated the entries for your protocol in accordance with your individual requirements.

Storing

The following customizable functions belong to the **storing** area:

- Archive from frontend
- Store file on frontend
- Store file from frontend
- Starting an External Application



This function is available as of R/3 Release 4.0A (see [Starting an External Application \[Page 316\]](#) [Page 316]).

You can store a document generated on the frontend, that is an incoming document, in either **one or two steps**.

- Single-step procedure:

A request to scan and store a document is sent to an content server-internal scan dialog.

- Two-step procedure:

If the program with which the document is scanned cannot store directly, the function is split up into the following two steps:

1. Store file on frontend
2. Store this file from the frontend into the content server



MS Word cannot store in an external content server. Thus you store WinWord files with the following two steps:

1. Archive file on the frontend:

A request is sent to WinWord to save the file that is currently being processed and to report the file name to the R/3 System.

2. Store this file from the frontend into the content server

The file is imported into the R/3 System and transferred to the content server from the R/3 application server, or the file name is transferred via OLE Automation 2.0 to a frontend component of the content server.

Storing Using Internal Entry Dialog

Storing Using Internal Entry Dialog

When you store using content server-internal entry dialogs via OLE Automation, you only have to set function **Store from frontend**. The functions *Store file on frontend* and *Display file from frontend* need not be explicitly customized.

Enter **OPEN** as the communication type after selecting the function and the document class. This acts as the generic term for OLE Automation.

When communicating using **OPEN**, you need to define a logical name for the application in question. This name can be assigned freely and is entered here while you maintain the precise communication commands for the relevant application later on in application maintenance. Every time you maintain a function that is to be handled using the communication type **OPEN**, you must enter the application.

Function: Archive from Frontend

When the function *Archive file from frontend* is executed, the R/3 System makes a request to an content server entry dialog to store the current document (entered or processed). An identification unique to the content server, *arch_doc_id*, is returned by the content server entry component to the R/3 System. Using the R/3 System, you can transfer the variable *archiv_id* to the entry component in order to allow the entry component to carry out possible checks if, for example, the *archiv_id* is not suitable for the entry component.

Storing PC Files, and Documents Entered Using External Entry Dialogs

Storing PC Files, and Documents Entered Using External Entry Dialogs

In the two-step storing procedure, you should first define that the function **Archive from frontend** is made up of the following two steps:

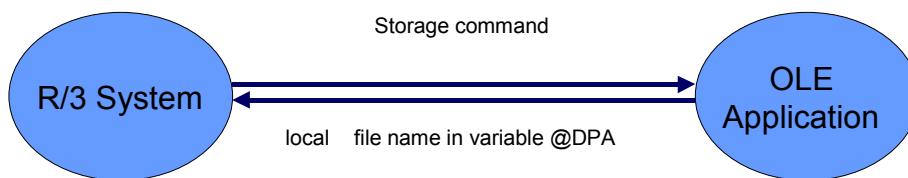
- **Store file on frontend** and
- **Store file from frontend**

In the following steps, you set up the procedure for the functions **Store file on frontend** and **Store file from frontend**.

After selecting the function **Store from frontend** and the document class, enter **ARCHIVELNK** as the communication type. This means that the function **Store from frontend** is split up into the subfunctions **Store file on frontend** and **Store file from frontend**.

Function **Store file on frontend**

This function is used to transfer the name of the file to be stored, to the R/3 System: The R/3 System finds out the name of the file currently being processed from a frontend application (absolute path incl. file name). The frontend application returns the absolute file name of the file currently being edited. Via the R/3 System, it is possible to transfer the variable **archiv_id** to the entry component to allow the entry component to carry out possible checks if, for example, **archiv_id** does not fit in with the entry component.



For the file resulting on the frontend, you must check whether the editing/creating application supports OLE Automation as a server. Examples of such applications are WinWord, MS Excel or internal scan dialogs that were certified for R/3 Release 3.0.

Storing PC Files, and Documents Entered Using External Entry Dialogs

- **OLE application**

After selecting the function *Store file on frontend* and the document class, enter **OPEN** as the communication type.

- **Non-OLE application**

If files that have not been edited/scanned/generated by an OLE-capable application are to be stored, you can enter the file name of the file to be stored in the R/3 System.

After selecting the function *Store file on frontend* and the document class, enter **ARCHIVELNK** as the communication type.



Example of application maintenance for the function *Store file on frontend*

Microsoft Word 7.0 for Office 95 is used here as an example.

The OLE class name of the object used is "Word.basic".

The following calls are executed:

- Call method "TOOLOPTIONSSAVE" with parameters 0, 0, 0.
This deactivates any dialog boxes for the following "FILESAVE".
- Call method "FILESAVE". You call this in order to ensure that the current status of the document is also stored.
- Call method "FILENAME"
This ascertains the name of the current file.
- This would normally suffice because SAP ArchiveLink contains the necessary values. The following method calls are required because it is not always possible for one application to open and read a file which has been opened by another application.
- Call method "FILESAVEAS C:\SAPFILE"
This saves the current edited file under the name C:\SAPFILE.
- Call method "FILENAME"
This ascertains the name of the current file (in this case C:\SAPFILE).
- Call method "FILECLOSE"
This closes the current file (in this case C:\SAPFILE).
- Call method "FILEOPEN"
This call reopens the original file.
- The administration in SAP ArchiveLink is as follows:

Method: **TOOLOPTIONSSAVE 0 ,0 ,0**

Method **FILESAVE**

Method **FILENAME = @FPA**

The name of the current file is ascertained and assigned to the SAP ArchiveLink variables **@FPA** (frontend path)

Method **FILESAVEAS C:\SAPFILE**

Method **FILENAME = @DPA**

The name of the current file (in this case C:\SAPFILE) is ascertained and assigned to the SAP ArchiveLink variables **@DPA**

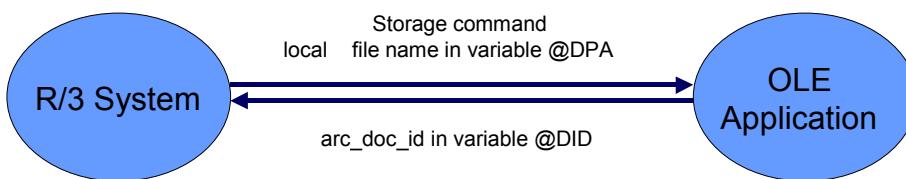
Storing PC Files, and Documents Entered Using External Entry Dialogs

Method **FILECLOSE**

Method **FILEOPEN @FPA**

Function Store file from frontend

This function is used to store the file whose name was transferred to the R/3 System by the function *Store file on frontend*: The R/3 System transfers the absolute file name of the file determined in the function, to the application, requests the application to store and expects a unique ID for the stored document to be returned. Via the R/3 System, it is possible to transfer the variable *archiv_id* to the entry component to allow the entry component to carry out possible checks if, for example, *archiv_id* does not fit in with the entry component.



It is necessary to ask whether the frontend has an internal component that is capable of OLE and to which files can be transferred for storing.

- **OLE-capable components:**

After selecting the function *Store file from frontend* and the document class, enter **OPEN** as the communication type.

- **Non-OLE-capable components:**

The file must be imported into the R/3 System, and stored via the application server. In this case, after selecting the function *Store file from frontend* and the document class, enter **ARCHIVELNK** or **TABLE** as the communication type.

- **ARCHIVELNK**

ARCHIVELNK here means that the file is imported from the frontend and transferred as a file to the content server via server communication.

The setting **ARCHIVELNK** is also supported by systems certified after R/3 Release 2.1.

Storing PC Files, and Documents Entered Using External Entry Dialogs

- **TABLE**

TABLE here means that the file is imported from the frontend and transferred as an internal table to the content server via server communication.

The setting **TABLE** requires an content server that was certified after R/3 Release 3.0.

Starting an External Application



This function is available as of R/3 Release 4.0A.

This function is used for starting an external application for archiving from the frontend.

Protocol maintenance is necessary for this function, if

- An external application is to be started,
- Manual input is to be made, and
- An action is to be executed again afterwards.



The Wang viewer can be started via the function *Create links → And store document* within generic object services (see [Storing Documents Generically \[Page 109\]](#) [Page 109]). A document can then be scanned. Then the scanned document can be stored via *Transfer*.

Only the communication type OPEN, that is, communication via OLE, is suitable for protocol maintenance for document class TIF.

To address an external scan dialog, start the Wang viewer as display component and then transfer a scanned document from the Wang viewer, the following should be specified for application maintenance for the function *Start external application*:

Application: WANGIMAGE.APPLICATION

Method: CreateImageViewerObject = IMG

Method: IMG.New

Display

Display

The following customizable functions belong to **Display**:

- Display stored document
 - Retrieval for frontend
 - Display local file
- Close Window

You can display a stored document in **one or two steps**.

- Single-step procedure:
A request to display a document is sent to an internal viewer.
- Two-step procedure:
If the program that is to display the document cannot directly access the content server, the function is split up into the following two steps:
 1. Retrieve the stored document in a file on the frontend
 2. Display this file



MS Word cannot access a content server. Displaying stored documents is therefore carried out in the following two steps:

1. Retrieve the stored document in a file on the frontend:
A request is sent to an content server component, instructing it to retrieve the stored document as a file on the frontend.
2. Display this file:
WinWord is called via OLE, to display the file retrieved.

Displaying Using Internal Viewers and OLE Automation

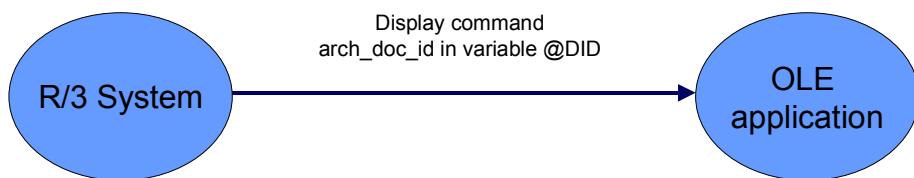
Displaying Using Internal Viewers and OLE Automation

You only have to set the function **Display stored document**. The functions *Retrieval for frontend* and *Display local file* need not be explicitly customized.

After selecting the function and the document class, enter **OPEN** as the communication class.

Function “Display stored document”

In this function, the R/3 System sends a request to a viewer application to display a stored document, which is identified by the unique number “ARCH_DOC_ID”.



Displaying PC Files and Displaying Using External Viewers and OLE Automation

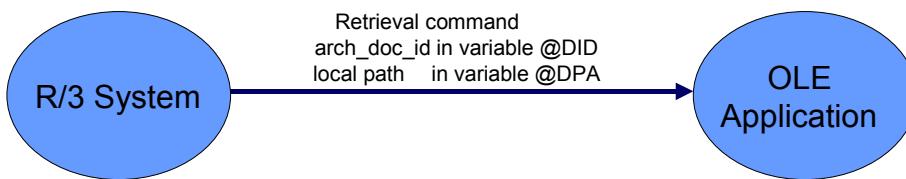
Displaying PC Files and Displaying Using External Viewers and OLE Automation

In the two-step procedure, you must first define that the function *Display stored document* is made up of the two steps *Retrieval for frontend* and *Display local file*. In the following steps, the procedure for the functions *Retrieval for frontend* and *Display local file* is set.

After selecting *Display stored document* and the document class, enter **ARCHIVELNK** as the communication type. This means that the function *Display stored document* is split up into the subfunctions *Retrieval for frontend* and *Display local file*.

Function: Retrieval for Frontend

This function is used to transfer the stored document from the content server to the frontend PC. The R/3 System sends a request to a frontend application to retrieve an stored document, identified by the unique number "ARCH_DOC_ID", as a local file, identified by a transferred file name (including absolute path), on the frontend PC.



Retrieval for frontend can be handled in three different ways:

1. It is still to be checked whether the content server supports this function via OLE Automation. As the OLE functions in the certification process are optional functions, this must be checked on an individual basis with the content server supplier.

The precondition is an content server certified for R/3 Release 3.0.

In this case, after selecting the function *Retrieval for frontend* and the document class, enter **OPEN** as the communication class.

2. An alternative to option 1 is that the content servers provide this function through the implementation of a frontend DLL (Windows). In this case, the function *Retrieval for frontend*

Displaying PC Files and Displaying Using External Viewers and OLE Automation

does not have to be selected for the selected document class (the DLL access is set as a default due to upward compatibility). If you wish to maintain explicitly, you should enter the communication type **ICC** after selecting the function *Retrieval for frontend* and the document class.

3. The third option is to transfer the stored document to the frontend PC via the R/3 application server using sever communication. In this case, after selecting the function *Retrieval for frontend* and the document class, enter **ARCHIVELNK** or **TABLE** as the communication type.

- **ARCHIVELNK**

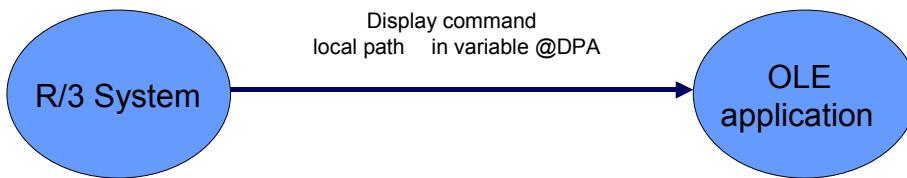
ARCHIVELNK here means that the stored document is transferred from the content server to the application server as a file, imported from the application server and downloaded to the frontend PC.

- **TABLE**

TABLE means that the stored document is transferred to the application server in the form of an internal table and is then transferred to the frontend PC.

Function *Display local file*

This function is used to display the file that is transferred to the frontend PC using the function *Rerieve on frontend*: The R/3 System transfers the request to display a local file, identified by the absolute file name transferred, to a viewer application.



The function *Display local file* can be handled in the following ways:

1. Displaying using an external viewer via OLE Automation

Displaying PC Files and Displaying Using External Viewers and OLE Automation

After selecting *Display local file* and the document class, enter **OPEN** as the communication type.

2. Display via an external viewer using an Execute call (see [Displaying PC Files and Displaying via External Viewers using EXECUTE \[Page 319\] \[Page 319\]](#)).

After selecting *Display local file* and the document class, enter **ARCHIVELINK** as the communication type.



Function *Display local file*

As an example, the Microsoft application Word 7.0 for Office 95 is used to display a document in WinWord format

The OLE class name of the object used is **Word.basic**.

The following calls are executed:

- Call method **FileOpen** with the parameters @DPA to open the local file with the file name @DPA
- Call method **AppShow** to activate the application
- The administration in SAP ArchiveLink is as follows:

Method: **FileOpen** @DPA

Method: **AppShow**

Displaying PC Files and Displaying Using External Viewers via EXECUTE

Displaying PC Files and Displaying Using External Viewers via EXECUTE

If the function *Display local file* was customized via the setting `ARCHIVELNK` as via an Execute call to a viewer, the program is called with the filename as the first parameter.

You can use the function *Program call* to specify the application to be called (and possibly other programs) in different ways:

- Under Windows 32 bit, if no further details are set, the application relevant to the document class (document class = extension) is ascertained and called by the registry. For example, WinWord is the application for document class DOC (Extension.doc), and is called automatically.
- You can use the function *Local program call* to define which application is called for **your frontend PC**.

For this to be possible, a system variable `host name = xyz` must be maintained for the PC.

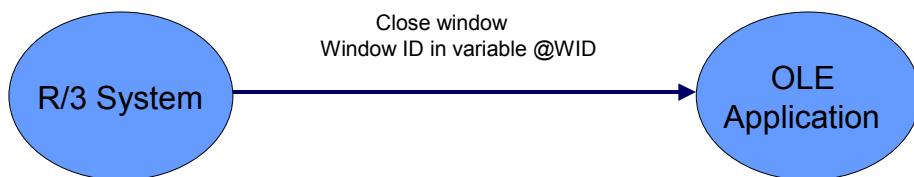
Displaying Outgoing Documents in PDF Format

The display of outgoing documents in PDF format is a case of the two-step display via EXECUTE (see [Displaying PC Files and Displaying using Archive-External Viewers via EXECUTE \[Page 319\] \[Page 319\]](#)). Adobe Acrobat Reader is used as the viewer.

Close Window

Close Window

In this function, the R/3 System transfers a request to a display program to close a window that is identified by the unique *Window_Id*.



This function is used in the storage scenario “Storing for subsequent entry”. **OPEN** is used as the setting.

Applications

Applications

Applications are used for freely administerable communication with frontend applications via OLE Automation 2.0.

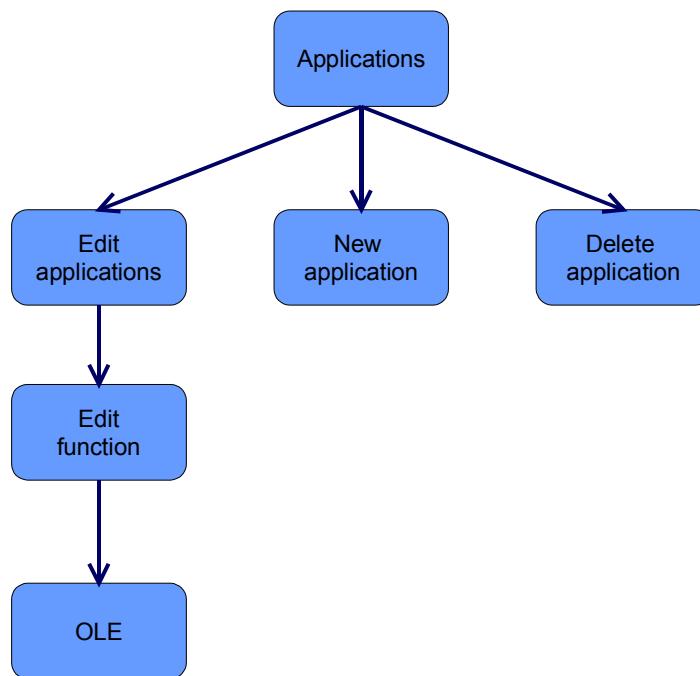
For each function, it is possible to define a sequence of OLE functions, which are processed at runtime. Applications are called via their definition in protocol management.



Application maintenance is only necessary if viewers or entry components are addressed via OLE Automation.

Application maintenance is only required for the functions for which the value "OPEN" was chosen in the protocols.

Process Flow



The syntax that the R/3 System uses to communicate with an application called is predefined by this application.



As of R/3 Release 4.6A, you can only maintain OLE2 applications for frontend functions. AppleScript and OpenDoc are no longer available.

Elements

The central application element is application maintenance. You can maintain applications in the following ways:

Applications

- In the Implementation Guide (IMG)

For this, choose the following in the IMG
“R/3”, “Basis”, “Basis Services”, “SAP ArchiveLink”, “Administration Settings” or “Content server settings”, “[Maintain applications \[Ext.\]](#)”

- In the system

For this, choose
Tools → Business documents,
Basic Settings → Communication → Application maintenance

For information on activity maintenance, refer to the IMG online help.



As of R/3 Release 4.0A, it is possible to determine in application maintenance whether the object generated is to continue to exist or be released again after processing. For further information, refer to the Implementation Guide (IMG) under *Basis - Basis Services - SAP ArchiveLink - Administration Settings (or Content Server Settings) - Maintain applications*.

Syntax: OLE Automation

Syntax: OLE Automation

Separate the **name** of the application and **commands** to communicate with this application. You can split these commands up into method calls, attribute settings and the retrieval of object attributes:

- **Name** of the application
- **Commands** for communication with the application
 - Method calls (see [Calling Object Methods \[Page 329\]](#) [\[Page 329\]](#))
 - Attribute settings (see [Setting Object Attributes \[Page 331\]](#) [\[Page 331\]](#))
 - Retrieval of object attributes (see [Retrieving Object Attributes \[Page 332\]](#) [\[Page 332\]](#))

The application names are described as is the communication syntax in the user documentation for the application to be called. The **application names** are also frequently called **object names**.



