

Business Document Service (BC-SRV-BDS)



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

Business Document Service (BC-SRV-BDS)	6
Concepts	8
Parameters	13
Tables	14
Signature Table	15
Component Table	16
File Table	17
URL Table	18
Content Table	19
Relationship Table	20
Attributes Table	21
Query Table	22
Methods	23
General Methods	24
ASSIGN.....	25
CALL_NAVIGATOR.....	26
CHANGE_PROPERTIES.....	27
DELETE	28
GET_FAST_INFO	29
GET_INFO	30
GET_INFO_NEWEST_ONLY	31
QUERY_DOCUMENT.....	32
COPY	33
CREATE_WITH_FILES/URL/TABLE/AS_TABLE	34
CREATE_WITH_OWN_URL	35
UPDATE_WITH_FILES/URL/TABLE/AS_TABLE	36
GET_WITH_FILES/URL/TABLE/AS_TABLE.....	37
CONFIRM_CREATE.....	39
CONFIRM_UPDATE.....	40
Specific Methods	41
Methods for Relationships.....	42
CREATE_RELATION	43
GET_RELATIONS	44
DELETE_RELATIONS.....	45
QUERY_RELATIONS.....	46
Methods for Versions and Variants	47
CREATE_VERSION_WITH_FILES/URL/TABLE/AS_TABLE	48
CREATE_VARIANT_WITH_FILES/URL/TABLE/AS_TABLE	50
Methods for Transport.....	52
GET_TRANSPORT_INFOS	53

DOCUMENT_ENQUEUE	54
DOCUMENT_ENQUEUE	55
Integration of the BDS in SAP Applications	56
Example: CREATE_WITH_FILES	58
Example: GET_WITH_FILES.....	59
Business Document Navigator	60
Displaying Hit Lists	63
Displaying Documents.....	64
Displaying Detailed Information on a Document	65
Storing Documents	66
Storing Documents Using Drag and Drop	68
Create New Documents	69
Storing and Displaying WWW Addresses	71
BDS Note	72
Editing Notes	73
Note Hit List.....	74
Note Display	75
Displaying Keywords	76
Displaying the Version String	77
Copying Documents	78
Deleting Documents.....	79
Editing Documents.....	80
Changing Attributes	81
Creating Annotations.....	82
Refreshing the Hit List.....	83
Navigation to BOR Objects	84
Exporting Documents	85
Transporting Documents.....	86

Business Document Service (BC-SRV-BDS)

Purpose

The **Business Document Service (BDS)** provides general and easily integrated document management functions for R/3 applications. The [Business Document Navigator \[Page 60\]](#) (**BDN**) navigation interface is the central component of these functions. This user interface enables users to perform document management functions such as displaying, creating and deleting documents. Users can interactively execute these functions.

The BDS is also closely related to the business processes in an enterprise, which is particularly apparent in general object relationships. Together with SAP Desktop Office Integration, the BDS is a *solution package* for integrating general desktop office applications (see also [Desktop Office Integration \[Ext.\]](#)).

Implementation Considerations

You do not need to customize when you implement the BDS.

Customizing is only possible for special requirements regarding assigning attributes to documents, specifying as client-specific or cross-client, transport connections and defining a repository other than the standard. (See also [Content Models \[Ext.\]](#).)

For more information on BDS Customizing, see the Implementation Guide (IMG) under *Basis - Basis Services - Business Document Service*.

Integration

The BDS can be integrated into all SAP applications that involve context documents and do not have any special requirements regarding document management. For an example application, choose *Office* → *Business Documents* → *Documents* → *Find*.

Class `CL_BDS_DOCUMENT_SET` is available within the BDS. For more information on integrating the methods belonging to this class, see [Integrating the BDS in SAP Applications \[Page 56\]](#).

Features

The BDS has the following features:

- Simple handling of desktop office products
- Support of scenarios for
 - Authoring
 - Publishing
 - Retrieval
- Storing of private and shared documents
- User interface for
 - Attribute assignment
 - Classification
 - Detailed information and document information

- Displaying
- Changing
- Navigation
- Connection to the Change & Transport System

Concepts

Concepts

Introduction

The Business Document Service (BDS) manages all the documents belonging to an application object. Application objects are objects in a BDS client application that is using the BDS to manage documents for this type of object.

Application objects are identified by their class name and the corresponding class type, as well as the object ID (see also [Parameters \[Page 13\]](#)). Individual documents are identified by their document ID, their version number and their variant number.

The link between the application object and documents is managed in the BDS. Document attributes and relationships, and the content of documents are managed in the [SAP Knowledge Provider \[Ext.\]](#) in special data structures - so-called *content models* (see also [Content Models \[Ext.\]](#)). The BDS can evaluate information from various content models, provided that they conform to a basic schema. This means that the individual client applications can use content models that differ according to delivery class, whether or not they are client-specific and available document attributes.

The document meta data are managed in the R/3 database. The document content can also be managed in the R/3 database or it can be transported to a content server via HTTP. It is also possible to move the document content from one physical storage medium to another at a later stage.

Document Model

Documents have various functions: They may be paper documents in the classical sense of the word, but may also be document templates, macros or notes, among other things. In addition, documents may be related to other documents. For example, a note created for a document has a particular relationship to the document itself. In the BDS document model therefore, documents are characterized by attributes and relationships. Attributes relate to the document type ID, while relationships are represented as "belongs-to" relations between documents.

In connection with the BDS document management system, a document is considered within the framework of its whole life cycle. For this reason, the sequence of content, language and format versions comes into consideration when defining a document. Context information and relationships with other objects and documents are also involved.

Business Document Set

The business document set is a central component of the BDS document model. A business document set contains all documents belonging to an application object. The set may contain just one individual document or several documents and their versions and variants. The connection of a documents to an application object can be brought about by a direct relationship, or indirectly by relationships within the document (for example, where there are content or language variants).

