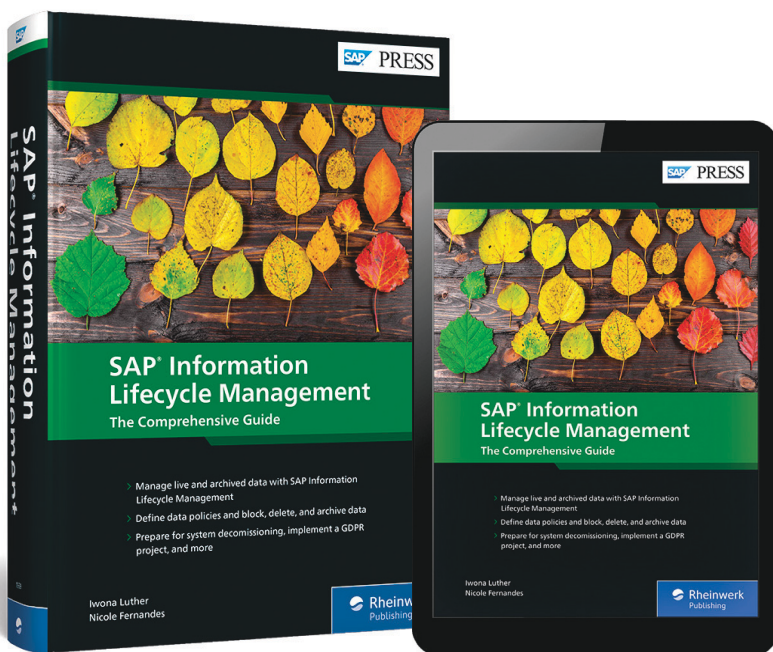


# Innovations in SAP® ILM

from the book

## SAP® Information Lifecycle Management: The Comprehensive Guide

by Iwona Luther and Nicole Fernandes



“Get the latest information  
on SAP ILM!”



PRESS

## Innovations in SAP ILM

Since the optimization of software solutions is an ongoing process, SAP ILM has been further developed while this book went to press. This online appendix introduces the latest innovations made to SAP ILM and describes how to configure, implement, and use them to optimize the treatment of data in your SAP system.

This appendix has been written as a direct companion to *SAP Information Lifecycle Management: The Comprehensive Guide*. Each major section in this appendix corresponds to a chapter in the book. The subsections within those major sections each relate to a topic that has been expanded or is entirely new. These subsections will each start with a text box indicating which section in the book they correspond to.

We do not recommend trying to read this appendix straight through, as the experience would be disjointed without the context the sections in the book provide. Its best use is as supplementary material, reading the new and updated material after you have finished the corresponding section. The one exception to this is the section [Destruction of Attachments](#) on destruction of attachments, which is an entirely new feature in SAP ILM.

## Chapter 2

This section will cover topics from Chapter 2 of the book that have been expanded since publication.

### New Transaction to View Change Documents for Audit Areas

**Note**

This section expands coverage of Section 2.3.2.

You might be wondering how changes to the Customizing for your audit areas can be logged. At the beginning of SAP ILM, this was only possible by logging changes to table entries in the related database tables. However, evaluating such logs (table DBTABLOG, Log Records for Table Changes) is cumbersome. Recently, change documents have been provided for this purpose, and the required change document object (object class) ILM\_ARA is available in this

context. You can display the resulting change documents using the new Transaction ILM\_CHDOC\_ARA (Change Documents: ILM Audit Areas; see [Figure 1](#)).

**Change Documents: ILM Audit Areas**

Object ID: [Yellow Highlighted Field]

User: [Field]

From Date: ☒

From Time: 00:00:00

To Date: ☒

To Time: 23:59:59

SAP | ILM\_CHDOC\_ARA | Idciy3 | INS | [Icons]

**Figure 1** Transaction ILM\_CHDOC\_ARA (Change Documents: ILM Audit Areas)

## Adding Comments to Policies and Rules

### Note

This section expands coverage of Section 2.5.4.

A **Comment** field for comments per policy is available in Transaction IRMPOL. You can use it to enter information related to a newly created or edited policy. Here, you can enter a comment of up to 255 characters.

As you can see from SAP Note 2936309 (the **Comment** column is not visible in Transaction IRMPOL, ILM Policies), a further column with the **Comment** heading is available after clicking on the **Open Settings Dialog** icon. Here you can store a comment for each line of a rule.