Object names:

- Application name for WinWord: **WORD . BASIC**
- Application name for Excel: **EXCEL . APPLICATION**

Transfer of Variables via SAP ArchiveLink

So that you can communicate universally from SAP Archive Link using OLE and Apple Script, is it necessary to transfer **variables/parameters** to OLE objects or to obtain values of OLE objects and to transfer these values to variables in SAP Archive Link. For this purpose, variables have been defined which you can transfer to OLE applications or into which you can transfer values of OLE objects. These parameters are specified in the application administration of the OLE function.



All variables are string variables (no integers).

The following variables are defined

OLE Variables

| Variable | Description |
|----------|--|
| @DPA | File path on frontend, display path |
| @AID | Content server |
| @DTI | Document class |
| @DID | Document ID (arc_doc_id) |
| @SPA | Path on application server, basic path or archive path |
| @WID | Window ID |
| @WIT | Window title |
| @ADA | Storage date |

Syntax: OLE Automation

| | |
|------|---|
| @ATI | Archiving time |
| @STA | Document status (online, offline,...) |
| @COU | Number of retrieved files |
| @DFN | Names of retrieved files |
| @UID | User name |
| @LAN | Logon language |
| @ETX | Error text |
| @DPO | Document path |
| @PAG | Page number |
| @FPA | Frontend path |
| @EID | Error number/return code (0 for okay, string for error) |



OLE parameter @DPA is used most frequently.

Calling Object Methods

Calling Object Methods

There are two syntax forms when you call object methods:

Syntax 1:

```
<ObjectMethod> = <Variable>
```

Syntax 2:

```
<ObjectMethod> <ExportVariable 1>.. <ExportVariable 5>
```

`<ObjectMethod>` is a name defined by the OLE application, whereas `<Variable>`, `<ExportVariable 1>.. <ExportVariable 5>` represent the occurrence of transferred variables. Export variables are used for transmitting values to the OLE object (similar to setting object attributes).

Syntax Variant 1

From a functional point of view, the syntax variant 1 is a sub-case of the syntax variant 2, because it can transfer only one variable. OLE applications do not provide alternative syntax variants, but generally only provide one form.

The syntax variant 1 allows the following functionality:

- Call an object method with an import variable
- Create a new (sub-)object from the current object.



Determine the current file name in Word:

```
filename = @DPA
```

Syntax Variant 2

Syntax variant 2 allows you to call an object method with up to five export variables.



Displaying a local file in Word:

```
fileopen @DPA
```

Sub-objects

You can also call object methods that do not refer to the global object but to a **sub-object** of the application. You should first create this object. You can edit the attributes there after this. The corresponding syntax is then:

Syntax:

```
<Object>.<ObjectMethod> = <Variable>  
or.  
<Object>.<ObjectMethod>  
<ExportVariable 1>.. <ExportVariable 5>
```



Displaying a file in MS Excel 97:

Excel manages an entire tree structure of objects. The root is the object "Excel.Application". Sub-objects are "Sheet", "Workbook", etc. To address an object, you need to create it from the root. Only then can you edit a method or attribute of this object.

| Object name | Excel.Application | Description |
|-------------------|-------------------------------|--|
| Setting attribute | <code>visible = true</code> | Setting the object attribute for the root object |
| Method 1 | <code>WORKBOOKS = mywb</code> | Create the object <code>mywb</code> of type <code>WORKBOOKS</code> |
| Method 2 | <code>mywb.open @DPA</code> | Call the method <code>OPEN</code> for the object <code>mywb</code> |

Variable Exchange

Unnecessary variables (see list of OLE parameters in [Syntax: OLE Automation \[Page 327\]](#) [Page 327]) can be used for the variable exchange between OLE calls.

Setting Object Attributes

Setting Object Attributes

Syntax:

```
<ObjectAttribute> = <Value>
```

<ObjectAttribute> is a name defined by the application, whereas **<Value>** represents the occurrence of the variables or the value of the constants transferred via the R/3 System.



To display a spreadsheet in Excel, you need to mark Excel as visible. To do this, the object "Excel.Application" provides the attribute "Visible". To display Excel, the call is as follows:

```
visible = 1
```

You can also edit object attributes that do not refer to the global object but to a sub-object of the application. You should first create this object. You can edit the attributes there after this. The corresponding syntax is then:

Syntax:

```
<Object>. <ObjectAttribute> = <Value>
```

An example can be found in [Calling Object Methods \[Page 329\]](#).

Retrieval of Object Attributes

Syntax:

```
<Value> = <ObjectAttribute>
```

`<ObjectAttribute>` is a name defined by the application, whereas `<Value>` represents the occurrence of the variables transferred via the R/3 System. The values of these variables are returned to SAP Archive Link.

You can also edit object attributes that do not refer to the global object but to a sub-object of the application. You should first create this object. You can edit the attributes there after this. The corresponding syntax is then:

Syntax:

```
<Value> = <Object>. <ObjectAttribute>
```

An example can be found in [Calling Object Methods \[Page 329\]](#).

Example for Applications

Example for Applications

The example below concerns storing MS Word 97 (for MS Office 97) files.

Configuring the function "Display local file"

| Object name | word.basic | Description |
|-------------|----------------------------|----------------------|
| Method 1 | <code>fileopen @DPA</code> | Open file |
| Method 2 | <code>AppShow</code> | Activate application |

Information for SAP Business Workflow

The following sections are intended for those configuring SAP Business Workflow. All settings relevant to SAP Business Workflow are described.

Default Settings for Storage Scenarios

Default Settings for Storage Scenarios

What is a Default Setting?

You can define default settings to simplify the task of the user at the storing work center whose task it is to assign incoming documents to document types. A default setting consists of a list of document types.

Examples of Default Settings

| Work center | Default Setting | Document type |
|------------------|-----------------|---|
| Mailroom | Mail | Accounting business object Sales business object |
| Sales department | Sales | Customer order Customer contract Complaint |

Maintaining Default Settings



As of R/3 Release 4.5A maintenance of default settings is simplified:

- There is only **one** instead of three maintenance transactions.
- Maintenance is carried out using view clusters.
- You can maintain language-dependent long texts.
- There is a connection to the Change & Transport System

You maintain the default settings for the storage scenarios in the Implementation Guide (IMG) or in SAP ArchiveLink administration. The procedure is explained in the IMG.

In the IMG, choose

“R/3”, “Basis”, “Basis Services”, “SAP ArchiveLink”, “Business Workflow Settings”, “Maintain default settings for storage scenarios”.

In SAP ArchiveLink administration, choose

Miscellaneous → Default settings.

Further Default Settings

General Settings

You can make certain general default settings for storing with SAP ArchiveLink in the Implementation Guide (IMG).

- Activate storing with bar codes

This refers to early and late storing with bar codes.

You activate storing with bar codes as follows:

- a) Via SAP ArchiveLink administration

Choose

*Tools → Business Documents,
Basic Settings → Bar codes → Barcode entry.*

- b) via the IMG, choose [Activate Bar Code Storage \[Ext.\]](#)

- Specify user content server

This refers to a content server specifically for the current user.

You specify the user content server as follows:

- a) Via SAP ArchiveLink administration

Choose

*Tools → Business Documents,
Miscellaneous → User content server.*

- b) via the IMG, choose [Maintaining User Content Servers \[Ext.\]](#)

Incoming Documents

For storing incoming documents with **SAP Business Workflow**, it is necessary to make further presettings. These are described below.

- Maintain global document types

Global document types comprise all existing document types in the system. Document types can be created in the following ways:

- a) Via SAP ArchiveLink administration

Choose

*Tools → Business Documents,
Document types → Global document types.*

- b) Via the SAP ArchiveLink document types Customizing Wizard

The SAP ArchiveLink document types Customizing Wizard leads the user step-by-step through the maintenance of all settings required to create new document types. For further information, refer to the Wizard online help.

Choose

*Tools → Business Documents,
Document types → Customizing Wizard*

Further Default Settings

- c) in the IMG, choose "[Maintaining Document Types \[Ext.\]](#)
or
[Calling the Document Types Customizing Wizard \[Ext.\]](#)"
- Make administrative settings for SAP Business Workflow
Basic settings for SAP Business Workflow comprise, amongst other things, the maintenance of an active plan variant and the maintenance of a workflow administrator.
Choose "[Maintaining Workflow Basic Settings \[Ext.\]](#)" in the Implementation Guide (IMG).
- Maintain Workflow document types

You **must** make this setting if the storage strategies "Storing for subsequent entry", "Storing for subsequent assignment", "Storing and entering", "Storing and assigning" are to be used and standard documents are not to be used.



You can only maintain those Workflow document types that are already defined as global document types.

Workflow document types can be created in the following ways:

- a) Via SAP ArchiveLink administration

Choose
Tools → *Business Documents*,
Document types → *Workflow document types*.

- b) Via the SAP ArchiveLink document types Customizing Wizard

The SAP ArchiveLink document types Customizing Wizard leads the user step-by-step through the maintenance of all settings required to create new document types. For further information, refer to the Wizard online help.

Choose
Tools → *Business Documents*,
Document types → *Customizing Wizard*

- c) via the Implementation Guide (IMG)

Choose "[Maintaining Workflow Document Types \[Ext.\]](#)"

or

[Calling the Document Types Customizing Wizard \[Ext.\]](#)"

For further information, refer to the online help in the IMG.

- Maintain Workflow parameters

You **must** make this setting if the storage strategies "Storing for subsequent entry", "Storing for subsequent assignment", "Storing and entering", "Storing and assigning" are to be used and standard documents are not to be used.



This maintenance is a special case, which arises primarily in connection with the standard tasks TS30001128, TS30001117 and TS00007869.

Further Default Settings

When maintaining Workflow document types, the parameters for a method are given values depending on the workflow document type (see concept for [Object types and Methods \[Page 46\]](#)). Parameters, which are relevant in the execution of the workflow task, are thus defined.

Workflow parameters can be created in the following ways:

- a) Via SAP ArchiveLink administration

Choose
Tools → *Business Documents*,
Document types → *Workflow parameters*.

- b) Via the SAP ArchiveLink document types Customizing Wizard

The SAP ArchiveLink document types Customizing Wizard leads the user step-by-step through the maintenance of all settings required to create new document types. For further information, refer to the Wizard online help.

Choose
Tools → *Business Documents*,
Document types → *Customizing Wizard*

- c) via the Implementation Guide (IMG)

Choose "[Maintaining Workflow Parameters \[Ext.\]](#)"

or

["Calling the Document Types Customizing Wizard \[Ext.\]"](#)

For further information, refer to the online help in the IMG.

- Call SAP ArchiveLink Workflow Wizard

The SAP ArchiveLink Workflow Wizard guides users step-by-step through the generation of object-specific workflow templates. The generated Workflow templates can be used for the storage strategies "Storing for subsequent entry", "Storing for subsequent assignment", "Storing and entering" and "Storing and assigning".

The SAP ArchiveLink Workflow Wizard can be found in the following places in the system.

- a) In SAP ArchiveLink administration

Choose
Tools → *Business Documents*,
Basic Settings → *Customizing* → *Workflow Wizard*.

- b) In the Implementation Guide (IMG)

Choose ["Calling the Workflow Wizard \[Ext.\]"](#) in the IMG.

For further information, refer to the Wizard online help.

Information for Developers

Information for Developers

This section is intended for developers who want to integrate SAP ArchiveLink functions into R/3 applications.

Storing Incoming Documents

Storing with Bar Codes

Storing with Bar Codes

Use

When storing with bar codes, the incoming document is assigned a bar code label either before it is entered or when it is entered. The bar code must be transferred when the business object is posted.



As of R/3 Release 4.0A, storing with bar codes can be performed using generic [Object Services \[Page 89\]](#) (see also [Connecting Applications to Object Services \[Ext.\]](#)). For more information on storing with bar codes, see [Assigning Bar Codes Generically \[Page 110\]](#).

For information on the general process of early and late storing with bar codes, see [Process Flow: Early Storing with Bar Codes \[Page 78\]](#) and [Process Flow: Late Storing with Bar Codes \[Page 79\]](#).

Integration

As well as connecting to generic object services (see above) you can also integrate storing with bar codes explicitly into your R/3 application by integrating the function module **ARCHIV_BARCODE_GLOBAL**.

The function module **ARCHIV_BARCODECHECK_(Name des Barcodetyps)** is also available: You can define new bar code types in SAP ArchiveLink Administration and in the IMG and specify whether a bar code check is to be performed for these. This function module performs the check. The bar code ID is transferred and a return code expected, where a return code = 0 indicates that the bar code is correct. A non-zero value indicates an error.

The standard R/3 System provides a bar code check for the following bar code types (see [Bar Codes \[Page 82\]](#)):

- 2/5_I (2/5 interleaved)
- EAN_8 (EAN 8)
- EAN_13 (EAN 13)
- UPC_A (UPC A)
- UPC_E (UPC E)



Only the check digits are checked and not the semantics of the bar code.

Integration of ARCHIV_BARCODE_GLOBAL

ARCHIV_BARCODE_GLOBAL

```
IMPORTING    AR_OBJECT
             NO_DOC_CHANGE  Default   SPACE
             OBJECT_ID
```

Storing with Bar Codes

| SAP_OBJECT | | |
|------------|---------|-------|
| UPDATETASK | Default | SPACE |
| NO_CHECK | Default | SPACE |
| NO_POPUP | Default | SPACE |
| BARCODE | Default | SPACE |

The function module does the following:

- Calls a dialog box in which the user enters the bar code. For this, the parameter **NO_POPUP** must have the value **SPACE**. If the parameter has the value **x**, no dialog box is displayed. This is preferable for processing batch input sessions, for example.
- If the **NO_CHECK** parameter has the value **x**, there is no check as to whether bar code entry is active for the user. However, an entry containing the object type transferred must be made in administration in each case.
- Enters the bar code and the corresponding application-related object key in the link table.

The parameter **NO_DOC_CHANGE** determines whether it is possible for the user to change the document type in the dialog box.

The parameter **UPDATETASK** determines whether the entry in the link table is made in the update task or in dialog.

Storing Outgoing Documents

Storing Outgoing Documents

Use

Outgoing documents are stored synchronously. Outgoing documents created in SAPscript are stored via the **transactional RFC**.

For more information, see administration information on [Storing Outgoing Documents \[Page 121\]](#).



As of R/3 Release 4.0B document type PDF exclusively is used for outbound documents. No more outgoing documents are created in OTF Format.

Advantages of synchronous storing

- Outgoing documents are stored synchronously, that is, as soon as the storage request is made.
- Using the transactional RFC does not initialize an update termination, if an error occurs during storing.
- Unsuccessful storage requests can be monitored via the monitor. Storage requests containing errors can be repeated individually. To do this, select *Environment* → *Transactional RFC* in the monitor.

Prerequisites

All transactional RFC calls are stored with the ABAP language element **COMMIT WORK** in the database. These RFC calls thus constitute a Logical Unit of Work (LUW).

It is assumed that usually in the relevant R/3 application after calling the SAPscript function module a **COMMIT WORK** ensues. If this is not the case, it is possible to transfer the value “Commit” when calling the SAPscript function module (for example, **OPEN_FORM**) in the import parameter **ARCHIVE_INDEX-RESERVE(6)**. If this value is set, a **COMMIT WORK** is triggered by an SAP ArchiveLink function module.



COMMIT WORK is not allowed during posting. If the value “Commit” is set in the import parameter **ARCHIVE_INDEX-RESERVE(6)** triggering a **COMMIT WORK** in the SAP ArchiveLink function module, there must be **no** posting active. Otherwise there is a runtime error **COMMIT_IN_POSTING**.



For more information, see [Storing Outgoing Documents \[Page 215\]](#) and [Form Overlay \[Page 217\]](#).

Storing Outgoing Documents

The content server is considered an additional logical printer. SAPscript has been enhanced so that storing appears to be carried out in the same way as a print operation from the application.

The SAPscript function modules OPEN_FORM and PRINT_TEXT have been enhanced by the import parameters

- **ARCHIVE_INDEX** and
- **ARCHIVE_PARAMS**

The SAPscript function module START_FORM has been enhanced by the import parameter **ARCHIVE_INDEX**.

You can find further information in the online documentation.

ARCHIVE_INDEX

Structure TOA_DARA / DARA index line

ARCHIVE_INDEX contains the DARA index line. The DARA index line must always be completed.

A DARA line identifies a sub-object of a document. The DARA parameters must be transferred by the application.

DARA lines are structured as follows:

| Line Structure | Offset | Length (bytes) | Description |
|----------------|---------|----------------|---|
| INDEX_NAME | 0-3 | 4 | Name of record type (DARA) |
| CLIENT | 4-6 | 3 | Client |
| EXPIRY_DATE | 7-14 | 8 | Date when the table entry referring to the stored document is removed |
| SAP_OBJECT | 15-24 | 10 | Business object type |
| AR_OBJECT | 25-34 | 10 | Document type |
| OBJECT_ID | 35-84 | 50 | Key for business object |
| FORM_DOC_ID | 85-124 | 40 | Document ID for the form which is assigned to outgoing documents. |
| FORM_ARCHIV_ID | 125-126 | 2 | Storage ID for form |
| RESERVE | 127-153 | 27 | Reserve |
| NOTE | 154-409 | 256 | Note which can be stored in the content server with a stored document |

To ensure that individual sub-documents are assigned to the business object, the structure components SAP_OBJECT, AR_OBJECT, and OBJECT_ID must have values.

Storing Outgoing Documents



For SAP_OBJECT and AR_OBJECT, a link must be maintained in **Customizing** so that no error scenario arises.

The structure TOA_DARA is created for indexing.

ARCHIVE_PARAMS

If print output does **not** occur ONLINE and under the control of the user, the parameters **ARCHIVE_PARAMS** must have values. The necessary information can be determined via the function module GET_PRINT_PARAMETERS.

To determine the export parameters **OUT_ARCHIVE_PARAMETERS**, the import parameters **AR_OBJECT** (document type) and **SAP_OBJECT** (business object type) must have values.

OPTIONS (ITCPO structure)

The **TDARMOD** field should be transferred. The parameter contains the storage mode (see online documentation) :

- 1 Print
- 2 Store
- 3 Store and print

The parameter **OPTIONS_TDARMOD** must contain 2 (store) or 3 (store and print), for the outgoing document to be stored.



The program OALINES is available as an example report.

Form Overlay



The form overlay for storing outgoing documents in PDF format is available as of R/3 Release 4.5A.

The form S_FORM_TEST_02 and the print program OATESTFORM are available as examples for **form overlay**.

For more information, see [Form Overlay \[Page 217\]](#).

Form Overlay



Examples of form overlay are available as of R/3 Release 4.5A for those R/3 applications in which form overlay is to be used when storing outgoing documents in PDF format.

Use

Integration of form overlay enables you to define forms for individual documents or groups of documents when storing outgoing documents, which are stored together with the documents. In this way graphics such as logos and also variable or fixed texts can be added to the actual content of the outgoing document.

Features

The following examples are available for form overlay:

- Form S_FORM_TEST_02
- Print program OATESTFORM

Activities

Form S_FORM_TEST_02

1. To go to the **form painter**, choose *Tools → SAPscript → Form*.

For information on the form painter, see the documentation [BC - Maintaining Styles and Forms \[Ext.\]](#).

2. Enter the name of the example form **s_FORM_TEST_02**.

Page layout for the first page:

| Content | Description |
|-------------|---|
| Main window | Introduction, content, concluding remark |
| Adress | Title, name, street, zip code, town |
| Footer | Variable or fixed text |
| Header | Variable or fixed text |
| Sender | Grayed-out text box with variable and/or fixed text |

Page layout for further pages:

| Content | Description |
|-------------|--|
| Main window | Introduction, content, concluding remark |

Form Overlay

| | |
|-------------|------------------------|
| Footer | Variable or fixed text |
| Header | Variable or fixed text |
| Page number | Number |

You can change the elements using the form painters by adding fixed and variable texts, graphics, tables and so on.