Individual documents can be addressed directly. Either the entire document including versions and variants or specific content versions or specific variants are addressed.

Isolated Documents

An isolated document is one single document that exists in one single version and one single variant. In this case, there are no relationships at all between this document and other documents. An isolated document can be addressed using the following parameter combination:

- DOC_ID
Document ID
- DOC_VER_NO
Document content version ID
- DOC_VAR_ID
Document variant ID

Versioning

The content of documents can take various forms. These forms are called content versions or versions. Each 'predecessor' version has **one** 'successor' version, which in turn has **one** successor version and so on. This BDS linear content versioning is characterized as follows:

- Each new version is appended to the linear chain of versions as the latest version.
- Versions are listed.
- Every version can be updated and/or deleted.

Variants

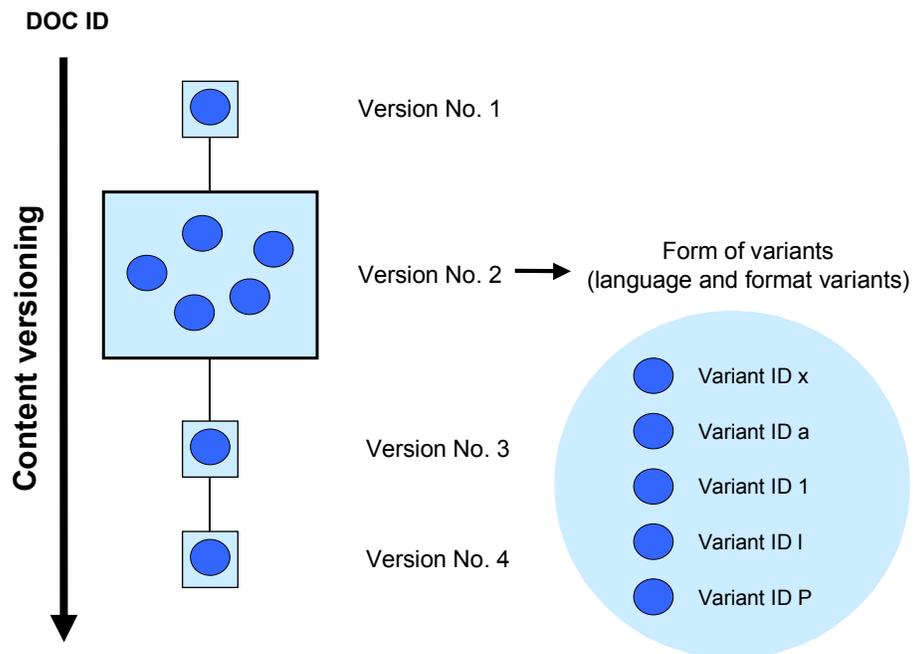
As well as existing in various content forms, documents/document content can be in various languages or formats. For example, a document may be a text editing file or an HTML file, in English or in French. In the BDS environment, these specific version types are called language and format variants or simply variants.

The following variant types are supported:

- LA
Language variant
- FO
Format variant
- OR
Original

The variant principle is shown in the diagram below:

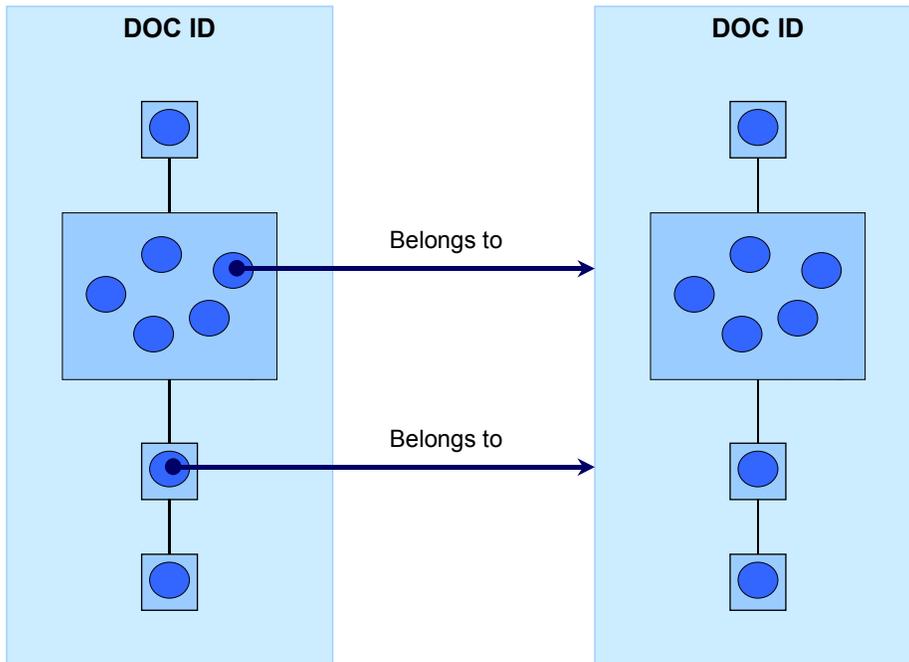
Concepts



- All variants are of equal status, that is, the original document is not marked especially.
- The set of variants is not structured.
- Variants are numbered.
- New versions relate to the set of variants as a whole.
- Every variant can be updated and/or deleted.

Relationships

Between physical documents, there can be relationships to logical documents. These relationships are expressed by the belongs-to relationship type:



Methods

The central BDS class CL_BDS_DOCUMENT_SET provides three method sets for checking documents from various sources in and out:

- Access via URL
- Access via file name
- Access via internal table

For more information, see the sections under [Methods \[Page 23\]](#).