## New Transaction to View Change Documents for ILM Object Groups

### Note

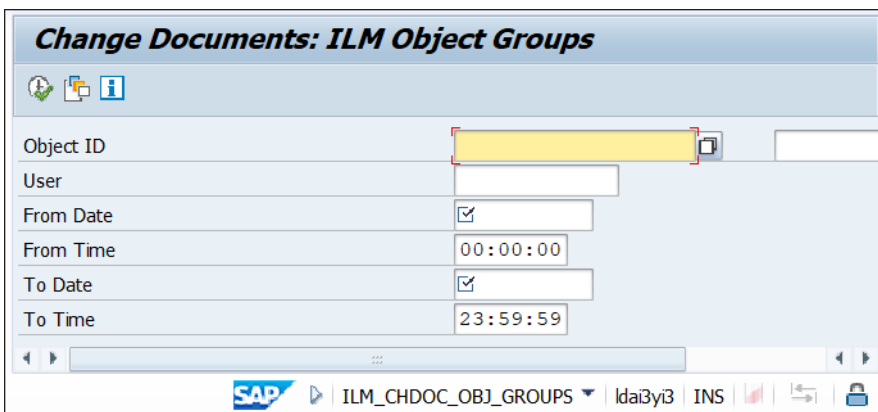
This section expands coverage of Section 2.5.6.

How does logging take place when you customize the following routines?

- Create, change, delete object groups

- Create, change, delete rule groups
- Assign SAP ILM objects to object groups or remove this assignment

At the beginning of SAP ILM, such logging was only possible by logging changes to table entries in the related database tables. However, evaluating such logs (table DBTABLOG, Log Records for Table Changes) is cumbersome. Recently, change documents have been provided for this purpose, and the required change document object (object class) ILM\_OBJ\_GROUPS is available in this context. You can display the resulting change documents using the new Transaction ILM\_CHDOC\_OBJ\_GROUPS (Change Documents: ILM Object Groups; see [Figure 2](#)).



**Figure 2** Transaction ILM\_CHDOC\_OBJ\_GROUPS (Change Documents: ILM Object Groups)

## Logging Changes to Policies and Rules

### Note

This section expands coverage of Section 2.5.8.

You might be wondering how changes to the Customizing for your policies and rules can be logged. At the beginning of SAP ILM, this was only possible by logging changes to table entries in the related database tables. However, evaluating such logs (table DBTABLOG, Log Records for Table Changes) is cumbersome. Recently, change documents have been provided for this purpose, and the change document object (object class) ILM\_POL is available in this context. You can display the resulting change documents using the new Transaction ILM\_CHDOC\_POL (Change Documents: ILM Policies; see [Figure 3](#)).

**Figure 3** Transaction ILM\_CHDOC\_POL (Change Documents: ILM Policies)

## Supported Attachment Types

### Note

This section expands coverage of Section 2.7.1.

You’ve already learned about the data that can be destroyed using SAP ILM. Regarding attachments—that is, linked documents—the following types are supported:

- SAP ArchiveLink (entries in table TOAnn)
- Generic Object Service (GOS)
- Document management

## Data Destruction Object and the IDENTITY SAP ILM Object

### Note

This section expands coverage of Section 2.7.3.

Let’s discuss a fairly new data destruction object, called IDENTITY (complete deletion of user master data), and the SAP ILM object of the same name. You can use them to delete user data from the technical user master in your system. For more information, see SAP Notes 550718 and 2826256, as well as the other SAP Notes mentioned therein.

It is also worth mentioning that the destruction of user data is not possible if the user is referenced in instances of BOR object types (such as documents)

that have been included in a legal hold (see Chapter 3, Section 3.6). It does not matter whether this has been done manually or via E-Discovery. This also applies if these instances have been archived, and it applies so long as the legal case status has not yet been assigned the value 50 (*case closed*).

## Destruction of Attachments

### Note

This section is entirely new and does not expand a particular section in Chapter 2. However, the topic does belong in the grouping of topics contained in Chapter 2.

Let's look in detail at the destruction of attachments—that is, linked documents. At the beginning, SAP ILM only supported the destruction of SAP ArchiveLink attachments (entries in table `TOAnn`). Recently, destruction of the following attachments has also been made possible:

- GOS
- Document management

The destruction of these types of attachments is provided at all relevant points—that is:

- The archiving action in the archiving object
- The data destruction action in the archiving object
- The data destruction object

### **SAP ArchiveLink and GOS**

Let's start with the SAP ArchiveLink attachments. Here we can distinguish two cases:

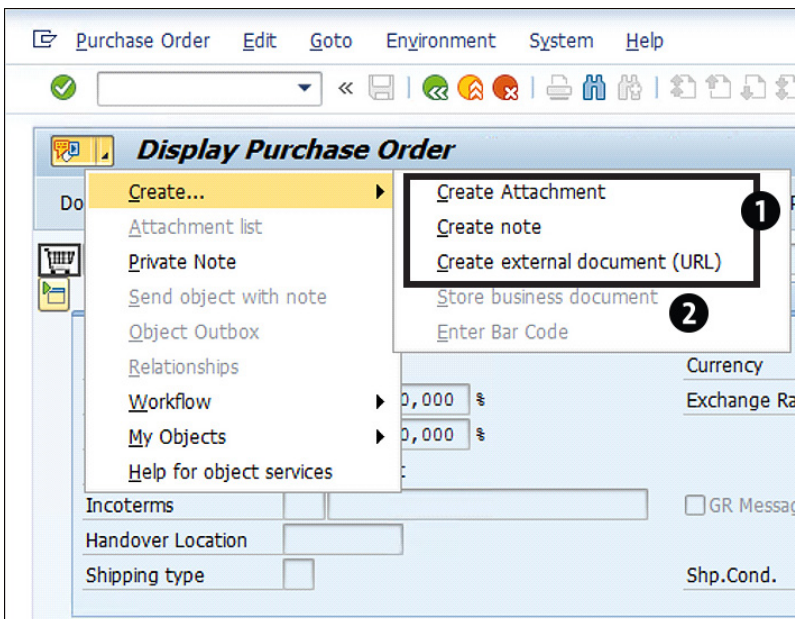
1. Attachments for which you have defined a retention period in SAP ILM (the `AL_DOCUMENTS` SAP ILM object, for documents stored using SAP ArchiveLink) and have propagated this retention period to the certified SAP ILM store
2. Attachments for which you did not do this; these attachments are supposed to be destroyed together with the leading object (e.g., the related FI document)

We have already described how you carry out the destruction in the first case in Chapter 2, Section 2.7.4. Now we will describe the second case. This is an

option that has only recently been made available. It is available both for SAP ArchiveLink and GOS attachments. We will therefore describe these together. Then we will look at the attachments from the Document Management System (DMS).

Destruction together with the leading object means that you define retention periods for this object (e.g., for an order or work plans) in SAP ILM. After this period, you plan the destruction of the leading object. Related attachments will also be destroyed in the process.

How can SAP ILM determine the existence of related attachments and initiate their deletion? To explain, we'll start with an example of how you can create GOS ❶ or SAP ArchiveLink ❷ attachments. [Figure 4](#) shows this using Transaction ME23N (Display Purchase Order) as an example.












**Figure 4** Creating SAP ArchiveLink or GOS Attachments for Leading Object











If you have created a GOS attachment using the appropriate menu items, a new entry is created in table SRGBTBREL (Links in GOS Environment). [Figure 5](#) shows you the fields in which the link type—for example, URL, note, or attachment ❶—as well as the BOR object type and key are stored ❷.

If you have created an SAP ArchiveLink attachment, a new entry is created in table TOA01 (Link Table 1). [Figure 6](#) shows you the fields in which the BOR object type and key are stored ❶.



Transparent Table	SRGBTBREL	Active
Short Description	Relationships in GOS Environment	
Attributes   Delivery and Maintenance   Fields   Input Help/Check   Currency/Quantit		
<div> Srch Help   Predefined Type</div>		
Field	Key	Ini... Short Description
CLIENT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Client
BRELGUID	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Globally Unique Identifier
RELTYPE	<input type="checkbox"/>	<input checked="" type="checkbox"/> Relationship type
INSTID_A	<input type="checkbox"/>	<input checked="" type="checkbox"/> Instance Ident. in BOR Compat. Persistent Object References
TYPEID_A	<input type="checkbox"/>	<input checked="" type="checkbox"/> Type of Objects in Persistent Object References
CATID_A	<input type="checkbox"/>	<input checked="" type="checkbox"/> Category of Objects in Persistent Object References

**Figure 5** Table SRGBTBREL (Details about GOS Attachments)

Transparent Table	TOA01	Active
Short Description	Link table 1	
Attributes   Delivery and Maintenance   Fields   Input Help/Check   Currency		
<div> Srch Help   Predefined Type</div>		
Field	Key	Ini... Short Description
MANDT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Client
SAP_OBJECT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> SAP ArchiveLink: Object type of business object
OBJECT_ID	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> SAP ArchiveLink: Object ID (object identifier)
ARCHIV_ID	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Content Repository Identification
ARC_DOC_ID	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> SAP ArchiveLink: Document ID
AR_OBJECT	<input type="checkbox"/>	<input type="checkbox"/> Document type

**Figure 6** Table TOA01 (Details about SAP ArchiveLink Attachments)

You may wonder what the entries in these tables would look like if the same attachment (original document) is linked to several leading objects. If the same attachment is linked to several leading objects (e.g., a purchase request and later a purchase order), this would result in several entries in the tables. This ensures that the attachment exists as long as it's needed. As a result, each leading object only destroys its entries in the tables mentioned. This applies to GOS and SAP ArchiveLink attachments.