Print program OATESTFORM

1. Execute the program **OATESTFORM** in the ABAP editor.

The screen "SAP ArchiveLink: Storing Outgoing Documents with Form" is displayed.

The following fields already contain entries, as far as is possible:

- *Customer number from... to...*
 - *Object type*
 - *Document type*
 - *Storage date from*
 - *Storage date to*
2. Change the content of the fields, if necessary.
 3. Choose *Program → Execute*.

You go to the print screen.

The fields on the print screen contain entries.

4. Change the content of the fields, if necessary.
5. Choose *Output → Print*.

In the dialog box containing the storage parameters, the object type and document type are specified.

6. Change the content of the fields if necessary and choose *Store* to confirm.

Entries are made in the DARA lines and storing is started, for the outgoing document.

You receive a message informing you that the storage or print request has been generated.

Storing Print Lists

Use

You can store print lists created in the R/3 System (ABAP lists and screen lists) in an external content server.

Prerequisites

To store asynchronously, the document to be stored must be available in the R/3 System in a file that can be accessed via a unique path.

Features

Print lists are stored **asynchronously**. Asynchronous storing means that the content server does not return a document ID directly after the storage request, but only after the document has been stored in the content server.

To store print lists asynchronously, the following steps are necessary:

- **Indexing**

If the document is indexed, the content server can later access specific subobjects of this document. Although print lists can be stored without indexing, it is recommended that they be indexed.

[Indexing Print Lists \[Page 221\]](#)

[Inserting Index Information and Hypertext Links \[Page 227\]](#)

- **Storing**

[Storing Print Lists Asynchronously \[Page 220\]](#)

Storing Print Lists Asynchronously

Storing Print Lists Asynchronously

Asynchronous storing of print lists is such that the content server is regarded as an additional logical printer. ABAP has been enhanced so that storing appears to be carried out in the same way as a print operation from the point of view of the application.

A request structure **ARC_PARAMS** (internal table **TOA_ARCAS**) was defined.

The commands **SUBMIT TO SAP-SPOOL** and **NEW-PAGE PRINT ON** were enhanced to include **ARCHIVE PARAMETERS**. The additional field **ARMOD** (indicating whether storing is performed) was added to the parameter **PRI_PARAMETERS**. The command is:

SUBMIT TO SAP-SPOOL ARCHIVE

where the variable **Arparams** contains the **ARC_PARAMS** sought.

If print output does **not** occur ONLINE and under the control of the user, the parameters **ARCHIVE PARAMETERS** must have values. The necessary information can be determined via the function module **GET_PRINT_PARAMETERS**.

If you want to use the command **NEW-PAGE PRINT ON** or **SUBMIT TO SAP-SPOOL**, transfer one of the following values in the supplement **ARCHIVE MODE**:

- 2 Store
- 3 Store and print

To determine the export parameters **OUT_ARCHIVE_PARAMETERS**, the import parameters **AR_OBJECT** (document type) and **SAP_OBJECT** (business object type) must have values.



For **SAP_OBJECT** and **AR_OBJECT**, a link must be maintained in **Customizing** so that no error scenario arises.



The program **OACONTA2** is available as an example report.

The **ARCHIVE_PARAMS** are obtained by calling the function module **GET_PRINT_PARAMETERS**. For further information, see the documentation on commands **SUBMIT TO SAP-SPOOL** and **NEW-PAGE PRINT ON** and function module **GET_PRINT_PARAMETERS**.

Indexing Print Lists

Use

The index information for the content server is separated from the data information by the R/3 System and transferred to two files (data file and description file):

Data files contain the actual information (print lists).

Description files contain the index information.

In the case of print lists, separation into data files and description files occurs in the spool system.

This means that when print lists are stored, the R/3 application itself must write the index information to the print list. The index information is then extracted in the spool and stored in the description file.

Features

The following function modules are available for indexing print lists:

| Function module | Description |
|---|--|
| ARCHIVELINK_DEFINE_KEY [Page 222] | Defines index key structure |
| ARCHIVELINK_CREATE_INDEX [Page 224] | Enters an index for a stored print list |
| ARCHIVELINK_CREATE_DARC [Page 225] | Generates a hyperlink for stored print lists |

ARCHIVELINK_DEFINE_KEY

ARCHIVELINK_DEFINE_KEY

The function module ARCHIVELINK_DEFINE_KEY defines the index key structure.

Definition

ARCHIVELINK_DEFINE_KEY

TABLES KEY_DEFINITION

KEY_DEFINITION defines the structure of the DAIN line.

Function

The structure of the DAIN line is described in the table **KEY_DEFINITION**.

The structure TOA_KEY has the following structure:

- **KEY_NAME**
Key name that is used later for the interactive search.
- **POSITION**
Position of the key component within the DAIN line.
- **LENGTH**
Length of the key component within the DAIN line.



There is an example application in the program OACONTA4.

Index Line DKEY

DKEY lines define keys or partial keys in the index lines (DAIN lines) and therefore the index structure.

The sequence of DKEY lines in the indexed data file or description file defines the order of the keys in the DAIN line.

DKEY lines are structured as follows:

| Line Structure | Offset | Length (bytes) | Description |
|-----------------------|---------------|-----------------------|--|
| INDEX_NAME | 0-3 | 4 | Name of record type (DKEY) |
| KEY_NAME | 4-43 | 40 | Key name |
| POSITION | 44-46 | 3 | Offset position The offset begins with 0 at the first position after DAIN |
| LENGTH | 47-49 | 3 | Length of key name in bytes |

The keys are defined by KEY_NAME, POSITION, and LENGTH.

ARCHIVELINK_CREATE_INDEX

ARCHIVELINK_CREATE_INDEX

The function module ARCHIVELINK_CREATE_INDEX enters an index for a stored print list.

Definition

ARCHIVELINK_CREATE_INDEX

IMPORTING WRITE_LINE

In **WRITE_LINE** the index line is transferred whose structure is defined using the function module [ARCHIVELINK_DEFINE_KEY \[Page 222\]](#).

Index Line DAIN

A DAIN line contains the index for a sub-object of a document.

The structure of a DAIN line is described by DKEY lines. A DAIN line is inserted before each sub-object which is to be indexed. It is used by the content server for direct positioning on the sub-object.

DAIN lines are structured as follows:

| Line Structure | Offset | Length (bytes) | Description |
|-----------------------|---------------|-----------------------|----------------------------|
| INDEX_NAME | 0-3 | 4 | Name of record type (DAIN) |
| LINE | 4-135 | 132 | Contents of DAIN line |

ARCHIVELINK_CREATE_DARC

The function module ARCHIVELINK_CREATE_DARC generates hyper text links for stored print lists.

Definition

```
ARCHIVELINK_CREATE_DARC
  IMPORTING    OBJTYPE
  OBJKEY
  METHOD        DEFAULT    'DISPLAY'
  TABLES        PARAMETERS
```

OBJTYPE transfers the business object type to which the hyperlink relates.

OBJKEY transfers key of the business object type to which the hyperlink relates.

METHOD supports the method display.

PARAMETERS transfers parameters that can be transferred to the object method. Since Display is only the method currently supported that can contain no mandatory parameters, this parameter will be used in the future.

Index Line DARC

A DARC line defines the position in a sub-object which is marked as hypertext.

When the stored print list is displayed, the user can double-click on this position to display the reference stored document. If several stored documents are referenced, double-clicking on the hypertext generates a list from which the user can select the required stored document.

A DARC statement or a series of DARC statements refers to a previous hypertext link in an output line. If an output line contains several hypertext links, to which various DARC lines then refer, the counter HYPNUMBER is used to ensure that the hypertext links are assigned to the DARC lines. Numbering is from left to right within the output line, running from 0 to 9. An output line therefore contains a maximum of ten hypertext links.

DARC lines are structured as follows:

| Line Structure | Offset | Length (bytes) | Description |
|----------------|---------|----------------|---|
| INDEX_NAME | 0-3 | 4 | Name of record type (DARC) |
| HYPNUMBER | 4-5 | 1 | Number of current hypertext |
| ARCHIV_ID | 6-15 | 10 | Storage ID |
| ARC_DOC_ID | 16-55 | 40 | Document ID |
| LANGTEXT | 56-95 | 40 | Document type |
| AR_DATE | 96-103 | 8 | Storage date |
| DOKINFO | 104-153 | 50 | Document information in index line DARC |

ARCHIVELINK_CREATE_DARC

Index Information and Hypertext Links

Function module **ARCHIVELINK_CREATE_HYPERLINK** is used to create a DARC index line in a print list.

Definition

```
ARCHIVELINK_CREATE_HYPERLINK
  IMPORTING    WRITE_LINE
              OBJTYPE
              OBJKEY
              METHOD      DEFAULT  'DISPLAY'
              HYPENUMBER  DEFAULT  '1'
  TABLES      PARAMETERS
```

WRITE_LINE transfers the text component that is to be marked as a hyperlink.

OBJTYPE transfers the business object type to which the hyperlink relates.

OBJKEY transfers key of the business object type to which the hyperlink relates.

METHOD supports the method display.

HYPENUMBER transfers the hyperlink number in the current line.

PARAMETERS transfers parameters that can be transferred to the object method. Since Display is only the method currently supported that can contain no mandatory parameters, this parameter will be used in the future.

Function

The function module ARCHIVELINK_CREATE_HYPERLINK enables object references within stored print lists to be generated easily. These object references enable business objects to be displayed from print lists. The text component that is the hyperlink is in color. Double-clicking calls the method Display for the relevant business object.



There is an example application in the program OACONTA4.

Information for Content Server Suppliers

Information for Content Server Suppliers

SAP Content Server HTTP 4.5 Interface

This document describes the new **SAP Content Server interface HTTP 4.5 Interface**.



Points to note when transferring from SAP ArchiveLink to the new Content Server interface are explained in the relevant parts of the document.

The aim of the new interface is that only general industry standards such as HTTP and BAPIs should be used in communication with external storage systems (content servers).



The SAP Content Server HTTP 4.5 interface can be certified.

Introduction

Introduction

Definition of Terms

For the purposes of the following description, a **document** comprises of **administrative data** and **content**:

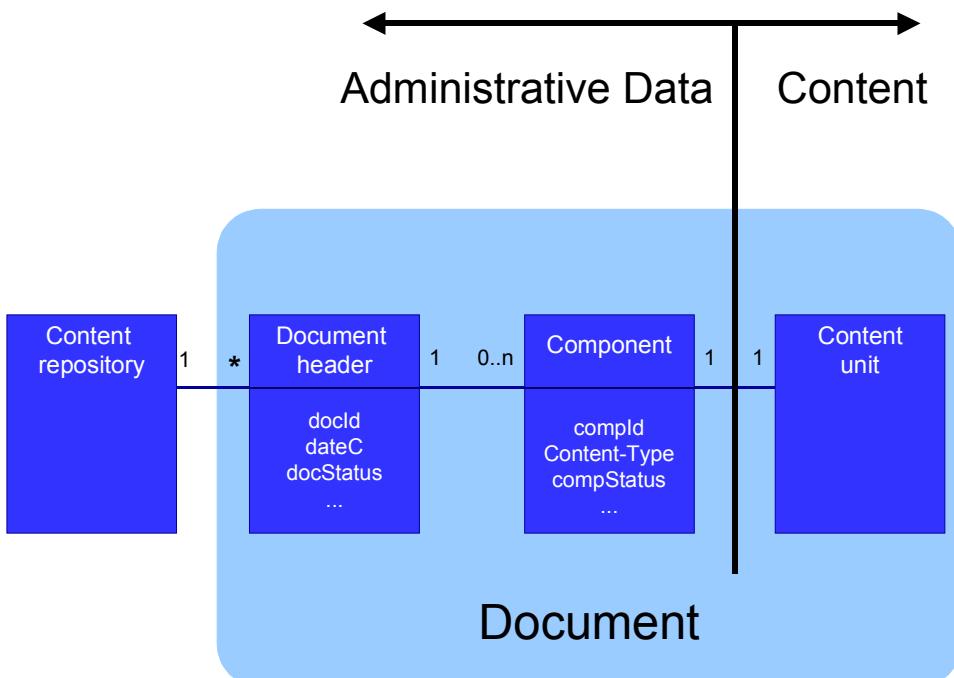
- **Administrative data** identifies and describes a document.
- The **content** of a document consists of closed datasets. The administrative data identifies and describes the content. One closed dataset is a **content unit**.

In SAP terminology, a **content server** is any server that manages content. A content server may be a database, a fileserver, an SAP R/3 System or an external archive.

The administrative data terms **content repository**, **document header** and **component** are of particular importance when identifying documents.

- A **content repository** represents the logical storage space for documents in a content server on an administrative level. Several content repositories can exist on one content server. A content repository is identified by the parameter `contRep`.
- The **document header** is an administrative quantity summarizing several components. It is identified by the parameter `docId`. A document header is assigned to one particular content repository.
- A **component** represents one particular content unit on an administrative level. It is assigned to one particular document header and is identified by the parameter `compId`.

The relationship between content repository, document header, component and content is shown in the diagram below:



Definition of Terms



It can be that a document with one component is generated and this component is then deleted. This leaves an "empty" document, that is, a document with no components. To avoid possible contradictions, we shall assume that such documents can occur. The special case of an empty component is also possible. This may occur, for example, where a file with a size of 0 bytes is stored.

The combination `contRep/docId` is the **one-to-one address** for a document header.

The combination `contRep/docId/compId` is the **one-to-one address** for a component.

Under certain circumstances documents are subject to **protection**. This means that functions executed on the document must be legitimized. For each document header, you can define whether or not legitimization is necessary for particular functions. This information is not defined in the document header for each Content Server interface function, but instead via **access modes**. Access modes are defined as disjunct groups of Content Server interface functions.

HyperText-Transfer-Protocol (HTTP) is a description of a communication process typically used to access objects on the **World Wide Web (WWW)**.

This protocol is currently being developed further by the W3C (WWW Consortium, <http://www.w3c.org>). The protocol HTTP/1.1 or HTTP/1.0 can be used for the communication process. RFC (**R**equest **F**or **C**omment) 2068 specifies protocol HTTP/1.1: Protocol HTTP/1.1 contains more precise regulations than protocol HTTP/1.0 (RFC 1945), which ensures the reliable implementation of HTTP characteristics.

The new interface is designed such that communication is always started by the client R/3 System. The content server addressed by the R/3 System is always only a server and never itself a client that instigates communication with the R/3 System.

HyperText-Markup-Language (HTML) is a standard format and description language for WWW pages.

Uniform Resource Locators (URLs), see RFC 1738) are a standardized mechanism used to address uniquely defined objects on the WWW. As well as the actual address, URLs can contain functions and parameters that can be interpreted by the object addressed.

UTC (Universal Time Coordinated) is used for all expressions of time in this specification.



The following rules apply to the spelling of functions, parameters and key words in this description:

All terms defined in the Hypertext Transfer Protocol HTTP/1.1 (RFC 2068) are used correspondingly (for example, `Content-Type`). Protocol HTTP/1.0 (RFC 1945) is also supported.

Terms specific to this interface description are not capitalized if they consist of one syntactical term (for example, `info`). A combination of lower and upper case is used if they consist of more than one syntactical term (for example, `contRep`).

Implementation

The HTTP protocol is used for communication with content servers. Servers and documents are addressed using URLs and data is transferred in the Request-Body or in the Response-Body.

The URL specifies the function to be executed on a document: `get` (transferring from the server to client), `info` (retrieving information on the document) or `create` (creating a new document). The necessary parameters for these functions are also part of the URL.

This specification describes the URL syntax and the semantics for the various functions.

Security

Security

Security and the related guarantee of secure data transfer are central aspects of the Content Server interface. The following principles apply:

- It is assumed that all authorization checks in the R/3 System are performed.
- To ensure that these authorization checks cannot be circumvented for content server access, a public/private key procedure is used (see also [Public Key Technology \[Ext.\]](#)).
- The public and private keys are R/3-specific, not user-specific.

The security concept of the Content Server interface is based on the fact that the R/3 System public key is stored in the Content Server. This is done using the command `putCert`. The content server uses the certificate to check URLs and signatures (see also [putCert \[Page 284\]](#)).



For more information, see the documentation [Secure Store & Forward / Digital Signatures \[Ext.\]](#).

secKey

The **secKey** ensures that a URL cannot be changed after it has been generated by the R/3 System. This ensures that access to the document is protected and that access protection is managed in the R/3 System. The **secKey** does not protect the content of the document. The following parameters are always signed in the **secKey**:

- **contRep** Content repository
- **accessMode** Access mode
- **authId** Client ID
- **expiration** Expiry time (UTC)

authId must be a unique identification of the client (for example, the R/3 System). The UTC expiry time is written in the format: yyyyymmddhhmmss. If the expiry time has been exceeded, the content server must report HTTP status code 401 to the client.

If a **secKey** is transferred with the URL, the parameters **accessMode**, **authId** and **expiration** must also be transferred. These parameters need not be transferred if the **secKey** is not transferred.

Additional parameters must be signed. These depend on the particular function and are specified in the function description. The name of the function itself is not signed. The parameters to be signed can appear in the URL in any order. To check the signature, it must be ensured that the order in which the parameters are transferred to the signalmodule is the same as the order in the URL.

The **secKey** for the chosen procedure is about 500 bytes long.

The parameters to be signed for a particular function are specified in the function definition. They are specified in the last column of the parameter table. If these are *mandatory* parameters, they must always appear in the URL and are therefore always signed. Optional parameters can clearly only be signed if they are used. *s-mandatory* parameters must appear in the URL if a signature is used. They are always signed. If no signature is used, these parameters are not evaluated.

The URL parameters to be signed are the *Message*. The message is used to determine a hash value. The parameters must be kept in the same order for determination of the hash value. The hash or message digest is a one-way function and so cannot be reversed. Using the sender private key, the hash value is digitally signed according to DSS (Digital Signature Standard) via the SAP SSF (Secure Store & Forward) module according to PKCS#7. The digital signature is transferred in the URL in the parameter **secKey** (as described above).

Once the digital signature has been created, the URL parameters are safe from distortion. They are not encoded. All recipients can check the URL parameters using the sender public key. Any changes would therefore be detected. This ensures that an action on the content server can only be started, if the URL transferred has not been distorted.

Using the sender public key, the content server generates the message digest again from the transferred URL. It then forms a hash from the message (the order of the parameters in the URL is important here) and compares the two hashes (the message hash and the hash generated by the sender). If they match, the URL has not been distorted during transfer between the R/3 System and the content server.

secKey

The library for checking signatures is available from SAP AG if required. Since the standard format PKCS#7 was used for the signature, other products can also be used for decoding.

Brief technical information:

- Format of digital signature: PKCS#7 "Signed-Data"
- Public key procedure: DSS
- Key length: 512 – 1024 bits
- Public exponent: $2^{16} + 1$
- Public key format: X.509 v3 certificate
- MD (Message Digest) algorithm: MD5 or RIPEMD-160

Protection / Right of Access

The degree of protection is specified when a document is stored. When a document is accessed, the function a user may execute on this document is defined. Similar functions are grouped together. The groups are called access modes. They are listed in the following table:

| Access mode | Abbreviation |
|-------------|--------------|
| Read | r |
| Create | c |
| Change | u |
| Delete | d |



Protection applies to all components of the document. If the access mode is “change”, corresponding components of a document can be deleted.

The access mode must be specified in the HTTP request as a parameter (`accessMode`). A combination of access modes can be specified, for example, `ud`. A `secKey` confirms the right of access. The corresponding access mode is specified in the descriptions of individual functions. When a document is accessed, the content server checks whether the `secKey` should be checked, that is, whether the document is protected regarding a particular function. It is often sensible that all users may read documents but only certain users may change documents. In this case, read protection would be deactivated (no `secKey` is required), for writing or deleting, however, a `secKey` must be transferred. The fact that the `secKey` can only be generated by the R/3 System ensures that an access protection check based on the R/3 authorization concept was performed.