Standard Attributes

The following standard attributes are available in the BDS:

Attribute Name	Meaning
BDS_ARCHIVED_AT	Archiving date
BDS_CONTREP	Content repository
BDS_DOCID	Document ID
BDS_DOCUMENTCLASS	Document class
BDS_DOCUMENTTYPE	Document type
BDS_KEYWORD	Keyword
BDS_STATE	BDS status of application
BDS_NOTE	Type of note

Concepts

BDS_VAR_ID	Variant ID of document
BDS_VAR_TAG	Variant tag of document
BDS_VERSION_NO	Version number of document



You cannot assign values to the attributes **BDS_ARCHIVED_AT**, **BDS_CONTRREP** and **BDS_DOCID**.

The attributes **BDS_DOCUMENTTYPE** and **BDS_DOCUMENTCLASS** must be used in accordance with the SAP ArchiveLink conventions for document types (**BDS_DOCUMENTTYPE**) and document classes (**BDS_DOCUMENTCLASS**). For more information, see the Implementation Guide (IMG) under *Basis* → *Basis Services* → *SAP ArchiveLink* → *System Settings*, sections [Maintaining Document Classes \[Ext.\]](#) and [Maintaining document Types \[Ext.\]](#).

The attributes **BDS_KEYWORD** and **BDS_STATE** can be used without restriction.



The following standard attributes of the SAP Knowledge Provider are also available, but they cannot be modified:

Attribute Name	Meaning
CHECKED_OUT	Flag for checked-out documents
CHECKOUT_USER	User
CREATED_AT	Creation time
CREATED_BY	Creator
DESCRIPTION	Short description
DOCUMENT_PROTECTION	Document protection
LANGUAGE	Language
LAST_CHANGED_AT	Time of change
LAST_CHANGED_BY	Last person to make changes
STATE	Status
STORAGE_CATEGORY	Content category

Parameters

Application objects are identified by the following parameters:

- LOGICAL_SYSTEM
 - Logical system, ID of R/3 System instance
 - For more information, see the Implementation Guide (IMG) under *Basis* → *Distribution (ALE)* → *Prepare sender system and recipient system* in the activities under *Setting up logical systems* ([Specifying a Logical System \[Ext.\]](#), [Assigning a Logical System to a Client \[Ext.\]](#), [Converting a Logical System in Application Tables \[Ext.\]](#)).
- CLASSNAME
 - Class name, name of application class
- CLASSTYPE
 - Class type, type of application class
 - A class type may be an object type from the Business Object Repository.
- OBJECT_KEY
 - Object ID
- CLIENT
 - Client
 - This parameter is optional, except in the case of client-specific documents, where it must be used.

The application object is identified and the relevant document set determined, by a query for all documents assigned to this application object.

An individual document is specified by the following parameters:

- DOC_ID
 - Document ID
- DOC_VER_NO
 - Document version number
- DOC_VAR_ID
 - Document variant ID

Tables

Tables

Signature Table

The document(s) is/are identified via the signature table, that is, by the parameter **SIGNATURE**. This table is used to perform both the query and the response: If you specify what you know about the document or documents concerned, a hit list of the documents that correspond to the criteria is displayed. Attributes of one or more documents can be transferred in the signature table. The signature table has the following fields:

Field	Description
DOC_COUNT	Document counter This field counts individual document variants. There may be several entries per variant. If so, the individual entries differ only in the fields attribute name and attribute value.
DOC_ID	Document identification
DOC_VER_NO	Document version number
DOC_VAR_ID	Document variant ID
DOC_VAR_TG	Document variant tag The type of variant is specified. A variant can take one of three forms: <ul style="list-style-type: none"> • Original • Language variant • Format variant
COMP_COUNT	Component counter, total number (sum) of document components
PROP_NAME	Attribute name
PROP_VALUE	Attribute value



The more fields you specify, the fewer hits will be found. This gives you the option of restricting your search to just one variant, for example.

Component Table

Component Table

The table **COMPONENTS** contains components of a document set. The components of several documents can be transferred:

Field	Description
DOC_COUNT	Document counter This field counts individual document variants. There may be several entries per variant. If so, the individual entries differ only in the fields attribute name and attribute value.
COMP_COUNT	Component counter for a document, numbering. The counter increases by one for each component.
COMP_ID	Component ID
MIMETYPE	MIME type of component
COMP_SIZE	Size of component

File Table

The table **FILES** is used for handling document contents with the aid of local files. The following fields are specified in this table:

Field	Description
DOC_COUNT	Document counter This field counts individual document variants. There may be several entries per variant. If so, the individual entries differ only in the fields attribute name and attribute value.
COMP_COUNT	Component counter for a document The counter shows the sum of the components of the document.
DIRECTORY	Name of source directory
FILENAME	File name
MIMETYPE	MIME type of component



Enter a backslash, if the root directory of a partition is to be addressed when the field **DIRECTORY** is transferred. For example, c: \

URL Table

URL Table

In the URL table, document content is transferred via URLs, that is, complete URLs are transferred. This specifies where the documents can be stored. The URL table has the following fields:

Field	Description
DOC_COUNT	Document counter This field counts individual document variants. There may be several entries per variant. If so, the individual entries differ only in the fields attribute name and attribute value.
COMP_COUNT	Component counter for a document The counter shows the sum of the components of the document.
URI_COUNT	URL counter The number of fields is entered in this field, depending on <code>URI</code> . The individual <code>URI</code> parts are numbered.
URI	URL This field has a restricted length of 4096 characters. If the URL does not fit into this field, it is split and continued in another field. This causes an increase (of one) in the <code>URI_COUNT</code> counter.

Content Table

In the table **CONTENT** (the binary table), document contents are transferred directly. Transfer is via an internal table. The relevant field is **LINE**:

Field	Data Element	Data Type	Length	Description
LINE	SDOK_SDATX	RAW	1022	Line for binary document content, length for upload

If a caller transfers the content of a document in the binary table and at the same time defines that the mime type is `text/subtyp`, the document is handled as a text document, that is, it is converted into the current R/3 code page and stored as text.

Several components of various documents can be transferred in a content table: A new row is started for each component. The size of the individual components (table [COMPONENTS \[Page 16\]](#)) indicates the beginning and end (row x) in the content table.

In the table **ASCII_CONTENT** (the ASCII table), document contents are transferred directly. Transfer is via an internal table. The relevant field is **LINE**:

Field	Data Element	Data Type	Length	Description
LINE	SDOK_SDAT	CHAR	1022	Line for text document content, length as for upload

Several components of various documents can be transferred in a content table. A new row is started for each component. The size of the individual components (table **COMPONENTS**) indicates the beginning and end (row x) in the content table.

In the Content table, one text line is transferred per table line. Line breaks are recognized implicitly by the character `≠ Space` after the last character, that is, there are no explicit line breaks. Continuous text can be defined for applications with very long text lines. This can be implemented using the statement `text_as_stream = 'X'`: *Carriage Return* and *Line Feed* are in the text implicitly. Therefore, line breaks can be specified explicitly.

Relationship Table

Relationship Table

In the table **RELATIONS**, relationships between documents are specified. The following attributes are provided:

Field	Description
DOC_COUNT	Document counter This field counts individual document variants. There may be several entries per variant. If so, the individual entries differ only in the fields attribute name and attribute value.
RELA_ID	Relationship ID

Attributes Table

In the table `PROPERTIES` both attributes are transferred:

Field	Description
PROP_NAME	Attribute name
PROP_VALUE	Attribute value

This table is used for methods that only address one document.

Query Table

Query Table

The selection criteria for the search query are specified in the table `QUERY`.

Each entry returns restrictions for an attribute, analogous to the structure of a ranges table. The following attributes are expected:

Field	Description
PROP_NAME	Attribute of a document or description
SIGN	Include-Exclude attribute in a query
OPTION	Comparison operator in a query
LOW	Lower interval limit in a search query
HIGH	Upper interval limit in a search query

Methods

General Methods

General Methods

ASSIGN

This method is used to assign a document to another application object. The ID of the application object and the document ID of the document that is to be assigned are transferred.

CALL_NAVIGATOR

CALL_NAVIGATOR

The application object (Document Set) is identified by the attributes described in the section [Parameters \[Page 13\]](#).

Documents are identified using the [signature table \[Page 15\]](#).

In the table **EXCLUDING** you can specify which functions are not offered. The following values are defined for **EXCLUDING**:

Function	Meaning
NO_MODIFY	No modifying of document
NO_DELETE	No deleting of document
NO_TRANSPORT	No transporting of document
NO_CREATE	No creating documents
NO_STANDARD_DOCS	No standard document types
NO_MODIFY_ATTRIS	No modifying attributes/keywords
NO_COPY	No copying documents
NO_USER_OPTIONS	No user-specific settings
NO_KEYWORDS	No keywords for documents
NO_DOCINFO	No document information
NO_DISPLAY	No displaying of documents
NO_EXPORT	No exporting of documents to frontend
NO_NOTE	No note functions
NO_EAI_TOOLS	No annotations
NO_EAI_FILE_DROP	No <i>Drag and Drop</i> in the EAI viewer
NO_OBJ_DISPLAY	No displaying of BOR objects
NO_DRAG_DROP	No <i>Drag and Drop</i> for documents



For example, you may wish it to be possible only to display documents in the BDN, in which case you would explicitly exclude the modifying and deleting functions in the table **EXCLUDING**.

CHANGE_PROPERTIES

This method is used to change or create document attributes.

The application object (Document Set) is identified by the attributes described in the section [Parameters \[Page 13\]](#). Individual documents are specified by the attributes that are also described in the section Parameters. The table [PROPERTIES \[Page 21\]](#) is used for changing attributes.

DELETE**DELETE**

This method is used to delete documents.

The application object is identified (Document Set) by the attributes described in the section [Parameters \[Page 13\]](#).

The document to be deleted is specified in the [signature table \[Page 15\]](#). Since the object ID is optional, all documents can be deleted.

GET_FAST_INFO

This method does not return documents but rather information about existing documents. This information is stored in the [signature table \[Page 15\]](#). The signature table is used to store general information and attributes. Whereas the component table contains technical information about the individual components that the documents consist of.

The application object (Document Set) is identified by the attributes described in the section [Parameters \[Page 13\]](#).

In contrast to the method [GET_INFO \[Page 30\]](#), the method `GET_FAST_INFO` does not return information about the components of the document nor does it return detailed information about all the individual documents.

Depending on the value of the parameter `GET_PROPERTIES`, some document attributes are returned (`GET_PROPERTIES = 'X'`) or no document attributes are returned.

GET_INFO**GET_INFO**

This method does not retrieve documents, but information about existing documents. This information is stored in the [signature table \[Page 15\]](#) and [component table \[Page 16\]](#). General information and attributes are stored in the signature table. The component table contains technical information on the individual components, which make up the documents.

The application object is identified (Document Set) by the attributes described in the section [Parameters \[Page 13\]](#).

GET_INFO_NEWEST_ONLY

This method retrieves information about the newest version of a document only. Other than this, it is analogous to the method [GET_INFO \[Page 30\]](#).

QUERY_DOCUMENT**QUERY_DOCUMENT**

This method queries the attributes of documents.

The selection conditions are specified in the table [QUERY \[Page 22\]](#). Each entry returns restrictions for an attribute, analogous to the structure of a ranges table.

COPY

This method is used to copy a document.

The application object is identified (Document Set) by the attributes described in the section [Parameters \[Page 13\]](#). The content version is identified by the attributes likewise described in the section "Parameters".