Protection is defined when a document is created. This is done using the parameter `docProt`.



| Protection | Description |
|-------------------------|--|
| <code>docProt=</code> | No access constraints |
| <code>docProt=du</code> | Delete and update operations only permitted with signed URLs. For delete operations, the <code>accessMode</code> must contain at least one <code>d</code> and for update operations at least one <code>u</code> . Read operations can be performed without a signature. |



It is permitted to transfer several access modes: for example, `accessMode rd` for a read operation. This makes specific scenarios possible: If a get-URL with `accessMode=rd` and the corresponding signature is transferred to a client program, the client has the option not only of reading the document, but also of deleting the

Protection / Right of Access

entire document. To use the URL for deleting, it suffices to replace the command `get` with `delete` and to not transfer the `compId` (if it exists). Since the same parameters are signed for `get` and `delete`, the signature remains valid here. If the `accessMode` contains a `d`, it is possible in this example to delete the document.

On the basis of the access mode of an operation and the concrete protection of a document, the content server decides whether the `secKey` is to be checked. If the content server decides that it is not to be checked, all s-mandatory parameters are obsolete and it is not necessary to check these parameters either.



If they are unnecessarily transferred anyway, you can check them, but this does not increase security and is therefore superfluous, especially since you can increase performance in operations where protection is not required.

The parameter `docProt` is optional, but is generally transferred even if the URL is not signed. If neither the content server nor the R/3 System use the signature, this does not make a difference to the protection definition when creating documents.

If the parameter `docProt` is not transferred, the server default setting is to be used. The content server has complete freedom here.

If the R/3 System uses this opportunity, it must employ maximum protection and use the corresponding signed URLs for all subsequent accesses to the relevant documents.



The signature in the R/3 System may only be deactivated if no check is to take place in the content server.

In productive operation, you should generally use the signatures, however.

For all access modes, it must be possible to set as default on the Content Server whether a `secKey` must be specified or not. This default can be overwritten for the functions `create` and `mCreate`. If no protection is specified, the default is used.

Old data and documents that were stored in the content server not using the HTTP interface are subject to the highest level of protection, that is, all accesses must be signed.

Syntax

General

General

The URL syntax is:

http://servername:port/script?command¶meters

The **servername** is the name of the server machine which is accessed and **port** (optional) is a TCP/IP port that can be used to address the server. **script** is the name of the program used to access the content server. This may be a DLL, a CGI script or an **Active Server Page** (ASP). The object is created by the content server provider. A command must exist, followed by one or more **parameters**.



There must not be any blank spaces in the URL.



http://pswdf009:1080/ContentServer/ContentServer.dll?get&pVersion=0045&contRep=K1&docId=361A524A3ECB5459E0000800099245EC

Coding in the URL

The structure of URLs is described in RFC 1738. It also specifies which character set may be used for a URL and how characters not in this set should be encoded.

Only characters from a ASCII character set may be used in a URL (0x00 - 0x7F). Some of these characters must be encoded. (The characters 0x00 - 0x1F and 0x7F.) If they are to appear in the URL, a '%' (percentage sign) followed by the hexadecimal representation of the character should be used.



A line feed (0x0A) is represented as %0A (for example) in a URL.

Unsafe Characters

A set of *unsafe* characters must be coded in the same way: space, <, >, " , #, %, {, }, |, \, ^, ~, [,]. These characters are unsafe either because they execute special functions in the URL or because they could be interpreted as special characters during transfer.

Reserved characters

There are also *reserved* characters: ;, /, ?, :, @, =, &.

Reserved characters must also be encoded.

Transferring Binary Data

A further problem occurs if binary data is to be transferred in a URL. This is the case when using this interface since the `secKey` consists of binary data. Coding must first be carried out in the ASCII character set. Base64 coding must be used (RFC 1521).

Example

1. Compiling the URL

```
http://pswdf009:1080/ContentServer/ContentServer.dll?get&pVersion=0045&contRep=K1&docId=361A524A3ECB5459E0000800099245EC&accessMod=e=r&authId=pawdf054_BCE_26&expiration=19981104091537
```

2. Generating the secKey

The `secKey` is made up of the encoded parameters. The parameters to be signed are specified in the function definition.

In the present example (`get` function) they are:

ContRep = K1
DocId = 361A524A3ECB5459E0000800099245EC
AccessMode = r
AuthId = pawdf054_BCE_26
Expiration = 19981104091537

In the next step the parameter values are summarized according to the sequence in the URL without separators to form a message:

Coding in the URL

K1361A524A3ECB5459E0000800099245ECrpaadf054_BCE_2619981104091537

The message is used to form the hash from which the **SecKey** is calculated. For reasons of clarity, arbitrary values will be taken for the **secKey** in this example.

secKey value: 0x83, 0x70, 0x21, 0x42.

3. Encoding the **secKey** in the ASCII character set

Base64 must always be used to encode the **secKey**.

0x83, 0x70, 0x21, 0x42 -> g3AhQg==

4. Encoding the URL in accordance with the URL character set limitations

Characters may need to be encoded. That is the case in this example:

g3AhQg== -> g3AhQg%3D%3D

The following URL is generated:

```
http://pswdf009:1080/ContentServer/ContentServer.dll?get&pVersion  
=0045&contRep=K1&docId=361A524A3ECB5459E0000800099245EC&accessMod  
e=r&authId=pawdf054_BCE_26&expiration=19981104091537&secKey=g3AhQ  
g%3D%3D
```

Coding in the Response Body

Coding in the Response Body

Many of the functions described return information in the response body. If the information is returned in ASCII format, the lines always consist of key/value pairs separated by a semicolon:

```
key1="value1";key2="value2";...keyn="value2";CRLF
```

Only ASCII characters that can be printed may be used. If a value contains an inverted comma, this is in addition to the inverted commas already inserted around the value.

Functions

Functions

The functions available and their parameters are described below. For each function, the effect, the possible parameters and an example are given.

- **Effect**

Under Effect, the executed function is described. The meaning of the individual parameters is given.

- **Default**

Under Default, the effect of transferring only the mandatory parameters of a function with many parameters is described.

- **Access mode**

Under Access mode, the access mode for the function is specified.

- **Client → Server**

Under Client → Server, the parameters that are transferred from the client to the server are listed and specified as optional or mandatory. It is specified whether the parameters are optional or mandatory. s-mandatory means that the relevant parameter must only be specified if a **secKey** is transferred. The HTTP-Request type is defined and the way the parameters are to be coded in the URL and/or in the body is described.

- **Example**

Under Example, a function is performed using example parameters. Line breaks in the examples are purely to aid legibility. The actual URLs do not contain any line breaks.

- **Server → Client**

Under Server → Client, the structure of the HTML-Response is defined. This response is generated by the server and sent to the client.

The HTTP status codes specific to the content server are also listed in this section. If no security key is entered, this may cause (for example) error 401 (unauthorized); a wrongly addressed document can cause error 404 (not found). If an error occurs, the content server must also deliver an ASCII string describing the error. The error must be entered in the header field **x-ErrorDescription**.

Function Overview

| Command | Effect | Access mode |
|----------------|--|-------------|
| info | Retrieve information about the document | r |
| get | Fetch (a range of) a content unit of a component | r |
| docGet | Fetch the whole content of a document | r |
| create | Create a new document | c |
| mCreate | Create several new documents | c |

Functions

| | | |
|-------------------|--|---|
| append | Append data to a content unit | u |
| update | Modify an existing document | u |
| delete | Delete a document or a component | d |
| search | Search for a text pattern within a content unit | r |
| attrSearch | Search for one or more attributes within a document (search within a print list) | r |
| putCert | Transfer client (for example, the R/3 System) certificate | - |
| serverInfo | Retrieve information about the content server and the corresponding content repositories | - |

Access Functions

Access Functions

info

Effect

Document information is retrieved. As well as the document header information, the server sends information on all components. If information on only one component is required, a `compId` should be specified. The command `info` has the same effect as the command `docGet`, except that with `info` no component data is transferred.

Using `resultAs`, you can specify the format in which the information is to be provided. Return values can be provided in an ASCII format which can be parsed easily or in an HTML file. Use of `resultAs` is optional, `ascii` is standard. The format is defined further below.

If `resultAs=ascii` and the function is executed successfully, the data is transferred as an entity body in *multipart/form-data* format (see RFC 1867) as a response to an `HTTP GET-Request`.

Default

Standard information about the document header and the components of the addressed document is returned in ASCII format. The results are given in ASCII format.

Access Mode

read (r)

Client → Server

The client sends an `HTTP-GET-Request`. The URL contains the following parameters:

| Parameter | Optional/Mandatory | Default | Sign |
|-------------------------|--------------------|--------------------|------|
| <code>contRep</code> | mandatory | | X |
| <code>docId</code> | mandatory | | X |
| <code>compId</code> | optional | | |
| <code>pVersion</code> | mandatory | | |
| <code>resultAs</code> | optional | <code>ascii</code> | |
| <code>accessMode</code> | s-mandatory | | X |
| <code>authId</code> | s-mandatory | | X |
| <code>expiration</code> | s-mandatory | | X |
| <code>secKey</code> | optional | | |



s-mandatory means that this parameter must only be specified if the URL is signed.

info**Example**

`http://pswdf009:1080/ContentServer/ContentServer.dll?info&pVersion=0045&contRep=K1&docId=361A524A3ECB5459E0000800099245EC`

The example is a request for information about the document header and all the document components. Information about the document header and all components of the document is requested.

Server → Client

The server answers the request with a response. The response status code indicates the outcome of the call.

| HTTP Status Code | Meaning |
|-----------------------------|---------------------------------------|
| 200 (OK) | OK, information delivered |
| 400 (bad request) | Unknown function or unknown parameter |
| 401 (unauthorized) | Breach of security |
| 404 (not found) | Document/component not found |
| 409 (conflict) | Administrative data not accessible |
| 500 (Internal Server Error) | Internal error in content server |

The response header contains the following information about the document:

| Keyword | Format | Meaning |
|-----------------------|----------------|--|
| Content-Type | String | Content-Type (if known) |
| boundary | String | Separator between individual components |
| Content-Length | Integer string | Entire length of the body actually transferred |
| X-dateC | YYYY-MM-DD | Creation date (UTC) |
| X-timeC | HH:MM:SS | Creation time |
| X-dateM | YYYY-MM-DD | Last changed on (UTC) |
| X-timeM | HH:MM:SS | Last changed at (UTC) |
| X-numberComps | Integer string | Number of components |
| X-contentRep | String | Content repository |
| X-docId | String | Document ID |
| X-docStatus | String | Status |
| X-pVersion | String | Version |

Each time the function is called, all document header information is provided and, if no particular component is addressed, information on all components. If information is required on only one

component, output can be limited to this component by specifying the `compId`. The following combinations are possible:

- `docId=ID, compId=ID`:
Information about the document header and one component is provided.
- `docId=ID`:
Information about the document header and all components of the document is provided.

The component header contains the following information about the component:

| Keyword | Format | Meaning |
|-------------------------------|----------------|---|
| <code>Content-Type</code> | String | <code>Content-Type</code> (if known) |
| <code>charset</code> | String | Character set (if known) |
| <code>version</code> | String | Application version used to create the content of the component |
| <code>Content-Length</code> | Integer string | Actual body size in the response, always 0 |
| <code>X-Content-Length</code> | Integer string | Size of the component in bytes |
| <code>X-compId</code> | String | Component ID |
| <code>X-compDateC</code> | YYYY-MM-DD | Creation date (UTC) |
| <code>X-compTimeC</code> | HH:MM:SS | Creation time |
| <code>X-compDateM</code> | YYYY-MM-DD | Last changed on (UTC) |
| <code>X-compTimeM</code> | HH:MM:SS | Last changed at (UTC) |
| <code>X-compStatus</code> | String | Component status |
| <code>X-pVersion</code> | String | Interface version |

There are two ways of coding the results in the response body. The parameter `resultAs` controls coding.

1. `resultAs=ascii (default)`

The server sends a response in *multipart/form-data* format (see RFC 1867). The total length of the body is specified by the parameter `Content-Length` in the response header. The individual parts of the response body are separated by a boundary defined in the response header. Each part represents one component. Each component has a component header and a component body with Length 0, because no component data is transferred (in contrast to the `docGet` command). The component parameter `Content-Length` is therefore always immediately set to 0. Alternatively, the component length can be determined by the parameter `X-Content-Length`.

If the `charset` of a component is known, it must be transferred as a `Content-Type` parameter. Likewise, the parameter `version` (that is, the version number of the application used to create the component content (see [Parameters and Key Words](#)

info

[[Page 291](#)]) for a component must be transferred as a **Content-Type** parameter, if known.



```
HTTP/1.1 200 (OK)
Server: Microsoft-IIS/4.0
Date: Wed, 04 Nov 1998 07:41:03 GMT
Content-Type: multipart/form-data;
boundary=A495ukjfasdfddrg4hztzu...
Content-Length: 32413
X-dateC: 1998-10-07
X-timeC: 07:55:57
X-dateM: 1998-10-07
X-timeM: 07:55:57
X-contentRep: K1
X-numberComps: 2
X-docId: ID
X-docStatus: online
X-pVersion: 0045

--A495ukjfasdfddrg4hztzu898aA0jklmAxvla12319981147528895
Content-Type: application/x-alf; charset=
Content-Length: 0
X-compId: descr
X-Content-Length: 2591
X-compDateC: 1998-10-07
X-compTimeC: 07:55:57
X-compDateM: 1998-10-07
X-compTimeM: 07:55:57
X-compStatus: online
X-pVersion: 0045

--A495ukjfasdfddrg4hztzu898aA0jklmAxvla12319981147528895
Content-Type: application/x-alf; charset=
Content-Length: 0
X-compId: data
X-Content-Length: 29213
X-compDateC: 1998-10-07
X-compTimeC: 07:55:57
X-compDateM: 1998-10-07
X-compTimeM: 07:55:57
X-compStatus: online
X-compStatus: online
X-pVersion: 0045

--A495ukjfasdfddrg4hztzu898aA0jklmAxvla12319981147528895--
```

With the **info** command for an empty document, for example, the response body contains the following:

```
--A495ukjfasdfddrg4hztzu898aA0jklmAxvla1231999102562159269
--A495ukjfasdfddrg4hztzu898aA0jklmAxvla1231999102562159269--
```

2. **resultAs=html**

If **resultAs=html** is set, the server sends an HTML page. The structure of the HTML page is not specified and graphical elements can be used freely.

get

get

Effect

A content unit of a component or a range within a content unit is retrieved from the content repository. The parameters `ContRep`, `docId` and `compId` describe the component. The range of the content unit is described by `fromOffset` and `toOffset`.

If the function is executed successfully, the content unit is transferred from the server to the client as an entity body in the response to an HTTP GET-Request.

Default

If no `compId` is specified, the following conditions must be tested in the corresponding order:

1. If there is a component "data", this component is returned.
2. If there is a component "data1", this component is returned.

The function returns error 404 (not found), if a wrong `compId` or no `compId` was specified and none of the above conditions is fulfilled.

Access mode

Read (r)

Client → Server

The client sends an **HTTP-GET-Request**. The URL contains the following parameters:

| Parameter | Optional/Mandatory | Default | Sign |
|-------------------------|--------------------|-----------|------|
| <code>contRep</code> | mandatory | | X |
| <code>docId</code> | mandatory | | X |
| <code>compId</code> | optional | see above | |
| <code>pVersion</code> | mandatory | | |
| <code>fromOffset</code> | optional | 0 | |
| <code>toOffset</code> | optional | -1 | |
| <code>accessMode</code> | s-mandatory | | X |
| <code>authId</code> | s-mandatory | | X |
| <code>expiration</code> | s-mandatory | | X |
| <code>secKey</code> | optional | | |



s-mandatory means that this parameter must only be specified if the URL is signed.

Example

```
http://pswdf009:1080/ContentServer/ContentServer.dll?get&pVersion=0045&
contRep=K1&docId=361A524A3ECB5459E0000800099245EC&compId=data
```

The document component “data” is requested.

Server → Client

The server answers the request with a response. The response status code indicates the outcome of the call.

| HTTP Status Code | Meaning |
|-----------------------------|--|
| 200 (OK) | OK, content unit of component is transferred |
| 400 (bad request) | Unknown function or unknown parameter |
| 401 (unauthorized) | Breach of security |
| 404 (not found) | Document/component not found |
| 409 (conflict) | Document/component not accessible |
| 500 (internal server error) | Internal error in content server |

The response header contains the following standard information about the document:

| Keyword | Meaning |
|----------------|---|
| Content-Type | Content-Type |
| charset | The character set of the component (as a Content-Type parameter). |
| version | The version of the component (as a Content-Type parameter). |
| Content-Length | Length of document |

The response Content-Type depends on the Content-Type of the component requested. If the **charset** of a component is known, it must be transferred as a **Content-Type** parameter.

Likewise, the parameter **version** (that is, the version number of the application used to create the component content (see [Parameters and Key Words \[Page 291\]](#))) for a component must be transferred as a **Content-Type** parameter, if known.

The content unit (or range within the content unit) of the component is transferred in the response body.

docGet**docGet****Effect**

The entire content of a document is retrieved from the content repository.

If an incorrect `docId` was specified, error 404 (not found) occurs. The error is always error 404 (not found).

If the function is executed successfully, the data is transferred as an entity body in *multipart/form-data* format (see RFC 1867) as a response to an `HTTP GET-Request`.

Default

-

Access mode

Read (r)

Client → Server

The client sends an `HTTP-GET-Request`. The URL contains the following parameters:

| Parameter | Optional/Mandatory | Default | Sign |
|------------|--------------------|---------|------|
| contRep | mandatory | | X |
| docId | mandatory | | X |
| pVersion | mandatory | | |
| accessMode | s-mandatory | | X |
| authId | s-mandatory | | X |
| expiration | s-mandatory | | X |
| secKey | optional | | |



s-mandatory means that this parameter must only be specified if the URL is signed.

Example

```
http://pswdf009:1080/ContentServer/ContentServer.dll?docGet&pVersion=00
45&contRep=K1&docId=361A524A3ECB5459E0000800099245EC
```

The entire content of a document is transferred to the client.

Server → Client

The server answers the request with a response. The response status code indicates the outcome of the call.

| HTTP Status Code | Meaning |
|-----------------------------|---------------------------------------|
| 200 (OK) | OK, document is transferred |
| 400 (bad request) | Unknown function or unknown parameter |
| 401 (unauthorized) | Breach of security |
| 404 (not found) | Document/component not found |
| 409 (conflict) | Document/component not accessible |
| 500 (internal server error) | Internal error in content server |

The server sends a response in *multipart/form-data* format (see RFC 1867). The individual parts of the response body are separated by a boundary defined in the response header. In contrast to the `info` command, when `docGet` is used, components are actually transferred and the Length of the transferred components is specified in the field `Content-Length` of the relevant component, that is, `Content-Length` and `x-Content-Length` have identical values. The response header contains the following information about the document:

| Keyword | Format | Meaning |
|-----------------------------|---------------|--|
| <code>Content-Type</code> | String | Content-Type, always multipart/form-data |
| <code>boundary</code> | String | Separator between individual components |
| <code>Content-Length</code> | Integerstring | Entire length of the body actually transferred |
| <code>X-datetimeC</code> | YYYY-MM-DD | Creation date (UTC) |
| <code>X-timeC</code> | HH:MM:SS | Creation time |
| <code>X-datetimeM</code> | YYYY-MM-DD | Last changed on (UTC) |
| <code>X-timeM</code> | HH:MM:SS | Last changed at (UTC) |
| <code>X-numComps</code> | Integerstring | Number of components |
| <code>X-contRep</code> | String | Content repository |
| <code>X-docId</code> | String | Document ID |
| <code>X-docStatus</code> | String | Status |
| <code>X-pVersion</code> | String | Version |

The component header contains the following information about the component:

| Keyword | Format | Meaning |
|---------------------------|--------|-------------------------|
| <code>Content-Type</code> | String | Content-Type (if known) |

docGet

| | | |
|-------------------------|---------------|--|
| charset | String | Character set (if known) |
| version | String | |
| Content-Length | Integerstring | Actual Body size in the response, always 0 |
| X-Content-Length | Integerstring | Size of the component in bytes |
| X-compId | String | Component ID |
| X-compdateC | YYYY-MM-DD | Creation date (UTC) |
| X-compTimeC | HH:MM:SS | Creation time |
| X-compDateM | YYYY-MM-DD | Last changed on (UTC) |
| X-compTimeM | HH:MM:SS | Last changed at (UTC) |
| X-compStatus | String | Component status |
| X-pVersion | String | Interface version |

If the **charset** of a component is known, it must be transferred as a **Content-Type** parameter. Likewise, the parameter **version** (that is, the version number of the application used to create the component content (see [Parameters and Key Words \[Page 291\]](#))) for a component must be transferred as a **Content-Type** parameter, if known.