When a document is copied, the link is also copied. The following attributes are transfer parameters:

- **NEW_CLASSNAME**
Class name, name of application class
- **NEW_CLASSTYPE**
Class type, type of application class
- **NEW_OBJKEY**
Object ID
- **NEW_CLIENT**
Client
This parameter is optional, except in the case of client-specific documents, where it must be used.

The attribute name and value are transferred in the table [PROPERTIES \[Page 21\]](#).

The new document is appended to either the outgoing business object or a new document. The new variant is identified by the following attributes (in the same way as the content version):

- **NEW_DOC_ID**
Document ID
- **NEW_DOC_VER_NO**
Document version number
- **NEW_DOC_VAR_ID**
Document variant ID

This means that this method can be used for both copying and reassigning.

CREATE_WITH_FILES/URL/TABLE/AS_TABLE

CREATE_WITH_FILES/URL/TABLE/AS_TABLE

The following four methods are available for creating documents:

- **CREATE_WITH_FILES**
- **CREATE_WITH_TABLE**
- **CREATE_WITH_AS_TABLE**
- **CREATE_WITH_URL**

The difference between these methods is the source from which the data is retrieved. The source can either be a file name, a table or a URL.

The application object (Document Set) is identified by the attributes described in the section [Parameters \[Page 13\]](#).

The method **CREATE_WITH_FILES** is identified by the following additional parameters:

- [FILES \[Page 17\]](#)
- [SIGNATURE \[Page 15\]](#)

The method **CREATE_WITH_TABLE** is identified by the following additional parameters:

- [COMPONENTS \[Page 16\]](#)
- [CONTENT \[Page 19\]](#)
- [SIGNATURE \[Page 15\]](#)

The method **CREATE_WITH_AS_TABLE** is identified by the following additional parameters:

- [COMPONENTS \[Page 16\]](#)
- **ASCII_CONTENT**
- [SIGNATURE \[Page 15\]](#)

The method **CREATE_WITH_URL** is identified by the following additional parameters:

- [COMPONENTS \[Page 16\]](#)
- [URIS \[Page 18\]](#)
- [SIGNATURE \[Page 15\]](#)



CREATE_WITH_URL must always be called in conjunction with [CONFIRM_CREATE \[Page 39\]](#) in order for a document to be stored successfully.

CREATE_WITH_URL only transfers a target URL. **CONFIRM_CREATE** passes on the information that a document has been created successfully.

CREATE_WITH_OWN_URL

This method is a slight variation of the method [CREATE_WITH_URL \[Page 34\]](#). In this case, the source URL is not generated but it can be specified externally. This presupposes that the content of the document can be reached by this URL.

The source URL is transferred by the parameter **URL**.

The MIME type of the document must be transferred in the parameter **MIMETYPE**.

It is specified in the parameter **BINARY_FLAG** whether the content of the document is binary (**BINARY_FLAG** = 'X').



It is not necessary to call the method [CONFIRM_CREATE \[Page 39\]](#).

UPDATE_WITH_FILES/URL/TABLE/AS_TABLE**UPDATE_WITH_FILES/URL/TABLE/AS_TABLE**

The following four methods are available for changing documents:

- UPDATE_WITH_FILES
- UPDATE_WITH_TABLE
- UPDATE_WITH_AS_TABLE
- UPDATE_WITH_URL

The difference between these methods is the source from which the data is retrieved. The source can either be a file name, a table or a URL. If it is a table, it can either be an ASCII table or a binary table.

The application object (Document Set) and the document are identified by the attributes described in the section [Parameters \[Page 13\]](#).

The method **UPDATE_WITH_FILES** is identified by the following additional parameters:

- [FILES \[Page 17\]](#)
- [SIGNATURE \[Page 15\]](#)

The method **UPDATE_WITH_TABLE** is identified by the following additional parameters:

- [COMPONENTS \[Page 16\]](#)
- [CONTENT \[Page 19\]](#)
- **SIGNATURE**

The method **UPDATE_WITH_AS_TABLE** is identified by the following additional parameters:

- **COMPONENTS**
- **ASCII_CONTENT**
- **SIGNATURE**

The method **UPDATE_WITH_URL** is identified by the following additional parameters:

- **COMPONENTS**
- [URIS \[Page 18\]](#)
- **SIGNATURE**



UPDATE_WITH_URL must always be called in conjunction with [CONFIRM_CREATE \[Page 40\]](#) in order for a document to be identified as successfully changed.

UPDATE_WITH_URL only transfers the target URL. **CONFIRM_UPDATE** passes on the information that a document has been changed successfully.

GET_WITH_FILES/URL/TABLE/AS_TABLE

These methods are used to retrieve a document. Four different methods are available:

- **GET_WITH_FILES**
- **GET_WITH_TABLE**
- **GET_WITH_AS_TABLE**
- **GET_WITH_URL**

The difference between these methods is the source from which the data is retrieved. The source can either be a file name, a table or a URL.

The application object (Document Set) is identified by the attributes described in the section [Parameters \[Page 13\]](#).

The method **GET_WITH_FILES** is identified by the following additional parameters:

- [FILES \[Page 17\]](#)

The target directory is transferred in the table **FILES**.
- [SIGNATURE \[Page 15\]](#)

This table is used to limit the number of hits when the method is called and afterwards for information.

The method **GET_WITH_TABLE** is identified by the following additional parameters:

- [COMPONENTS \[Page 16\]](#)
- [CONTENT \[Page 19\]](#)
- **SIGNATURE**

The method **GET_WITH_AS_TABLE** is identified by the following additional parameters:

- **COMPONENTS**
- **ASCII_CONTENT**
- **SIGNATURE**

The method **GET_WITH_URL** is identified by the following additional parameters:

- **SIGNATURE**
- **COMPONENTS**

The entire contents are available in this table.
- [URIS \[Page 18\]](#)

This table only specifies the URL that points to the document.
- **URL_LIFETIME**

Time period for validity, lifetime. This parameter can have two different meanings:

GET_WITH_FILES/URL/TABLE/AS_TABLE

- a. This parameter is used only for R/3 System URLs. It specifies the lifetime of the URL and the URL that this URL points to. If the content of the document is in the R/3 database, **URL_LIFETIME** is used to control the lifetime of the table, on the client.

For more information, see the documentation for the function module **DP_CREATE_URL**, in particular the documentation for the parameter **LIFETIME**.

- b. If **GET_WITH_URL** returns a general URL on an external content server, the lifetime results from the validity of the URL, which is a maximum of two hours. The prerequisite for this is the use of a digital signature.

CONFIRM_CREATE

This method is always called in conjunction with `CREATE_WITH_URL`, `CREATE_VARIANT_WITH_URL` and `CREATE_VERSION_WITH_URL`.

This method must always be called when (a) document(s) has/have been stored successfully via URL. Otherwise the administrative data must be deleted using `DELETE`.

The application object is identified (Document Set) by the attributes described in the section [Parameters \[Page 13\]](#).

This method is specified by the following parameters:

- [COMPONENTS \[Page 16\]](#)
- [URIS \[Page 18\]](#)
- [SIGNATURE \[Page 15\]](#)

CONFIRM_UPDATE**CONFIRM_UPDATE**

This method is always called in conjunction with **UPDATE_WITH_URL**.

This method must always be called when (a) document(s) has/have been stored successfully via URL. Otherwise the administrative data must be deleted using **DELETE**.

The application object is identified (Document Set) by the attributes described in the section [Parameters \[Page 13\]](#).

This method is specified by the following parameters:

- [COMPONENTS \[Page 16\]](#)
- [URIS \[Page 18\]](#)
- [SIGNATURE \[Page 15\]](#)

This method is analogous to [CONFIRM_CREATE \[Page 39\]](#), with the difference that it concerns updating and not creating documents.

Specific Methods

Methods for Relationships

Methods for Relationships

CREATE_RELATION

This method is used to create relationships for an application object.

The outgoing document is identified by the attributes described in the section [Parameters \[Page 13\]](#). The relationships specified in the table [SIGNATURE \[Page 15\]](#) are generated for this document and returned using the table [RELATION \[Page 20\]](#).

GET_RELATIONS**GET_RELATIONS**

This method makes the relationships belonging to a document available along with the corresponding partners.

The outgoing document is identified by the attributes described in the section [Parameters \[Page 13\]](#). The table [RELATION \[Page 20\]](#) contains the relationships existing for the document. The partner documents are specified in the table [SIGNATURE \[Page 15\]](#).

DELETE_RELATIONS

This method is used for deleting the relationships that exist for a document.

The application object (Document Set) is specified by the attributes described in the section [Parameters \[Page 13\]](#). The relationships that exist for this document, which are specified in the table [RELATION \[Page 20\]](#), are explicitly deleted.

QUERY_RELATIONS

QUERY_RELATIONS

This method queries attributes of a relationship as well as attributes of the documents to which a relationship is to exist.

Besides the identification of the application object and the document from which the relationships to be found are to start, the method also has two other transfer parameters:

- For the attributes of the documents to be found (parameter **SIGNATURE**) and
- For the transfer of the search area of the relationship attributes to be found (parameter **QUERY**).

The selection criteria are specified in the table [QUERY \[Page 22\]](#). Each entry returns restrictions for an attribute, analogous to the structure of a ranges table.

Methods for Versions and Variants

CREATE_VERSION_WITH_FILES/URL/TABLE/AS_TABLE**CREATE_VERSION_WITH_FILES/URL/TABLE/AS_TABLE**

The following four methods are available for creating content versions:

- **CREATE_VERSION_WITH_FILES**
- **CREATE_VERSION_WITH_TABLE**
- **CREATE_VERSION_WITH_AS_TABLE**
- **CREATE_VERSION_WITH_URL**

The difference between these methods is the source from which the data is retrieved - as is the case with **CREATE_WITH_FILES**, **CREATE_WITH_TABLE**, **CREATE_WITH_AS_TABLE** and **CREATE_WITH_URL**. The source can either be a file name, a table or a URL.

The application object (Document Set) is identified by the attributes described in the section [Parameters \[Page 13\]](#). The version is identified by the attributes, which are also described in the section Parameters.

The new variant is identified by the parameter **NEW_DOC_VER_NO**, that is, by the new version number.

The parameter **PROPERTIES** is used instead of the parameter **SIGNATURE**.

The method **CREATE_VERSION_WITH_FILES** is identified by the additional parameter **FILES** (see section [File Table \[Page 17\]](#)).

The method **CREATE_VERSION_WITH_TABLE** is identified by the following additional parameters (see sections [Component Table \[Page 16\]](#) and [Content Table \[Page 19\]](#)):

- **COMPONENTS**
- **CONTENT**

The method **CREATE_VERSION_WITH_AS_TABLE** is identified by the following additional parameters (see sections [Component Table \[Page 16\]](#) and [Content Table \[Page 19\]](#)):

- **COMPONENTS**
- **ASCII_CONTENT**

The method **CREATE_VERSION_WITH_URL** is identified by the following additional parameters (see sections [Component Table \[Page 16\]](#) and [URL Table \[Page 18\]](#)):

- **COMPONENTS**
- **URIS**



UPDATE_WITH_URL must always be called in conjunction with [CONFIRM_CREATE \[Page 39\]](#) in order for a document to be identified as successfully stored.

CREATE_VERSION_WITH_URL only transfers a target URL. **CONFIRM_CREATE** passes on the information that a document has been created successfully.