Example:

```
HTTP/1.1 200 (OK)
Server: Microsoft-IIS/4.0
Date: Wed, 04 Nov 1998 07:41:03 GMT
Content-Type: multipart/form-data; boundary=A495ukjf asdfddrg4hztzu...
...some more header information...
Content-Length: 32413
X-dateC: 1998-10-07
X-timeC: 07:55:57
X-dateM: 1998-10-07
X-timeM: 07:55:57
X-contRep: K1
X-numComps: 2
X-docId: ID
X-docStatus: online
X-pVersion: 0045

--A495ukjf asdfddrg4hztzu898aA0jklmAxcvla12319981147528895
Content-Type: application/x-alf; charset=
Content-Length: 2591
X-compId: descr
X-Content-Length: 2591
X-compDateC: 1998-10-07
X-compTimeC: 07:55:57
X-compDateM: 1998-10-07
X-compTimeM: 07:55:57
X-compStatus: online
X-pVersion: 0045
```

```
...component data ...
--A495ukjfасdfddrg4hztzu898aA0jklmAxcvla12319981147528895

Content-Type: application/x-alf; charset=
Content-Length: 29313
X-compId: data
X-Content-Length: 29213
X-compDateC: 1998-10-07
X-compTimeC: 07:55:57
X-compDateM: 1998-10-07
X-compTimeM: 07:55:57
X-compStatus: online
X-compStatus: online
X-pVersion: 0045

...component data ...
--A495ukjfасdfddrg4hztzu898aA0jklmAxcvla12319981147528895--
```

When the `docGet` command is used on an empty document, the following is an example of what could be in the Response-Body:

```
--A495ukjfасdfddrg4hztzu898aA0jklmAxcvla1231999102562159269
--A495ukjfасdfddrg4hztzu898aA0jklmAxcvla1231999102562159269--
```

create

create

Effect

A document with one or more components is stored in the content repository. The parameters **contRep**, **docId** and **compId** describe the component (see [Definition of Terms \[Page 231\]](#)). The function is used to create new documents. If a document already exists in the content repository, an error occurs in the function. The functions **update** and **append** can be used to modify existing documents. The **create** function always creates an entire document.

The function can be called once with an HTTP-PUT or POST (for further information, see [HTTP-PUT \[Page 260\]](#) and [HTTP-POST multipart/form-data \[Page 261\]](#)).

Default

A new document with the specified **docId** is created. One or more components are stored in the content repository. Protection is according to the standard set on the content server.

Access Mode

create (c)

Client → Server

The following parameters exist:

| Parameter | Optional/Mandatory | Default | Position (POST/PUT) | Sign (POST/PUT) |
|-----------------------|--------------------|----------------|-----------------------------|-----------------|
| contRep | mandatory | | URL/URL | X/X |
| compId | mandatory | | body/URL | -/X |
| docId | mandatory | | URL/URL | X/X |
| pVersion | mandatory | | URL/URL | |
| Content-Type | optional | | body/body | |
| charset | optional | | body/body | |
| version | optional | | body/body | |
| Content-Length | mandatory | | Header body/ Header body | |
| docProt | optional | server setting | URL/URL | X/X |
| accessMode | s-mandatory | | URL/URL | X/X |

create

| | | | | |
|-------------------|-------------|--|---------|-----|
| authId | s-mandatory | | URL/URL | X/X |
| expiration | s-mandatory | | URL/URL | X/X |
| secKey | optional | | URL/URL | |



For HTTP-POST, the **Content-Length** in the request header is the total length of the body and the **Content-Length** in each part header is the length of the individual content units. For HTTP-PUT, the **Content-Length** is always the total length of the body.

It makes a difference whether the parameter **docProt** is not transferred or whether nothing is transferred as **docProt (docProt=)**. In the first case, the content server default is used. In the second case it is specified explicitly that no protection exists.

s-mandatory means that this parameter must only be specified if the URL is signed.

The function can be executed in two ways. A single component can be transferred using an [HTTP-PUT \[Page 260\]](#). Alternatively, an [HTTP-POST \[Page 261\]](#) is used in the format **multipart/form-data**. In the first version, only one single component can be loaded on the server in each case. This restriction does not apply in the second version; 0 to n components can be transferred.

HTTP-PUT

HTTP-PUT

In this case, all of the parameters are entered in accordance with the table in the [create \[Page 258\]](#) section.



```
http://pswdf009:1080/ContentServer/ContentServer.dll?create&pVersion=0045  
&contRep=K1&docId=4B7689654E73D21197E70060B0672A3C&compId=data&  
Content-Length=300
```

The document component “data” is stored. The component data is transferred as an entity body.

HTTP-POST multipart/form-data

The data is transferred as **HTTP-POST** and as **multipart/form-data**. The document header information is transferred in the URL. One or more components are transferred in the body. This version of the function is particularly suitable for transferring documents consisting of several components into the content repository, as a whole. The component information is specified in the header of each part; the data in the body.

In practice, this means that the URL contains the parameters **contRep**, **docId**, **pVersion**, **docProt**, **accessMode**, **authId**, **expiration** and **secKey**. All the other parameters are in the body.

The request body is in **multipart/form-data** format. With this format, it is possible to transfer several independent parts to an HTTP content server. The individual parts have a header and a body and are in MIME format (RFC 2045, 2046). This MIME format enables several components to be transferred to the content server simultaneously. If an error occurs when storing a component, the entire action is cancelled.

The parameters **compId** and **Content-Type** are contained in the header of each part. The **CompId** is transferred in field **X-compId**. The component length is in the field **Content-Length**. The parameters **charset** and **version** can be appended to the **Content-Type**.

Example 1

```
http://pswdf009:1080/ContentServer/ContentServer.dll?create&pVersion=00  
45&contRep=K1&docId=4B7689654E73D21197E70060B0672A3C
```

A document consisting of one or more components is transferred in **multipart/form-data** format.

Document header

```
Content-Type: multipart/form-data;  
boundary=A495ukjfasdfddrg4hztzu898aA0jklm  
...some more header information...  
Content-Length: 32413
```

Content part

```
--A495ukjfasdfddrg4hztzu898aA0jklm  
X-compId: data  
Content-Type: application/msword; charset=ISO-8859-1; version=6  
Content-Length: 4242  
... 4242 Bytes Data ...  
--A495ukjfasdfddrg4hztzu898aA0jklm--
```

Example 2 (Create with 0 Components)

```
http://pswdf009:1080/ContentServer/ContentServer.dll?create&pVersion=00  
46&contRep=M1&docId=3810FF00804C257DE10000009B38FA09&docProt=ud&accessM  
ode=c&authId=CN%3DKPR&expiration=19991025080635&secKey=MIIBlQYJKoZIhvCN  
A ...
```

HTTP-POST multipart/form-data

A document consisting of one or more components is transferred in `multipart/form-data` format.

Document header

```
Content-Type: multipart/form-data; boundary=KoZIhvcNAQcB  
...some more header information...  
Content-Length: 38
```

Content part

```
--KoZIhvcNAQcB  
--KoZIhvcNAQcB--
```

Server → Client

The server answers the request with a response. The response status code indicates the outcome of the call.

| HTTP Status Code | Meaning |
|-----------------------------|---------------------------------------|
| 201(created) | OK, document(s) created |
| 400 (bad request) | Unknown function or unknown parameter |
| 401 (unauthorized) | Breach of security |
| 403 (forbidden) | Document already exists |
| 500 (internal server error) | Internal error in content server |

The content server must set the dates (`dateC` and `compDateC`) and the times (`timeC` and `compTimeC`) for creating components and the document.

mCreate

Effect

One or more documents each with one or more components are stored in the content repository. The function has a similar effect to that of several sequential `create` functions, but is significantly more efficient for signed URLs in storing many new documents at once.

Within an `mCreate` call, objects can only be stored in one and the same content repository. The content repository is described by the URL parameter `contRep`.

The individual components of the documents are transferred in a *multipart/form-data* entity body. Components of the same document must be transferred one after the other so that transfer of a document can be assumed to be complete, as soon as a component from a different document begins.

The parameter `docId` is absolutely necessary for all components (each multipart-part) and is entered as the headerfield "`X-docId`".

Storage of one document is performed within one transaction, but not the storage of all documents transferred in one `mCreate` call.

Access mode

Create (c)

Client → Server

The client sends an **HTTP-POST-Request**. The following parameters exist:

| Parameter | Header Field in Body | Optional/Mandatory | Default | Position | Sign |
|---------------------------|---------------------------|--------------------|---------|--|----------------------------|
| <code>contRep</code> | | mandatory | | URL | X |
| <code>compId</code> | " <code>X-compId</code> " | mandatory | | body | |
| <code>docId</code> | " <code>X-docId</code> " | mandatory | | body (1. <code>docId</code> also in URL) | X (1. <code>docId</code>) |
| <code>pVersion</code> | | mandatory | | URL | |
| <code>Content-Type</code> | | optional | | body | |
| <code>charset</code> | | optional | | body | |
| <code>version</code> | | optional | | body | |

mCreate

| | | | | | |
|-----------------------|--|-------------|----------------|------|---|
| Content-Length | | mandatory | | body | |
| docProt | | optional | server setting | URL | X |
| accessMode | | s-mandatory | | body | X |
| authId | | s-mandatory | | URL | X |
| expiration | | s-mandatory | | URL | X |
| secKey | | optional | | URL | |



s-mandatory means that this parameter must only be specified if the URL is signed.

The parameters for which the request body is specified as position are transferred in the header field of the multipart part of the corresponding components. For special parameters in this interface, the name of the header field is in column 2 of the table.

Server → Client

The server answers the request with a response. The response status code indicates the outcome of the call. A distinction is made between general status codes, which describe the success of the call as a whole, and specific status codes, which document the creation of individual documents.

| General HTTP Status Codes | Meaning |
|---------------------------------|---|
| 201(created) | OK, all documents were created |
| 250 (missing documents created) | OK, all missing documents were created In practice, this status can only occur when mcreate is called more than once. |
| 400 (bad request) | Unknown function or unknown parameter |
| 401 (unauthorized) | Breach of security |
| 500 (internal server error) | Internal error in content server |

| Specific HTTP Status Codes | Meaning |
|----------------------------|---------|
| | |

| | |
|-----------------------------|----------------------------------|
| 201(created) | OK, document created |
| 403 (forbidden) | Document already exists |
| 500 (internal server error) | Internal error in content server |

An ASCII text must be returned whether the function is executed successfully or whether an error occurs. The documents stored (HTTP status code 201) and/or not stored (HTTP status code 403) are specified in this text. The following format is used:

docId="string";retCode="integerstring";errorDescription="string";CRLF

The value of the parameter **retCode** is the corresponding specific HTTP status code.

As a summary, the response body contains the following standard information about each document:

| Keyword | Format | Meaning |
|------------------|---------------|--------------------------------------|
| docId | string | Document ID |
| retCode | Integerstring | HTTP status code |
| errorDescription | string | Text explaining the error (optional) |

append

append

Effect

Data is appended to a content unit of a component in the content repository. The parameters **ContRep**, **docId** and **compId** describe the component (see [Definition of Terms \[Page 231\]](#)). The document addressed and the corresponding component must exist.

Default

Data is appended to the content unit of the addressed component.

Access mode

change (u)

Client → Server

The client sends an **HTTP-PUT-Request**. The URL contains the following parameters:

| Parameter | Optional/Mandatory | Default | Sign |
|-------------------|--------------------|---------|------|
| contRep | mandatory | | X |
| docId | mandatory | | X |
| compId | mandatory | | X |
| pVersion | mandatory | | |
| accessMode | s-mandatory | | X |
| authId | s-mandatory | | X |
| expiration | s-mandatory | | X |
| secKey | optional | | |



s-mandatory means that this parameter must only be specified if the URL is signed.

The data to be appended is transferred as an entity body.

Example

`http://pswdf009:1080/ContentServer/ContentServer.dll?append&pVersion=0045&contRep=K1&docId=4B7689654E73D21197E70060B0672A3C&compId=data&Content-Length=980`

Data transferred in the request body is appended to the content unit of the component "data" in the specified document.

Server → Client

The server answers the request with a response. The response status code indicates the outcome of the call.

| HTTP Status Code | Meaning |
|-----------------------------|---------------------------------------|
| 200 (OK) | OK, data appended |
| 400 (bad request) | Unknown function or unknown parameter |
| 401 (unauthorized) | Breach of security |
| 404 (not found) | Document/component not found |
| 409 (conflict) | Document/component not accessible |
| 500 (Internal Server Error) | Internal error in content server |

The content server must set the dates (`dateM` and `compDateM`) and the times (`timeM` and `compTimeM`) for changing components and the document.

update

update

Effect

One or more components of a document in the content repository are overwritten. The parameters `ContRep`, `docId` and `compId` describe the component (see [Definition of Terms \[Page 231\]](#)). The function is used to modify existing documents and components and can be called once with an [HTTP-PUT \[Page 270\]](#) or [HTTP-POST \[Page 271\]](#). The function can be called once with a HTTP-PUT or POST (see below for details).

The variant `HTTP-PUT` is used to create or overwrite a component of an existing document.

The variant `HTTP-POST` (multipart/form_data) is used to bring an entire document up to date. When this variant is used, the entire document is overwritten, not only individual components.

Default

One or more components are stored in the content repository. Protection is according to the standard set on the content server.

Access mode

change (u);

Client → Server

The following parameters exist:

| Parameter | Optional/mandatory | Default | Position (POST/PUT) | Sign (POST/PUT) |
|-----------------------------|--------------------|---------|---------------------|-----------------|
| <code>contRep</code> | mandatory | | URL/URL | X/X |
| <code>compId</code> | mandatory | | body/URL | -/X |
| <code>docId</code> | mandatory | | URL/URL | X/X |
| <code>pVersion</code> | mandatory | | URL/URL | |
| <code>Content-Type</code> | optional | | body/body | |
| <code>charset</code> | optional | | body/body | |
| <code>version</code> | optional | | body/body | |
| <code>Content-Length</code> | mandatory | | body/body | |
| <code>accessMode</code> | s-mandatory | | URL/URL | X/X |

update

| | | | | |
|-------------------|-------------|--|---------|-----|
| authId | s-mandatory | | URL/URL | X/X |
| expiration | s-mandatory | | URL/URL | X/X |
| secKey | optional | | URL/URL | |



s-mandatory means that this parameter must only be specified if the URL is signed.

The function can be executed in two ways. Either a single component can be transferred using an [HTTP-PUT \[Page 270\]](#) or an [HTTP-POST \[Page 271\]](#) in **multipart/form-data** format is used. If the former version (**HTTP-PUT**) is used, only one component at a time can be loaded onto the content server. There is no such restriction if the latter version (**HTTP-POST**) is used.

HTTP-PUT**HTTP-PUT**

This variant is used to create or overwrite an individual component of a document.

For further details, see the command [create \[Page 260\]](#).

HTTP-POST multipart/form-data

Similarly to the `create` function, this variant of the function is used to replace a complete document with all its components in the content repository at once. Document components not already in the content repository are created if necessary. Components in the content repository that are not transferred when the `update` function is executed are considered obsolete and deleted. The structure details of the request is the same as for the function [create \[Page 261\]](#).

Server → Client

The server answers the request with a response. The response status code indicates the outcome of the call.

| HTTP Status Code | Meaning |
|-----------------------------|---------------------------------------|
| 200 (OK) | OK, document(s)/component(s) changed |
| 400 (bad request) | Unknown function or unknown parameter |
| 401 (unauthorized) | Breach of security |
| 404 (not found) | Document/component not found |
| 409 (conflict) | Document/component not accessible |
| 500 (Internal Server Error) | Internal error in content server |

The content server must set the dates (dateM, compDateM and compDateC) and the times (timeM, compTimeM and compTimeC) for changing components and the document.

delete

delete

Effect

A component or an entire document is deleted. A document to be deleted is addressed via **contRep** and **docId**. The parameters **contRep**, **docId** and **compId** identify the component to be deleted.

Default

The document, including all administrative data (document header and components) and the content, is deleted completely.

Access mode

delete (d)

Client → Server

The client sends an **HTTP-GET-Request**. The URL contains the following parameters:

| Parameter | Optional/Mandatory | Default | Sign |
|-------------------|--------------------|----------------|------|
| contRep | mandatory | | X |
| docId | mandatory | | X |
| compId | optional | all components | X |
| pVersion | mandatory | | |
| accessMode | s-mandatory | | X |
| authId | s-mandatory | | X |
| expiration | s-mandatory | | X |
| secKey | optional | | |



s-mandatory means that this parameter must only be specified if the URL is signed.

Example

`http://pswdf009:1080/ContentServer/ContentServer.dll?delete&pVersion=0045&contRep=K1&do
cld=4B7689654E73D21197E70060B0672A3C&compId=data`

Component “data” is deleted in the named document.

delete

Server → Client

The server answers the request with a response. The response status code indicates the outcome of the call.

| HTTP Status Code | Meaning |
|-----------------------------|---------------------------------------|
| 200 (OK) | OK, document/component(s) deleted |
| 400 (bad request) | Unknown function or unknown parameter |
| 401 (unauthorized) | Breach of security |
| 404 (not found) | Document/component not found |
| 409 (conflict) | Document/component not accessible |
| 500 (Internal Server Error) | Internal error in content server |

search

search

Effect

This function searches for a text pattern in the content unit of a component. The range of the search can be restricted. The search begins at the point specified by `fromOffset` and continues until the `toOffset` point. If `fromOffset > toOffset`, the function searches the component backwards.

A text pattern is found if the following conditions are met:

- if `fromOffset <= toOffset`
- the location of first character of the text found is greater than or equal to `fromOffset`
- the location of the last character of the text found is smaller than or equal to `toOffset`
- if `fromOffset >= toOffset`
- the location of the last character of the text found is less than or equal to `fromOffset`
- the location of the first character of the text found is greater than or equal to `toOffset`

The pattern contains the string searched for. The string can contain blank characters.

The number of result entries and up to `numResults` hits are returned as the result. A hit is the entry of the character position. The character position is the position of the found location in relation to the start of the document. The position of the first character of the text searched for is defined as the position of the found location, irrespective of the search direction.

Default

The pattern is searched for in the whole addressed component.

Access mode

Read (r)

Client → Server

The client sends an **HTTP-GET-Request**. The URL contains the following parameters:

| Parameter | Optional/Mandatory | Default | Sign |
|-----------|--------------------|---------|------|
| contRep | mandatory | | X |
| docId | mandatory | | X |
| pattern | mandatory | | |
| compld | mandatory | | |
| pVersion | mandatory | | |

| | | | |
|---------------|-------------|----|---|
| caseSensitive | optional | n | |
| fromOffset | optional | 0 | |
| toOffset | optional | -1 | |
| numResults | optional | 1 | |
| accessMode | s-mandatory | | X |
| authId | s-mandatory | | X |
| expiration | s-mandatory | | X |
| secKey | optional | | |



s-mandatory means that this parameter must only be specified if the URL is signed.

Example

```
http://pswd009:1080/ContentServer/ContentServer.dll?search&pVersion=00
45&contRep=K1&docId=4B7689654E73D21197E70060B0672A3C&compId=data&patter
n=Manfred%20M%FCller&fromOffset=80
```

A search for “Manfred Mueller” is carried out in the component data of the named object from Offset=80 to the end.

Server → Client

The server answers the request with a response. The response status code indicates the outcome of the call.

| HTTP Status Code | Meaning |
|-----------------------------|---------------------------------------|
| 200 (OK) | OK, component was searched |
| 400 (bad request) | Unknown function or unknown parameter |
| 401 (unauthorized) | Breach of security |
| 404 (not found) | Document/component not found |
| 409 (conflict) | Document/component not accessible |
| 500 (internal server error) | Internal error in content server |

The result of the search is the number of hits and the Offset for each hit. An ASCII string with the following structure is returned:

number;offset;offset;...

search

There are no blank characters between the individual characters. There is a semicolon between the values and at the end.



2;122;222;

attrSearch

This function is used for attribute-based searches in print lists (attribute search). It is a prerequisite of this search that a print list has a description file (`compId=descr`) as well as a data file (`compId=data`). Unlike [search \[Page 274\]](#) this is a specific search, which is carried out in the description file of a print list (`compId=descr`). Only the description file is relevant for the implementation of an attribute search.

Basic Principles

The structure of the description file is explained below with the aid of an example.



Content of a description file (extract; the periods stand for blank characters):

```
0 72 DPRL
73 0 DKEYclient.....0 3
73 0 DKEYcompany_code.....3 5
73 0 DKEYaccount_number.....8 7
73 0 DKEYcustomer_name.....15 25
73 138 DAIN00100010147119Broeselpc
211 120 DAIN0010000201471290belixplc
...
1147 1 DEPL
```

The description file consists of a sequence of lines (**index lines**). These index lines describe attributes of a range of the relevant data file.

An index line consists of:

- Offset and Length in the data file:
Specification of the Offset and the Length of the range described (in bytes) relative to the start of the data file.
- Record type
Type of line. The record type consists of four bytes. The following record types are used:
 - DPRL
Prolog
 - DKEY
Description of attributes
 - DAIN
Value of attributes
 - DEPL
Epilog

attrSearch

The various record types occur in the description file in the order specified here. The fact that the DAIN lines come after the DKEY lines is of particular note.

Only DKEY and DAIN lines are decisive for the attribute search. The DKEY lines specify the attributes and how they are stored. The DAIN lines specify the attribute values.

- Parameter

Remaining content of the index line dependent on the record type.

The individual index lines are closed with linefeed (0x0A). The value for the "Offset in the data file" increases steadily within the description file.



Interpreted content of the description file (extract):

| Offset and Length in the Data File | Record Type | Parameter | | | | | |
|------------------------------------|-------------|-----------|----------------|------|-------------|---------------|--|
| 0 | 72 | DPRL | | | | | |
| 73 | 0 | DKEY | Client | 0 | 3 | | |
| 73 | 0 | DKEY | Company code | 3 | 5 | | |
| 73 | 0 | DKEY | Account number | 8 | 7 | | |
| 73 | 0 | DKEY | Customer name | 15 | 25 | | |
| 73 | 138 | DAIN | 001 | 0001 | 01471 19 | Bröse lplc | |
| 211 | 120 | DAIN | 001 | 0002 | 01471 29 | Obeli xplc | |
| ... | | | | | | | |
| 1147 | 1 | DEPL | | | | | |

The structure of the DKEY and DAIN lines is described in detail below.

DKEY lines (description of attributes)

| Content | Length (in Bytes) |
|----------------------|-------------------|
| Offset in data file | Variable |
| Separator (space) | 1 |
| Length in data file | Variable |
| Separator (space) | 1 |
| Record type ("DKEY") | 4 |
| Attribute names | 40 |

| | |
|-----------------------------------|---|
| Offset in the DAIN line parameter | 3 |
| Length in the DAIN line parameter | 3 |

The DKEY lines specify the names (attribute names) and the structure (Offset and Length in the DAIN line parameter) of the attributes that occur in the DAIN lines. The Offset position is counted starting with 0. Each DKEY line describes one particular attribute.

The values "Offset and Length in data file" are not relevant here.

DAIN lines (values of attributes)

| Content | Length (in Bytes) |
|----------------------|-------------------|
| Offset in data file | Variable |
| Separator (space) | 1 |
| Length in data file | Variable |
| Separator (space) | 1 |
| Record type ("DAIN") | 4 |
| Parameter | Variable |

Each DAIN line specifies the attribute value for a specific range of the data file. The DAIN line parameter consists of the attribute values corresponding to the standards in the DKEY lines. Blank characters at the end of the DAIN lines are irrelevant. If the DAIN line contains less data than is specified in the DKEY lines, the attributes must be filled with blank characters.

Here, the specifications "Offset and Length in data file" are relevant. They relate to the data file and specify the range for which the given attribute values are valid.



The content of the above example is as follows:

Description of attributes

| Attribute Name | Offset in the DAIN Line Parameter | Length in the DAIN Line Parameter |
|----------------|-----------------------------------|-----------------------------------|
| Client | 0 | 3 |
| Company code | 3 | 5 |
| Account number | 8 | 7 |
| Customer name | 15 | 25 |

Value of attributes

| Offset in Data File | Length in Data File | Attribute Name | Attribute Value |
|---------------------|---------------------|----------------|-----------------|
| 73 | 138 | Client | "001" |

attrSearch

| | | | |
|-----|-----|----------------|-------------|
| | | Company code | "00001" |
| | | Account number | "0147119" |
| | | Customer name | "Bröselplc" |
| 211 | 120 | Client | "001" |
| | | Company code | "00002" |
| | | Account number | "0147129" |
| | | Customer name | "Obelixplc" |
| ... | ... | ... | ... |

Effect

This function is used for attribute-based searches in print lists. The parameters `contRep` and `docId` describe the component. The parameter `pattern` specifies (as well as the pattern) the attribute to be searched for. The attribute is described by its Offset and Length. The pattern is made up of the Offset, followed by the character "+", followed by the Length, followed by the character "+", followed by the attribute value. If several attributes are to be searched for, the individual `patterns` should be separated by a #.

The patterns can contain any characters. Unsafe and reserved characters are coded as normal here. This is also true for the separator "#" if this occurs in the pattern (see [Coding in the URL \[Page 241\]](#)).

The result of the attribute search is the values for "Offset and Length in the data file" in the DAIN lines, that correspond to the pattern and are in the search range.

For a pattern to be found, the following conditions must be fulfilled:

- The value for the "Offset in the data file" in the DAIN line is within the search range.
- The attribute values given in the `pattern` match those in the DAIN line. This means that each attribute value specified by the `pattern` must be contained fully in the corresponding attribute value in the DAIN line.

The number of result entries (number of appropriate DAIN lines), as well as the result entries themselves (values for "Offset and Length in the data file" in the DAIN line) are returned as the result. The values are separated by a semicolon. The result entries are arranged according to the search direction. Control of the search direction is as for the [search \[Page 274\]](#) function. The number of results can be restricted by the parameter `numResults`.

Default

The pattern is searched for in the document addressed by the parameters `contRep` and `docId`. The `compId` is not specified in the call. The function always searches in the description file (`compId=descr`).

Access mode

Read (r)

Client → Server

The client sends an **HTTP-GET-Request**. The URL contains the following parameters:

| Parameter | Optional/Mandatory | Default | Sign |
|----------------------|--------------------|---------|------|
| contRep | mandatory | | X |
| docId | mandatory | | X |
| pattern | mandatory | | |
| pVersion | mandatory | | |
| caseSensitive | optional | n | |
| fromOffset | optional | 0 | |
| toOffset | optional | - 1 | |
| numResults | optional | 1 | |
| accessMode | s-mandatory | | X |
| authId | s-mandatory | | X |
| expiration | s-mandatory | | X |
| secKey | optional | | |



s-mandatory means that this parameter must only be specified if the URL is signed.

Example

```
http://pswdf009:1080/ContentServer/ContentServer.dll?attrSearch&pVersio
n=0045&contRep=K1&docId=361A524A3ECB5459E0000800099245EC&pattern=3+5+12
345#15+25+GmbH&numResults=5
```

A search is run in a component for 12345 in the attribute with Offset 3 and Length 5 (in the example, this is the *company code*) and also for the content *GmbH* in the attribute with Offset 15 and Length 25 (in the example, this is the *customer name*). Up to 5 hits are found.



As can be seen from the example, # is not coded.

attrSearch**Server → Client**

The server answers the request with a response. The response status code indicates the outcome of the call.

| HTTP Status Code | Meaning |
|-----------------------------|---------------------------------------|
| 200 (OK) | OK, component was searched |
| 400 (bad request) | Unknown function or unknown parameter |
| 401 (unauthorized) | Breach of security |
| 404 (not found) | Document/component not found |
| 409 (conflict) | Document/component not accessible |
| 500 (internal server error) | Internal error in content server |

The result of the search is the number of hits and the Offset and Length for each hit:

number;offset;length;...



2 ; 73 ; 138 ; 211 ; 120 ;

If an attribute search cannot be carried out properly because the values in the attribute do not match the standards in the DKEY lines (for example, attribute names wrong or attribute value too long in the **pattern**), status code 400 (bad request) is returned.

If, however, nothing is found, the status code is set to 200 (OK) and 0 is returned as the result.

Administration Functions

In the current HTTP Content Server interface 4.5, only two administrative functions are defined: `putCert` and `serverInfo`. Further administration functions will be defined in later versions of the interface.

putCert

putCert

Effect

The client certificate is transferred. The system identifies itself via its authenticity (**authId**).

The client certificate (see [secKey \[Page 235\]](#)) is decoded in the message body and transferred in binary format.

For reasons of security, it is recommended that after the certificate has been transferred, manual action by an administrator is necessary before access is actually allowed. This could be a public key fingerprint check or any other plausibility check.

The logon procedure therefore consists of two steps:

- Certificate is transferred and entered in a central location
- Administrator allows access via a tool

The client can only access after the second step of this procedure. After the first step, the certificate is only created.

Access mode

Client → Server

The client sends an **HTTP-PUT-Request**.

| Parameter | Optional/Mandatory | Sign |
|-----------|--------------------|------|
| authId | mandatory | |
| pVersion | mandatory | |
| contRep | mandatory | |

The certificate is transferred in the request body, all the other parameters are transferred in the URL. The URL does not contain a **secKey**.

Server → Client

The server answers the request with a response. The response status code indicates the outcome of the call.

| HTTP Status Code | Meaning |
|-------------------|---------------------------------------|
| 200 (OK) | OK |
| 400 (bad request) | Unknown function or unknown parameter |

| | |
|-----------------------------|----------------------------------|
| 406 (not acceptable) | Certificate cannot be recognized |
| 500 (Internal Server Error) | Internal error in content server |

serverInfo

serverInfo

Effect

The function supplies information about the status of the content server and the content repositories that it manages.

Default

The standard information is returned to the content server addressed and the content repositories that it manages. The results are given in ASCII format.

Access mode

-

Client → Server

The client sends an **HTTP-GET-Request**. The following parameters exist:

| Parameter | Optional/Mandatory | Default | Sign |
|-----------|--------------------|----------------|------|
| contRep | optional | all components | |
| pVersion | mandatory | | |
| resultAs | optional | ascii | |

Example

```
http://pswdf009:1080/ContentServer/ContentServer.dll?serverInfo&pVersion=0045
```

The example requests information about the content server and all the content repositories it manages.

Server → Client

The server answers the request with a response. The response status code indicates the outcome of the call.

| HTTP Status Code | Meaning |
|-----------------------------|---------------------------------------|
| 200 (OK) | OK |
| 400 (bad request) | Unknown function or unknown parameter |
| 500 (internal server error) | Internal error in content server |

The following information about the content server status is provided:

| Keyword | Format | Meaning |
|---------|--------|---------|
| | | |

| | | |
|--------------------------------|------------|--|
| serverStatus | | Status of content server: [running stopped error] |
| serverVendorId | | Vendor and software version |
| serverVersion | | Version of server |
| serverBuild | | Build of server |
| serverTime | HH:MM:SS | Content server time (UTC) |
| serverDate | YYYY-MM-DD | Content server date (UTC) |
| serverStatusDescription | | Text describing server status |
| pVersion | | Content server interface version |

The following information about the status of each content repository is provided:

| Keyword | Format | Meaning |
|---------------------------------|---------------|--|
| contRep | | Content repository |
| contRepDescription | | Text describing content repository content |
| contRepStatus | | Status of content repository: [running stopped error] |
| contRepStatusDescription | | Text describing content repository status |



The specified parameters are mandatory. The list of parameters is designed with a view to extension.

For each function call, all information about the content server is provided. If no content repository is addressed, information on all content repositories is provided. **ContRep** can be used to limit the content repository information to a single content repository.

There are two ways of coding the results in the response body. The parameter **resultAs** controls coding.

1. **resultAs=ascii (default)**

A pure ASCII-Text is returned. The information about the content server is at the start of the string, the information about the content repositories follows. The following format is used:

- For the content server:

```
serverStatus="string";serverVendorId="string";serverTime="string";
serverDate="string";serverErrorDescription="string";pVersion="0045";CRLF
```

serverInfo

- For each content repository:

```
contRep="string";contRepDescription="string";contRepStatus="string";pVersion="0045";CRLF
```

If no value is entered, the value remains free.

```
contRepDescription="" ;contRepStatus="string";...
```

The order of the key words does not matter but there must not be any blank characters. The key words (together with their values) are separated from each other by a semicolon. The corresponding values are in quotation marks.

2. resultAs=html

If **resultAs=html** is set, the server sends an HTML page. The structure of the HTML page is not specified and graphical elements can be used freely.

Error Codes

An error occurring when the function is executed is recognizable from the HTTP status code.

| HTTP Status Code | Meaning | Used for |
|---------------------------------|--|--|
| 200 (OK) | OK, information/component is delivered/transferred/changed /appended/deleted | info, get, docGet, update, append, delete, putCert, search, attrSearch |
| 201(created) | OK, component(s) created (if create) OK, (all) document(s) created (if mCreate) | create, mCreate |
| 250 (missing documents created) | OK, all missing documents were created | mCreate |
| 400 (bad request) | Unknown function or unknown parameter | All functions |
| 401 (unauthorized) | Breach of security | info, get, docGet, create, update, append, delete, mCreate, search, attrSearch |
| 403 (forbidden) | Document/component already exists | create, mCreate |
| 404 (not found) | Document/component/content repository not found | info, get, docGet, update, append, delete, search, attrSearch |
| 406 (not acceptable) | Certificate cannot be recognized | putCert |
| 409 (conflict) | Document/component/administrative data is inaccessible | info, get, docGet, append, update, delete, search, attrSearch |
| 500 (Internal Server Error) | Internal error in content server | All functions |

If an error occurs, the content server must also deliver an ASCII string describing the error. The error must be entered in the header field **X-ErrorDescription**.

Appendix

Appendix

Parameters and Key Words

A parameter appears no more than once per URL. The parameters and key words defined are listed below in alphabetical order. The data type is given in square brackets (a “string” consists of characters from the ASCII character set and an “integerstring” consists of characters from the set {0,1,2,3,4,5,6,7,8,9}): The following parameters and key words are defined:

- **accessMode [string]**
Access mode
- **authId [string]**
Client ID
- **caseSensitive [y|n]**
Determines whether the search is case-sensitive. Standard is **n**.
 - caseSensitive=n
Search is not case-sensitive
 - caseSensitive=y
Search is case-sensitive
- **charset [string]**
Describes the character set in which the component content is encoded (for example, ISO-8859-1; see also RFC 2046). Further values can be defined, but must have an **x-** placed before them. The character set is transferred as a **Content-Type** parameter.
- **compDateC [string]**
Date component created (UTC); format: YYYY-MM-DD
- **compDateM [string]**
Date component last changed (UTC); format: YYYY-MM-DD
- **compId [string]**
Identifies a component within a document.



Additional information for partners who already support the SAP ArchiveLink interface:

Data files for stored print lists and outgoing documents are interpreted as **compId** “data”; the corresponding description files as interpreted as **compId** “descr”. Notes have **compId** “note”.

Predefined values for components:

- “**data**” data file
- “**descr**” attribute file
- “**note**” note

Parameters and Key Words

- **compTimeC [string]**
Time component created (UTC); format: HH:MM:SS
- **compTimeM [string]**
Time component last changed (UTC); format: HH:MM:SS
- **compStatus [online|offline]**
Status of component in the content repository. Meaning:
 - **online:**
Component known and accessible
 - **offline:**
Component known and currently inaccessible
- **Content-Disposition**
The **Content-Disposition** can be transferred as an additional parameter to the **compId**, if documents are transferred as **multipart/form-data** (see also [HTTP-POST multipart/form-data \[Page 261\]](#)). In this case, the **Content-Disposition** must be transferred at the same time as the **X-compId**.



This parameter can be ignored.

- **Content-Length [integer|string]**
Size of body or component in bytes. The parameter **Content-Length** can occur both in the response header and in part headers.
- **Content-Type [string]**
Identifies the **Content-Type** of a component or a transferred document. Can occur in the response header and in part headers. With **charset**, the character set used to write the component content can be specified as the **Content-Type** parameter.
- **contRep [string]**
Specifies the content repository.
- **contRepDate [string]**
Content repository date (UTC); Format: YYYY-MM-DD.
- **contRepDescription [string]**
Text describing content repository content.
- **contRepErrorDescription [string]**
Text describing a content repository error.
- **contRepStatus [running|stopped|error]**
Content repository status. Meaning:
 - **running**

Parameters and Key Words

- Content repository is running
 - **stopped**
 - Content repository has been stopped
 - **error**
 - Error in content repository
- **contRepTime [string]**
Content repository time (UTC); format HH:MM:SS
- **contRepVendorId [string]**
Vendor and version of content repository software
- **dateC [string]**
Date document created (UTC); format: YYYY-MM-DD.
- **dateM [string]**
Date document last changed (UTC); format: YYYY-MM-DD
- **docID [string]**
Unique identifier for document header
- **docProt [string]**
Document protection **docProt** controls protection of a document and its information.
docProt is a combination of the access rights **r**, **c**, **u**, or **d**. A standard that can be overwritten by the parameter is set on the content server.
For example, the combination **docProt=rcud** provides full protection for a document.
- **docStatus [online|offline]**
Status of document in the content repository. Meaning:
 - **online**
Document known and accessible
 - **offline**
Document known and inaccessible
- **expiration [string]**
Expiry time of a signed URL, (UTC) format: YYYYMMDDHHMMSS.
- **fromOffset [integerstring]**
Specifies the starting point for a search or the beginning of a byte range within the component. The default is 0. This parameter is needed to be able to read parts of the component (with print lists, for example).
- **numComps [integerstring]**
Number of components in a document

Parameters and Key Words

- **numResults [integerstring]**

Determines the maximum number of results (hits) a search can give.

- **pattern [string]**

Character pattern searched for in the free search. See the function [search \[Page 274\]](#).

The character patterns must be replaced by corresponding **escape** coding in the form **escape = % HEX HEX**.

The disallowed characters must be replaced by corresponding **escape** coding in the form **escape = "%" HEX HEX**.

The counterpart of **pattern [string]** in the attribute search is **pattern [integerstring+integerstring+string]**.