CREATE_VARIANT_WITH_FILES/URL/TABLE/AS_TABLE

CREATE_VARIANT_WITH_FILES/URL/TABLE/AS_TABLE

The following four methods are available for creating language and format variants:

- **CREATE_VARIANT_WITH_FILES**
- **CREATE_VARIANT_WITH_TABLE**
- **CREATE_VARIANT_WITH_AS_TABLE**
- **CREATE_VARIANT_WITH_URL**

The difference between these methods is the source from which the data is retrieved - as is the case with **CREATE_WITH_FILES**, **CREATE_WITH_TABLE**, **CREATE_WITH_AS_TABLE** and **CREATE_WITH_URL**. The source may be a file name, a table or a URL.

The application object (Document Set) is identified by the attributes described in the section [Parameters \[Page 13\]](#). The variant is identified by the attributes that are also described in the section Parameters.

The new variant is identified by the following parameters:

- **NEW_DOC_VAR_ID**
New variant ID
- **NEW_DOC_VAR_TG**
New variant tag

The variant tag can take the following forms:

- Original
- Language variant
- Format variant

One of these forms of variant must be transferred when this method is called. The new variant ID is returned.

The parameter **PROPERTIES** is used instead of the parameter **SIGNATURE** (see section [Properties table \[Page 21\]](#)).

The method **CREATE_VARIANT_WITH_FILES** is identified by the additional parameter **FILES** (see section [File Table \[Page 17\]](#)).

The method **CREATE_VARIANT_WITH_TABLE** is identified by the following additional parameters (see sections [Component Table \[Page 16\]](#) and [Content Table \[Page 19\]](#)):

- **COMPONENTS**
- **CONTENT**

The method **CREATE_VARIANT_WITH_AS_TABLE** is identified by the following additional parameters (see sections [Component Table \[Page 16\]](#) and [Content Table \[Page 19\]](#)):

- **COMPONENTS**

CREATE_VARIANT_WITH_FILES/URL/TABLE/AS_TABLE

- **ASCII_CONTENT**

The method **CREATE_VARIANT_WITH_URL** is identified by the following additional parameters (see sections [Component Table \[Page 16\]](#) and [URL Table \[Page 18\]](#)):

- **COMPONENTS**
- **URIS**



CREATE_VARIANT_WITH_URL must always be called in conjunction with [CONFIRM_CREATE \[Page 39\]](#) in order for a document to be identified as successfully stored.

CREATE_VARIANT_WITH_URL only transfers a target URL. **CONFIRM_CREATE** passes on the information that a document has been created successfully.

Methods for Transport

Methods for Transport

GET_TRANSPORT_INFOS

This method is used to provide the necessary entries for transport.

The application object is identified (Document Set) by the attributes described in the section parameters. The document(s) is/are identified via the signature table, that is, by the parameter **SIGNATURE** (see also [Tables \[Page 15\]](#)).

The parameter **COMMFIL****_ENTRIES** provides the necessary entries for the transport, that is, the individual object entries for requests and tasks in the Workbench Organizer.

DOCUMENT_ENQUEUE**DOCUMENT_ENQUEUE**

This method locks a document for other users. Only a logical lock can be set, that is, this lock only has an effect on the logical level.

It is still possible to use the other methods. The method `DOCUMENT_ENQUEUE` should always be used if there is a possibility that more than one person may modify a document at the same time.

If an error occurs when this method is called, for instance because the document is already blocked by another user, modifying methods should not be used because otherwise data consistency cannot be guaranteed.

After the modifications have been made, the method [DOCUMENT_DEQUEUE \[Page 55\]](#) should be called. This method unlocks the document again.

DOCUMENT_ENQUEUE

This method unlocks a document that was locked using the method [DOCUMENT_ENQUEUE](#) [\[Page 54\]](#).

Integration of the BDS in SAP Applications

Purpose

You integrate the BDS into your SAP application if you wish to use the BDS functions. For more information, see [Business Document Service \[Ext.\]](#).

Prerequisites

- Your SAP application involves documents.
- You do not have any special requirements concerning the BDS document management.

Process Flow

1. Analysis

This analysis phase involves specifying general questions, as well as the scope of the intended integration of the BDS into the relevant R/3 application:

- What is the business background?
- For which release is the integration of the BDS to be completed?
- What sorts of documents are to be stored?
For example: Bitmaps, Office documents.
- How many documents are there likely to be?
- Does the SAP3 application need its own administration tables?
- What delivery class are the administration tables to have?
- What is the application class ID?
- What sort of application class is involved?
 - Business Object Repository
 - R/3 Class Library
 - Others
- Is a transport connection required?
- Is the application context client-specific?
- Which of the BDS methods are likely to be used?



`CALL_NAVIGATOR` (you want to call the BDN) or
`CREATE_VARIANT_WITH_FILES` (you want to create a variant and transfer the file name simultaneously).

- Are specific attributes required? If so, then which attributes?

2. Discussion and Decision

Integration of the BDS in SAP Applications

In this phase, your project team discusses how your requirements can best be realized. The team decides on the data model to be used (see also [Content Models \[Ext.\]](#)) and the methods to be implemented for your application.

3. Implementation

You integrate the methods required for your application. The methods provided are described in the sections under [General Methods \[Page 24\]](#) and [Specific Methods \[Page 41\]](#).

Example: CREATE_WITH_FILES

Example: CREATE_WITH_FILES

```
data: o_document_set type ref to cl_bds_document_set.
create object o_document_set.
move: '1'                to wa_signature-doc_count,
      '1'                to wa_files-doc_count,
      'C:\SAPPCADM'      to wa_files-directory,
      'BDSPresentation.PPT' to wa_files-filename.
append wa_signature to i_signature.
append wa_files to i_files.

call method o_document_set->create_with_files
  exporting
    classname      = 'MyApplication'
    classtype      = 'OT'
  changing
    object_key     = i_object_key
    files          = i_files
    signature      = i_signature.
```

Example: GET_WITH_FILES

```
data: o_document_set type ref to cl_bds_document_set.
create object o_document_set.
move: '1' to wa_signature-doc_count,
      'BDS_LOC1 B18EBB28E773D21197E70060B0672A3C'
      to wa_signature-doc_id,
      '1' to wa_signature-doc_ver_no,
      '1' to wa_signature-doc_var_id,
      '1' to wa_files-doc_count,
      'C:\SAPPCADM' to wa_files-directory,
append wa_signature to i_signature.
append wa_files to i_files.

call method o_document_set->get_with_files
exporting
      classname      = 'MyApplication'
      classtype      = 'OT'
      object_key      = '0815'
changing
      files           = i_files
      signature       = i_signature.
```

Business Document Navigator

Business Document Navigator

Purpose

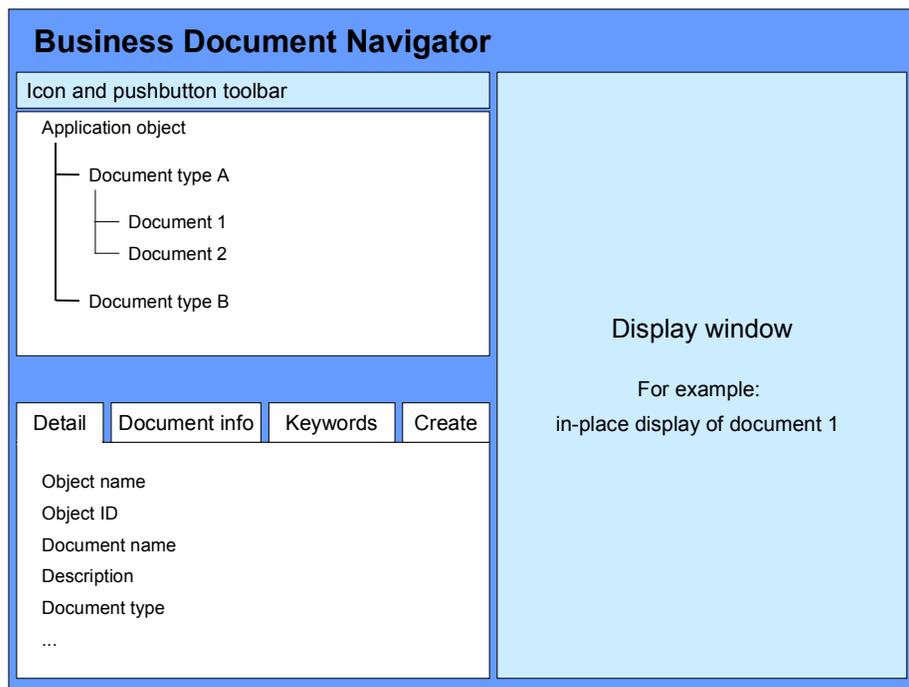
The **Business Document Navigator (BDN)** is the central tool for processing documents. You can use the BDN to execute standard document management functions on documents managed, for example, by SAP ArchiveLink.

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

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

For more information, see [Displaying the Hit List \[Page 63\]](#).

- Display individual documents

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

For more information, see [Displaying Documents \[Page 64\]](#).
- Display detailed information on stored documents

For more information, see [Displaying Detailed Information on a Document \[Page 65\]](#).
- 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 \[Page 66\]](#).
- 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 \[Page 69\]](#).
- 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 \[Page 72\]](#).
- Display keywords for a document

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

For more information, see [Displaying Keywords \[Page 76\]](#).
- Display version string

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

For more information, see [Displaying the Version String \[Page 77\]](#).
- Copy documents

For more information, see [Copying Documents \[Page 78\]](#).
- Transport documents

For more information, see [Transporting Documents \[Page 86\]](#).
- Delete documents

For more information, see [Deleting Documents \[Page 79\]](#).
- 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 \[Page 80\]](#).
- Change existing keywords

Business Document Navigator

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

For more information, see [Changing Attributes \[Page 81\]](#).

- Refresh the hit list
For more information, see [Refreshing the Hit List \[Page 83\]](#).
- Annotations for BDS documents that are displayed using the EAI viewer
For more information, see [Creating Annotations \[Page 82\]](#).
- Display documents using HTML control
For more information, see [Displaying Documents \[Page 64\]](#).
- Store WWW addresses
For more information, see [Storing Documents \[Page 66\]](#) and [Storing and Displaying WWW addresses \[Page 71\]](#).
- Navigation to any WWW address
For more information, see [Storing and Displaying WWW Addresses \[Page 71\]](#).
- Export documents to frontend
For more information, see [Exporting Documents \[Page 85\]](#).
- Lock documents when changes are being made
For more information, see [Editing Documents \[Page 80\]](#).
- Create a transport request when documents are deleted
For more information, see [Transporting Documents \[Page 86\]](#).
- Store documents using drag and drop
For more information, see [Storing Documents Using Drag and Drop \[Page 68\]](#)
- 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 \[Ext.\]](#)). 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

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

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

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



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 \[Page 71\]](#)).



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

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

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 \[Page 66\]](#)).

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

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 \[Page 66\]](#).

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

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

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

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 \[Page 75\]](#).

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

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 \[Page 69\]](#).
- All keywords created when documents were stored
See also [Storing Documents \[Page 66\]](#).
- All keywords created when attributes of SAP ArchiveLink documents were created or changed
See also [Changing Attributes \[Page 81\]](#).

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

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

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

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 \[Ext.\]](#)).

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 \[Page 86\]](#)).

Activities

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

Editing Documents

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

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

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

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

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 \[Page 79\]](#)).

Activities

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