- **pattern [integerstring+integerstring+string]**

Attribute pattern searched for in the attribute search. See function [attrSearch \[Page 277\]](#).

The counterpart of **pattern [integerstring+integerstring+string]** in the free search is **pattern [string]**.

- **pVersion [string]**

Specifies the interface version. Versions 0021, 0030 and 0031 were defined for the SAP ArchiveLink interface. The HTTP Content Server interface begins with version 0045.

- **resultAs [string]**

Chooses the presentation form for the result of the **info** function. Standard is **ASCII**.

- **resultAs=ascii**

Results given in ASCII format

- **resultAs=html**

Results given in HTML format

- **resultCode [integerstring]**

Part of the ASCII string sent by the content server to the client after an **mCreate** call.
Contains the HTTP status code for the corresponding document.

- **secKey [string]**

Specifies an access key that can be used to check access authorization. The access key is generated by the R/3 System.

- **serverDate [string]**

Content server date (UTC); Format: YYYY-MM-DD.

- **serverErrorDescription [string]**

Text describing content server error.

- **serverStatus [running|stopped|error]**

Content server status. Meaning:

- **running**

Parameters and Key Words

- Content server is running
 - **stopped**
 - Content server has been stopped
 - **error**
 - Error in content server
- **serverTime [string]**
Content server time (UTC); format: HH:MM:SS
- **serverVendorId [string]**
Vendor and version of content server software
- **timeC [string]**
Document creation date (UTC); format: HH:MM:SS
- **timeM [string]**
Time document last changed (UTC); format: HH:MM:SS
- **toOffset [integerstring]**
Specifies the end of a byte range within the component. The default -1 means that the search should continue to the end of the component. **toOffset** has priority over a potential Content-Range.
- **version [string]**
Describes the application version used to create a document/component. The version is transferred as the **Content-Type** parameter. For some MIME-Types, version numbers are registered at IANA, so that they are used in the same way worldwide. For example, versions 2w, 4, 5 and 6 are used for application/msword.
- **serverStatusDescription [string]**
Header field in which the content server enters an explanatory text if an error occurs.

Information on Migrating Existing Archives

Information on Migrating Existing Archives



This section is only relevant for those partners who have supported the SAP ArchiveLink interface and now want to support the new HTTP Content Server interface.

When a component is stored, the MIME type is transferred in the field **Content-Type** in the request header, see RFC 2045 / RFC 2046. The content server holds the MIME type, but does not evaluate it. The MIME type is used as the response **Content-Type** (for the **get** function, for example).

Until now, only document classes such as ALF, FAX, DOC have been recognized in the archives. An MIME type must be determinable from the old document classes in the new interface so that documents already stored can be accessed. The MIME type is derived from the document class.

Document Structure

There are already many documents in the SAP ArchiveLink interface that consist of several components. This is particularly true of documents of class FAX, OTF or ALF. As well as one or more (FAX) data files, these documents sometimes also contain a description file (ALF) and a note file.

During migration a component ID (**compId**) must be assigned to each component. The procedure is as follows.

- The **compId** data is assigned to the data file. If several data files exist (for document class FAX, for example, if there are several pages and each page is saved separately as a TIFF file), they are assigned to the components data1, data2, In this case there is no component data.
- A description file (ALF only) is assigned to the component descr.
- Finally, the note file is assigned to the component note.

Converting from Document Classes to MIME Types

When an archive is migrated, the Mime type of the component data (or data1, data2, ...) is determined on the basis of the document class.

Conversion is shown in the following table. The order of the entries is significant. The first appropriate entry from the top is used for conversion.

| Document Class | MIME Type |
|----------------|--------------------------|
| FAX | image/tiff |
| BIN | application/octet-stream |
| DOC | application/msword |
| PDF | application/pdf |

Information on Migrating Existing Archives

| | |
|------|-------------------------------|
| PS | application/postscript |
| RTF | application/rtf |
| XLS | application/vnd.ms-excel |
| MPP | application/vnd.ms-project |
| PPT | application/vnd.ms-powerpoint |
| ALF | application/x-alf |
| OTF | application/x-otf |
| RAW | application/x-raw |
| REO | application/octet-stream |
| SCR | application/x-scr |
| BMP | image/bmp |
| GIF | image/gif |
| JPG | image/jpeg |
| PCX | image/pcx |
| TIFF | image/tiff |
| TIF | image/tiff |
| HTM | text/html |
| TXT | text/plain |



For document classes not listed above, no MIME type is set.

The components descr (ALF only) and note contain the MIME type as follows:

| compld | MIME Type |
|--------|-------------------------|
| note | application/x-note |
| descr | application/x-alf-descr |

Information on Migrating Existing Archives

SAP ArchiveLink 4.5 Bar Code BAPI

Bar code confirmations are reported to the R/3 System for the Content Server HTTP interface 4.5 via the BAPI **Barcode.SendList**. The called function module **BAPI_BARCODE_SENLIST** has the following interface:

```
BAPI_BARCODE_SENLIST
    exporting
        return like bapiret2
    tables
        barcodetable structure bapibarc
```

The table **barcodetable** has the following structure:

| Field name | Field format | Field length | Description |
|------------|--------------|--------------|---|
| barcode | Character | 40 | Bar code that the external system returns to the R/3 System |
| contrep | Character | 40 | Content-Repository |
| docid | Character | 40 | ID of stored document |
| ardate | Character | 8 | Date on which the document was stored. |
| doctype | Character | 20 | Document class |

The structure **return** transfers error messages from the R/3 Systems to the external system. It has the following structure:

| Field name | Field format | Field length | Description |
|------------|--------------|--------------|----------------|
| type | Character | 1 | Message type |
| ID | Character | 20 | Message ID |
| number | Character | 3 | Message number |
| message | Character | 220 | Message text |

The following error are reported (all error are of type E and from message class OA):

| Error Number | Message text |
|--------------|--|
| 240 | The specified bar code does not exist and could not be deleted |
| 241 | No bar code entry could be made |
| 242 | No bar code exists |
| 243 | No links were transferred that can be entered |

SAP ArchiveLink 4.5 Bar Code BAPI

If an error message is returned in the structure `return`, the table `barcodetable` contains the unsuccessful bar code entries.



For more information on processing bar codes, see the SAP ArchiveLink documentation (for example, in sections [Bar Codes \[Page 82\]](#), [Concept: Storing with Bar Codes \[Page 76\]](#), [Storing with Bar Codes \[Page 131\]](#)).

For information on calling BAPIs, see [BAPI User Manual \[Ext.\]](#).

Communication

Communication includes both protocols and application maintenance.

Using **protocol maintenance**, you can define how certain SAP ArchiveLink functions, such as Display, Archive, and Retrieve, can be handled for individual document classes.

Using **application maintenance**, you can define according to function, the sequence of the calls relocated by the R/3 System to the specified partner application, in order to handle a particular function.

Protocols

Protocols

Protocols are defined in order to control communication according to the document class. For the individual document classes, you can determine how (that is, using which communication protocol) every SAP ArchiveLink function, for example, Display, Archive and Retrieve should be handled.

By assigning

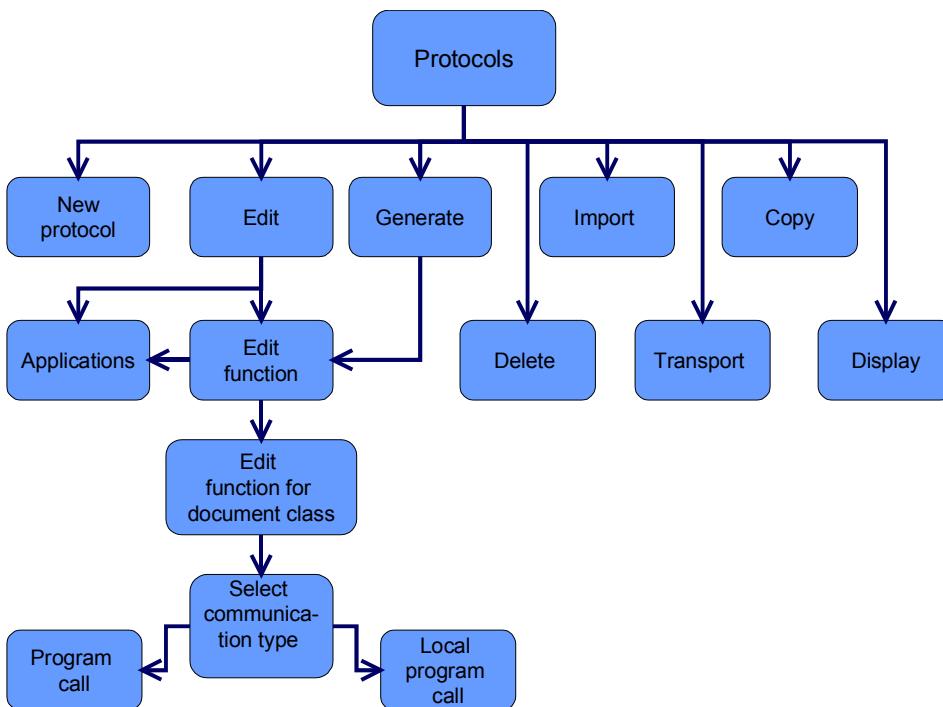
- document classes to document types and
- document types to business objects from R/3 applications.

you establish the relationship between the protocol definition and the business objects.



Protocol maintenance always refers to document classes.

Process Flow



You can maintain protocols in the following ways:

- In the IMG

For this, choose the following in the IMG
“R/3”, “Basis”, “Basis Services”, “SAP ArchiveLink”, “Administration Settings” or “Content server settings”, “Maintain protocols”.
- In the system

Protocols

For this, choose
Tools → *Business documents*,
Basic Settings → *Communication* → *Protocols*

For information on activity maintenance, refer to the online help in the IMG.

Protocols: Concept

The R/3 server communicates with content servers via RPC (Remote Procedure Call) or RFC (Remote Function Call) or HTTP (Hypertext Transfer Protocol) for file access.



As of R/3 Release 4.0A, the function ***Start external application*** is available (see [Start External Application \[Page 316\]](#)).

As of R/3 Release 4.6A, protocol maintenance is simplified: The backend functions are therefore no longer available for protocol maintenance:

- Synchronous Archiving Server
- Synchronous Archiving Table
- Asynchronous Archiving Server
- Synchronous Retrieval Server
- Synchronous Retrieval Table
- Synchronous retrieval byte stream
- Asynchronous Retrieval Server
- Status query
- Deleting a stored document
- Format stored document

Example: Protocol and possible functions

| Example | Interface version 0045 | Example Protocol |
|----------------------------|--|------------------|
| Functions | | |
| Display stored document | | |
| | → Retrieve for frontend → Display local file | |
| Store from frontend | | |
| | → Store file on frontend → Store file from frontend | |
| Close window | | |
| Start external application | | |



As of R/3 Release 4.0A, it is possible for protocols to be generated by the system (see [Generating Protocols \[Page 308\]](#)).

The functions for storing or displaying can:

- be carried out directly using one function (or in one step) (for example, ICC) or
- be carried out using a combination of two functions (for example, a file to be stored can be retrieved via OLE to the frontend and then stored in the content server by the R/3 server via RFC).

The protocols are assigned to a content server in content server maintenance.



As of R/3 Release 4.5A, there is a **connection to the Change & Transport System** for protocols (*Protocol → Transport*).

You can also copy protocol definitions from other systems that can be accessed via RFC (*Protocol → Import*). This enables **remote Customizing** of the protocols.

So that users do not hinder each other during protocol maintenance, there is a **block against multiple access** for the protocols as of Release 4.5A. If, for example, a protocol is already being processed and another user wants to process this protocol, the second user receives an error message informing that the protocol is currently being processed by the first user.

Protocols: Concept

As of R/3 Release 4.6A, there is only one configuration for each protocol. The platform to be used is Windows 32. This means that the frontend functions are no longer frontend-dependent.

Standard communication

The following standard communication is defined for the various functions for Windows 32:

| Function | Standard communication |
|--------------------------|--|
| Display stored document | ICC (for ALF, FAX and OTF), otherwise Archivelnk |
| Retrieval for frontend | Archivelnk |
| Display local file | ICC (for FAX), otherwise Archivelnk |
| Archive from frontend | ICC |
| Store file on frontend | Archivelnk |
| Store file from frontend | Archivelnk |
| Close Window | ICC |



Generally, you must enter the communication type **HTTP** for an HTTP content server.

Generally, you must enter the communication type **Archivelnk** for a file content server.

Elements

The central element of a protocol is protocol maintenance. You can maintain protocols in the following ways:

- In the IMG

For this, choose the following in the IMG
 "R/3", "Basis", "Basis Services", "SAP ArchiveLink", "Administration Settings" or "Content server settings", "[Maintain protocols \[Ext.\]](#)"

- In the system

For this, choose
Tools → *Business documents*,
Basic Settings → *Communication* → *Protocols*

For information on activity maintenance, refer to the IMG online help.

Application Scenarios

The following examples are application scenarios for which protocols must be maintained:

- Storing and displaying PC files (for example in MS Word, MS Excel, MS PowerPoint, etc.)
- Using external entry dialogs (a part of the content server which is not integrated)
- Using external viewers (a part of the content server which is not integrated)

Protocols: Concept

- Using external scan dialogs, if these are called via OLE Automation 2.0
- Using external viewers, if these are called via OLE Automation 2.0.

Non-application scenarios

You do not have to maintain protocols for the following application scenarios:

- You can only store incoming documents by storing with bar codes.
- Using internal scan dialogs which you can call via ICC
- Using internal viewers which you can call via ICC

Generating Protocols



Generating protocols is available as of R/3 Release 4.0A. Currently, only entries for the display of documents are generated.

Purpose

You can create a **standard protocol**. In this case, entries for document classes DOC, PDF, TIF, HTM and XLS are generated. The application registered for the relevant document class is started when you display.

You can also generate **individual entries** for the following applications and the corresponding document classes:

| Application | Document classes |
|---------------------------------------|-----------------------------------|
| MS Word | DOC |
| MS Excel | XLS |
| Wang Imaging for Windows NT | TIF, BMP |
| Graphic Converter for Apple Macintosh | TIF |
| MS Internet Explorer | TIF, DOC, XLS, PDF, HTM, GIF, JPG |

Prerequisites

You must create a protocol before you can start generating protocol entries.

Process Flow

Individual Entries

2. You must make specifications regarding:
 - Application and/or
 - Document class
3. You can then make specifications regarding:
 - Display local document
 - Execute application
 - Via EXECUTE or
 - Via OLE
4. Default values are suggested by the system for the document class and the display function.
5. You can change these default values or have the system make the respective entries via *Generate*.

Result

The system has generated the entries for your protocol in accordance with your individual requirements.

Storing

Storing

The following customizable functions belong to the **storing** area:

- Archive from frontend
- Store file on frontend
- Store file from frontend
- Starting an External Application



This function is available as of R/3 Release 4.0A (see [Starting an External Application \[Page 316\]](#) [\[Page 316\]](#)).

You can store a document generated on the frontend, that is an incoming document, in either **one or two steps**.

- Single-step procedure:

A request to scan and store a document is sent to an content server-internal scan dialog.

- Two-step procedure:

If the program with which the document is scanned cannot store directly, the function is split up into the following two steps:

3. Store file on frontend
4. Store this file from the frontend into the content server



MS Word cannot store in an external content server. Thus you store WinWord files with the following two steps:

3. Archive file on the frontend:

A request is sent to WinWord to save the file that is currently being processed and to report the file name to the R/3 System.

4. Store this file from the frontend into the content server

The file is imported into the R/3 System and transferred to the content server from the R/3 application server, or the file name is transferred via OLE Automation 2.0 to a frontend component of the content server.

Storing Using Internal Entry Dialog

When you store using content server-internal entry dialogs via OLE Automation, you only have to set function **Store from frontend**. The functions *Store file on frontend* and *Display file from frontend* need not be explicitly customized.

Enter **OPEN** as the communication type after selecting the function and the document class. This acts as the generic term for OLE Automation.

When communicating using **OPEN**, you need to define a logical name for the application in question. This name can be assigned freely and is entered here while you maintain the precise communication commands for the relevant application later on in application maintenance. Every time you maintain a function that is to be handled using the communication type **OPEN**, you must enter the application.

Function: Archive from Frontend

When the function *Archive file from frontend* is executed, the R/3 System makes a request to an content server entry dialog to store the current document (entered or processed). An identification unique to the content server, *arch_doc_id*, is returned by the content server entry component to the R/3 System. Using the R/3 System, you can transfer the variable *archiv_id* to the entry component in order to allow the entry component to carry out possible checks if, for example, the *archiv_id* is not suitable for the entry component.

Storing PC Files, and Documents Entered Using External Entry Dialogs

Storing PC Files, and Documents Entered Using External Entry Dialogs

In the two-step storing procedure, you should first define that the function **Archive from frontend** is made up of the following two steps:

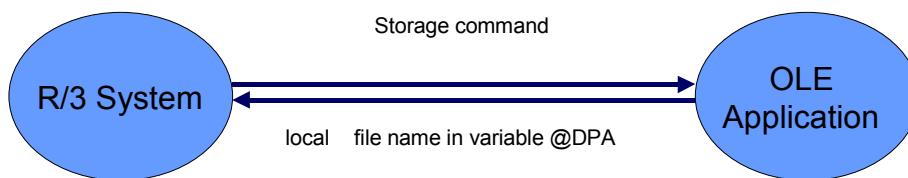
- **Store file on frontend** and
- **Store file from frontend**

In the following steps, you set up the procedure for the functions **Store file on frontend** and **Store file from frontend**.

After selecting the function **Store from frontend** and the document class, enter **ARCHIVELNK** as the communication type. This means that the function **Store from frontend** is split up into the subfunctions **Store file on frontend** and **Store file from frontend**.

Function **Store file on frontend**

This function is used to transfer the name of the file to be stored, to the R/3 System: The R/3 System finds out the name of the file currently being processed from a frontend application (absolute path incl. file name). The frontend application returns the absolute file name of the file currently being edited. Via the R/3 System, it is possible to transfer the variable **archiv_id** to the entry component to allow the entry component to carry out possible checks if, for example, **archiv_id** does not fit in with the entry component.



For the file resulting on the frontend, you must check whether the editing/creating application supports OLE Automation as a server. Examples of such applications are WinWord, MS Excel or internal scan dialogs that were certified for R/3 Release 3.0.

Storing PC Files, and Documents Entered Using External Entry Dialogs

- **OLE application**

After selecting the function *Store file on frontend* and the document class, enter **OPEN** as the communication type.

- **Non-OLE application**

If files that have not been edited/scanned/generated by an OLE-capable application are to be stored, you can enter the file name of the file to be stored in the R/3 System.

After selecting the function *Store file on frontend* and the document class, enter **ARCHIVELINK** as the communication type.



Example of application maintenance for the function *Store file on frontend*

Microsoft Word 7.0 for Office 95 is used here as an example.

The OLE class name of the object used is "Word.basic".

The following calls are executed:

- Call method "TOOLOPTIONSSAVE" with parameters 0, 0, 0.
This deactivates any dialog boxes for the following "FILESAVE".
- Call method "FILESAVE". You call this in order to ensure that the current status of the document is also stored.
- Call method "FILENAME"
This ascertains the name of the current file.
- This would normally suffice because SAP ArchiveLink contains the necessary values. The following method calls are required because it is not always possible for one application to open and read a file which has been opened by another application.
- Call method "FILESAVEAS C:\SAPFILE"
This saves the current edited file under the name C:\SAPFILE.
- Call method "FILENAME"
This ascertains the name of the current file (in this case C:\SAPFILE).
- Call method "FILECLOSE"
This closes the current file (in this case C:\SAPFILE).
- Call method "FILEOPEN"
This call reopens the original file.
- The administration in SAP ArchiveLink is as follows:

Method: **TOOLOPTIONSSAVE 0 ,0 ,0**

Method **FILESAVE**

Method **FILENAME = @FPA**

The name of the current file is ascertained and assigned to the SAP ArchiveLink variables **@FPA** (frontend path)

Method **FILESAVEAS C:\SAPFILE**

Method **FILENAME = @DPA**

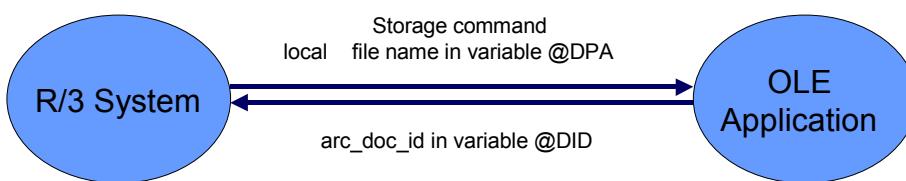
The name of the current file (in this case C:\SAPFILE) is ascertained and assigned to the SAP ArchiveLink variables **@DPA**

Storing PC Files, and Documents Entered Using External Entry Dialogs

Method **FILECLOSE**
 Method **FILEOPEN @FPA**

Function Store file from frontend

This function is used to store the file whose name was transferred to the R/3 System by the function *Store file on frontend*: The R/3 System transfers the absolute file name of the file determined in the function, to the application, requests the application to store and expects a unique ID for the stored document to be returned. Via the R/3 System, it is possible to transfer the variable *archiv_id* to the entry component to allow the entry component to carry out possible checks if, for example, *archiv_id* does not fit in with the entry component.



It is necessary to ask whether the frontend has an internal component that is capable of OLE and to which files can be transferred for storing.

- **OLE-capable components:**

After selecting the function *Store file from frontend* and the document class, enter **OPEN** as the communication type.

- **Non-OLE-capable components:**

The file must be imported into the R/3 System, and stored via the application server. In this case, after selecting the function *Store file from frontend* and the document class, enter **ARCHIVELNK** or **TABLE** as the communication type.

- **ARCHIVELNK**

ARCHIVELNK here means that the file is imported from the frontend and transferred as a file to the content server via server communication.

The setting **ARCHIVELNK** is also supported by systems certified after R/3 Release 2.1.

Storing PC Files, and Documents Entered Using External Entry Dialogs– **TABLE**

TABLE here means that the file is imported from the frontend and transferred as an internal table to the content server via server communication.

The setting **TABLE** requires an content server that was certified after R/3 Release 3.0.

Starting an External Application



This function is available as of R/3 Release 4.0A.

This function is used for starting an external application for archiving from the frontend.

Protocol maintenance is necessary for this function, if

- An external application is to be started,
- Manual input is to be made, and
- An action is to be executed again afterwards.



The Wang viewer can be started via the function *Create links → And store document* within generic object services (see [Storing Documents Generically \[Page 109\]](#) [Page 109]). A document can then be scanned. Then the scanned document can be stored via *Transfer*.

Only the communication type OPEN, that is, communication via OLE, is suitable for protocol maintenance for document class TIF.

To address an external scan dialog, start the Wang viewer as display component and then transfer a scanned document from the Wang viewer, the following should be specified for application maintenance for the function *Start external application*:

Application: WANGIMAGE.APPLICATION

Method: CreateImageViewerObject = IMG

Method: IMG.New

Display

The following customizable functions belong to **Display**:

- Display stored document
 - Retrieval for frontend
 - Display local file
- Close Window

You can display a stored document in **one or two steps**.

- Single-step procedure:
A request to display a document is sent to an internal viewer.
- Two-step procedure:
If the program that is to display the document cannot directly access the content server, the function is split up into the following two steps:
 3. Retrieve the stored document in a file on the frontend
 4. Display this file



MS Word cannot access a content server. Displaying stored documents is therefore carried out in the following two steps:

3. Retrieve the stored document in a file on the frontend:
A request is sent to an content server component, instructing it to retrieve the stored document as a file on the frontend.
4. Display this file:
WinWord is called via OLE, to display the file retrieved.

Displaying Outgoing Documents in PDF Format

The display of outgoing documents in PDF format is a case of the two-step display via EXECUTE (see [Displaying PC Files and Displaying using Archive-External Viewers via EXECUTE \[Page 319\] \[Page 319\]](#)). Adobe Acrobat Reader is used as the viewer.

Displaying PC Files and Displaying Using External Viewers via EXECUTE

Displaying PC Files and Displaying Using External Viewers via EXECUTE

If the function *Display local file* was customized via the setting `ARCHIVELNK` as via an Execute call to a viewer, the program is called with the filename as the first parameter.

You can use the function *Program call* to specify the application to be called (and possibly other programs) in different ways:

- Under Windows 32 bit, if no further details are set, the application relevant to the document class (document class = extension) is ascertained and called by the registry. For example, WinWord is the application for document class DOC (Extension.doc), and is called automatically.
- You can use the function *Local program call* to define which application is called for **your frontend PC**.

For this to be possible, a system variable `host name = xyz` must be maintained for the PC.

Displaying PC Files and Displaying Using External Viewers and OLE Automation

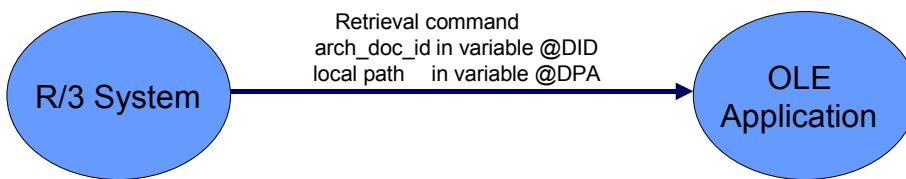
Displaying PC Files and Displaying Using External Viewers and OLE Automation

In the two-step procedure, you must first define that the function *Display stored document* is made up of the two steps *Retrieval for frontend* and *Display local file*. In the following steps, the procedure for the functions *Retrieval for frontend* and *Display local file* is set.

After selecting *Display stored document* and the document class, enter **ARCHIVELNK** as the communication type. This means that the function *Display stored document* is split up into the subfunctions *Retrieval for frontend* and *Display local file*.

Function: Retrieval for Frontend

This function is used to transfer the stored document from the content server to the frontend PC. The R/3 System sends a request to a frontend application to retrieve an stored document, identified by the unique number "ARCH_DOC_ID", as a local file, identified by a transferred file name (including absolute path), on the frontend PC.



Retrieval for frontend can be handled in three different ways:

4. It is still to be checked whether the content server supports this function via OLE Automation. As the OLE functions in the certification process are optional functions, this must be checked on an individual basis with the content server supplier.

The precondition is an content server certified for R/3 Release 3.0.

In this case, after selecting the function *Retrieval for frontend* and the document class, enter **OPEN** as the communication class.

5. An alternative to option 1 is that the content servers provide this function through the implementation of a frontend DLL (Windows). In this case, the function *Retrieval for frontend*

Displaying PC Files and Displaying Using External Viewers and OLE Automation

does not have to be selected for the selected document class (the DLL access is set as a default due to upward compatibility). If you wish to maintain explicitly, you should enter the communication type **ICC** after selecting the function *Retrieval for frontend* and the document class.

6. The third option is to transfer the stored document to the frontend PC via the R/3 application server using sever communication. In this case, after selecting the function *Retrieval for frontend* and the document class, enter **ARCHIVELNK** or **TABLE** as the communication type.

- **ARCHIVELNK**

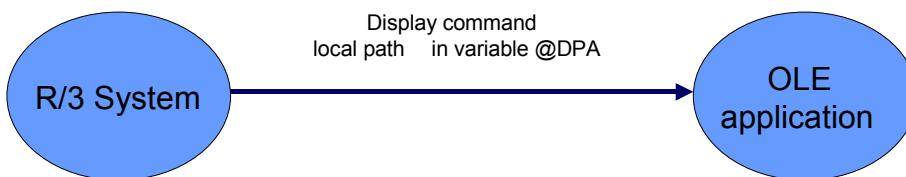
ARCHIVELNK here means that the stored document is transferred from the content server to the application server as a file, imported from the application server and downloaded to the frontend PC.

- **TABLE**

TABLE means that the stored document is transferred to the application server in the form of an internal table and is then transferred to the frontend PC.

Function *Display local file*

This function is used to display the file that is transferred to the frontend PC using the function *Rerieve on frontend*: The R/3 System transfers the request to display a local file, identified by the absolute file name transferred, to a viewer application.



The function *Display local file* can be handled in the following ways:

3. Displaying using an external viewer via OLE Automation

Displaying PC Files and Displaying Using External Viewers and OLE Automation

After selecting *Display local file* and the document class, enter **OPEN** as the communication type.

4. Display via an external viewer using an Execute call (see [Displaying PC Files and Displaying via External Viewers using EXECUTE \[Page 319\] \[Page 319\]](#)).

After selecting *Display local file* and the document class, enter **ARCHIVELINK** as the communication type.



Function *Display local file*

As an example, the Microsoft application Word 7.0 for Office 95 is used to display a document in WinWord format

The OLE class name of the object used is **Word.basic**.

The following calls are executed:

- Call method **FileOpen** with the parameters @DPA to open the local file with the file name @DPA
- Call method **AppShow** to activate the application
- The administration in SAP ArchiveLink is as follows:

Method: **FileOpen** @DPA

Method: **AppShow**

Displaying Using Internal Viewers and OLE Automation

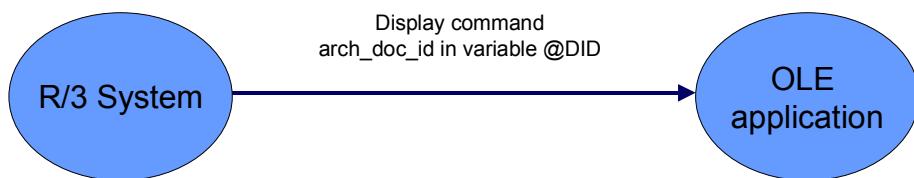
Displaying Using Internal Viewers and OLE Automation

You only have to set the function **Display stored document**. The functions *Retrieval for frontend* and *Display local file* need not be explicitly customized.

After selecting the function and the document class, enter **OPEN** as the communication class.

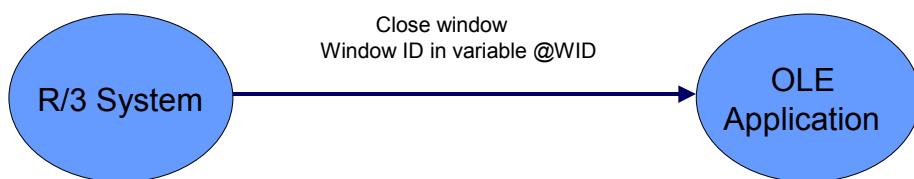
Function “Display stored document”

In this function, the R/3 System sends a request to a viewer application to display a stored document, which is identified by the unique number “ARCH_DOC_ID”.



Close Window

In this function, the R/3 System transfers a request to a display program to close a window that is identified by the unique *Window_Id*.



This function is used in the storage scenario “Storing for subsequent entry”. `OPEN` is used as the setting.

Applications

Applications are used for freely administerable communication with frontend applications via OLE Automation 2.0.

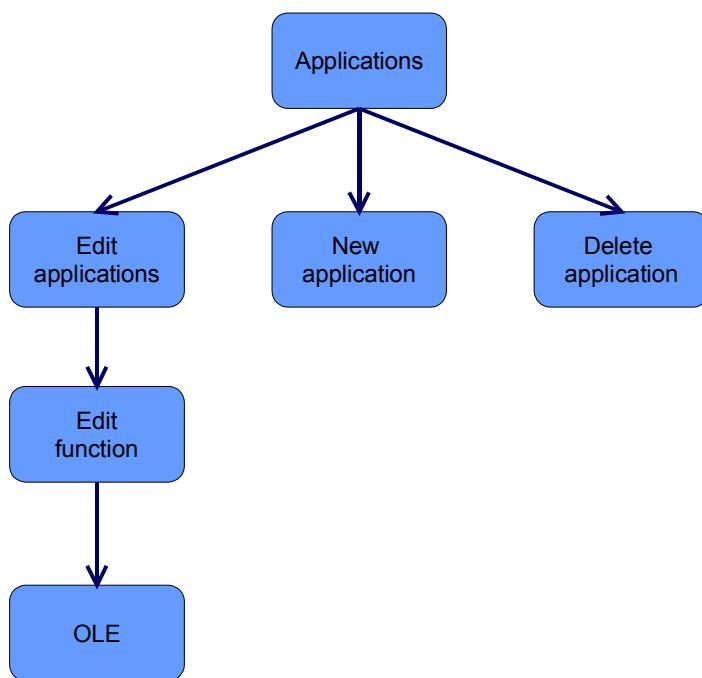
For each function, it is possible to define a sequence of OLE functions, which are processed at runtime. Applications are called via their definition in protocol management.



Application maintenance is only necessary if viewers or entry components are addressed via OLE Automation.

Application maintenance is only required for the functions for which the value "OPEN" was chosen in the protocols.

Process Flow



The syntax that the R/3 System uses to communicate with an application called is predefined by this application.



As of R/3 Release 4.6A, you can only maintain OLE2 applications for frontend functions. AppleScript and OpenDoc are no longer available.

Elements

The central application element is application maintenance. You can maintain applications in the following ways:

Applications

- In the Implementation Guide (IMG)

For this, choose the following in the IMG
“R/3”, “Basis”, “Basis Services”, “SAP ArchiveLink”, “Administration Settings” or “Content server settings”, “[Maintain applications \[Ext.\]](#)”

- In the system

For this, choose
Tools → Business documents,
Basic Settings → Communication → Application maintenance

For information on activity maintenance, refer to the IMG online help.



As of R/3 Release 4.0A, it is possible to determine in application maintenance whether the object generated is to continue to exist or be released again after processing. For further information, refer to the Implementation Guide (IMG) under *Basis - Basis Services - SAP ArchiveLink - Administration Settings (or Content Server Settings) - Maintain applications*.

Syntax: OLE Automation

Separate the **name** of the application and **commands** to communicate with this application. You can split these commands up into method calls, attribute settings and the retrieval of object attributes:

- **Name** of the application
- **Commands** for communication with the application
 - Method calls (see [Calling Object Methods \[Page 329\]](#) [\[Page 329\]](#))
 - Attribute settings (see [Setting Object Attributes \[Page 331\]](#) [\[Page 331\]](#))
 - Retrieval of object attributes (see [Retrieving Object Attributes \[Page 332\]](#) [\[Page 332\]](#))

The application names are described as is the communication syntax in the user documentation for the application to be called. The **application names** are also frequently called **object names**.



Object names:

- Application name for WinWord: **WORD . BASIC**
- Application name for Excel: **EXCEL . APPLICATION**

Transfer of Variables via SAP ArchiveLink

So that you can communicate universally from SAP Archive Link using OLE and Apple Script, is it necessary to transfer **variables/parameters** to OLE objects or to obtain values of OLE objects and to transfer these values to variables in SAP Archive Link. For this purpose, variables have been defined which you can transfer to OLE applications or into which you can transfer values of OLE objects. These parameters are specified in the application administration of the OLE function.



All variables are string variables (no integers).

The following variables are defined

OLE Variables

| Variable | Description |
|----------|--|
| @DPA | File path on frontend, display path |
| @AID | Content server |
| @DTI | Document class |
| @DID | Document ID (arc_doc_id) |
| @SPA | Path on application server, basic path or archive path |
| @WID | Window ID |
| @WIT | Window title |
| @ADA | Storage date |

Syntax: OLE Automation

| | |
|------|---|
| @ATI | Archiving time |
| @STA | Document status (online, offline,...) |
| @COU | Number of retrieved files |
| @DFN | Names of retrieved files |
| @UID | User name |
| @LAN | Logon language |
| @ETX | Error text |
| @DPO | Document path |
| @PAG | Page number |
| @FPA | Frontend path |
| @EID | Error number/return code (0 for okay, string for error) |



OLE parameter @DPA is used most frequently.

Calling Object Methods

There are two syntax forms when you call object methods:

Syntax 1:

```
<ObjectMethod> = <Variable>
```

Syntax 2:

```
<ObjectMethod> <ExportVariable 1>.. <ExportVariable 5>
```

`<ObjectMethod>` is a name defined by the OLE application, whereas `<Variable>`, `<ExportVariable 1>.. <ExportVariable 5>` represent the occurrence of transferred variables. Export variables are used for transmitting values to the OLE object (similar to setting object attributes).

Syntax Variant 1

From a functional point of view, the syntax variant 1 is a sub-case of the syntax variant 2, because it can transfer only one variable. OLE applications do not provide alternative syntax variants, but generally only provide one form.

The syntax variant 1 allows the following functionality:

- Call an object method with an import variable
- Create a new (sub-)object from the current object.



Determine the current file name in Word:

```
filename = @DPA
```

Syntax Variant 2

Syntax variant 2 allows you to call an object method with up to five export variables.



Displaying a local file in Word:

```
fileopen @DPA
```

Sub-objects

You can also call object methods that do not refer to the global object but to a **sub-object** of the application. You should first create this object. You can edit the attributes there after this. The corresponding syntax is then:

Syntax:

```
<Object>.<ObjectMethod> = <Variable>  
or.  
<Object>.<ObjectMethod>  
<ExportVariable 1>.. <ExportVariable 5>
```

Calling Object Methods



Displaying a file in MS Excel 97:

Excel manages an entire tree structure of objects. The root is the object "Excel.Application". Sub-objects are "Sheet", "Workbook", etc. To address an object, you need to create it from the root. Only then can you edit a method or attribute of this object.

| Object name | Excel.Application | Description |
|-------------------|-------------------------------|--|
| Setting attribute | <code>visible = true</code> | Setting the object attribute for the root object |
| Method 1 | <code>WORKBOOKS = mywb</code> | Create the object <code>mywb</code> of type <code>WORKBOOKS</code> |
| Method 2 | <code>mywb.open @DPA</code> | Call the method <code>OPEN</code> for the object <code>mywb</code> |

Variable Exchange

Unnecessary variables (see list of OLE parameters in [Syntax: OLE Automation \[Page 327\]](#) [Page 327]) can be used for the variable exchange between OLE calls.

Setting Object Attributes

Syntax:

```
<ObjectAttribute> = <Value>
```

<ObjectAttribute> is a name defined by the application, whereas **<Value>** represents the occurrence of the variables or the value of the constants transferred via the R/3 System.



To display a spreadsheet in Excel, you need to mark Excel as visible. To do this, the object "Excel.Application" provides the attribute "Visible". To display Excel, the call is as follows:

```
visible = 1
```

You can also edit object attributes that do not refer to the global object but to a sub-object of the application. You should first create this object. You can edit the attributes there after this. The corresponding syntax is then:

Syntax:

```
<Object>. <ObjectAttribute> = <Value>
```

An example can be found in [Calling Object Methods \[Page 329\]](#).

Retrieval of Object Attributes

Retrieval of Object Attributes

Syntax:

```
<Value> = <ObjectAttribute>
```

`<ObjectAttribute>` is a name defined by the application, whereas `<Value>` represents the occurrence of the variables transferred via the R/3 System. The values of these variables are returned to SAP Archive Link.

You can also edit object attributes that do not refer to the global object but to a sub-object of the application. You should first create this object. You can edit the attributes there after this. The corresponding syntax is then:

Syntax:

```
<Value> = <Object>. <ObjectAttribute>
```

An example can be found in [Calling Object Methods \[Page 329\]](#).

Example for Applications

The example below concerns storing MS Word 97 (for MS Office 97) files.

Configuring the function "Display local file"

| Object name | word.basic | Description |
|-------------|----------------------------|----------------------|
| Method 1 | <code>fileopen @DPA</code> | Open file |
| Method 2 | <code>AppShow</code> | Activate application |