Let's return to the question we asked earlier: How does SAP ILM determine the existence of related attachments and initiate their deletion?

The developer of the related SAP ILM object must tell SAP ILM what the BOR object type is and which fields are used to form the key. For the first specification, a new entry is made in Transaction IRM\_CUST\_BS (IRM Customizing (Business Suite)): **Mapping BOR Object Type for Destroying Attachments** ([Figure 7](#)). The key is specified using the **Determination of Object Instance Key for BOR Obj.** entry, which has already existed for some time.

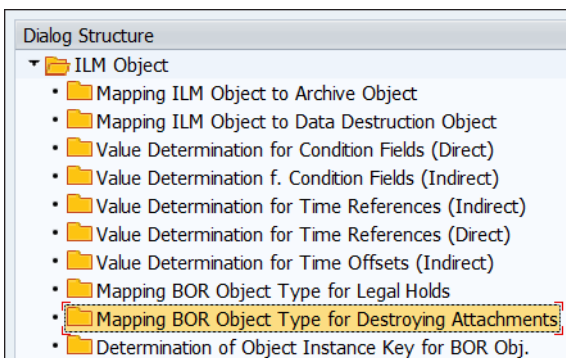


If the sequence of the entries mentioned is slightly different in your system, please refer to SAP Note 2936764.

### Determining the Object Instance Key

You should note that in more complicated cases, the key determination can take place in a BAdI. The **Determination of Object Instance Key for BOR Obj.** entry is then empty. We describe this in detail in Chapter 9, Section 9.5.9, in the “Settings for Destroying Attachments” section.

This also means that you can look in these entries, especially in the first one (**Mapping BOR Object Type for Destroying Attachments**), to see which SAP ILM objects (and therefore which business objects) provide this new functionality for the destruction of attachments in the respective release. Alternatively, you can look at the entries in the related table LRM\_T\_BS\_ATTCH (Mapping BOR Object Type for Destroying Attachments).



**Figure 7** Transaction IRM\_CUST\_BS (Map BOR Object Type for Destruction of Attachments)

### Data Destruction Object and Mapping BOR Object Type for Destroying Attachments

It is not technically necessary to maintain the two entries just discussed (e.g., **Mapping BOR Object Type for Destroying Attachments**) in Transaction IRM\_CUST\_BS (IRM Customizing (Business Suite)) if the SAP ILM object has been assigned to a data destruction object. The reason is that in this case the prompt for destroying the attachment takes place directly in the data destruction program. For the archiving object, the ADK handles this at the end of the delete program. When archive files are destroyed from an SAP ILM-enabled store, SAP ILM handles this after the archive file has been destroyed. The entries (particularly the object instance key) are needed here.

However, we recommend that developers maintain these entries to give end users a way to see which SAP ILM objects consider the destruction of attachments.

SAP Note 2787893 describes the availability of the discussed process of destruction and provides details about GOS attachments.

The “References” section of the SAP Note provides a list of applications (SAP ILM objects) that support this new option for attachment destruction. The related SAP Notes contain the necessary Customizing or corrections. This enables you to check as of when a particular application, such as Contract Accounts Receivable and Payable (FI-CA) or Collections and Disbursements (FS-CD), supports this functionality.

SAP Note 2844211 describes the availability of the discussed destruction process and contains additional SAP ArchiveLink-specific information.

SAP Notes 2865971 and 2912903 provide additional information and required corrections for customer and vendor master data.

When you have destroyed a leading object and attachments have also been destroyed in the process, you can see the related tables and the number of deleted table records in the ADK statistics (statistics of written data objects) of the deletion program (not the archive write program). The last line in [Figure 8](#) shows an example of the GOS attachments; the first line in the same table shows the same for SAP ArchiveLink attachments.

Test Mode: Statistics for Deleted Data Objects		
Archive File Key	001274-001PM_PLAN	
Number of Deleted Data Objects	1	
File can be destroyed	001274-001PM_PLAN	
Deleted Database Space in MB	0,001	
- Tables	0,001	
Type	No.	Description
Archivelink Connections	6	
BDS Connection Entries	0	
DRAD	0	Document-Object Link
EAMWS_SAFETY	0	Safety Data for Objects
SOFM	8	SAPoffice: Folder contents
SRGBTBREL	6	Relationships in GOS Environment

**Figure 8** Destruction of Attachments in ADK Statistics of Delete Program

**Note: Data Destruction Objects and Table Statistics**

A comparable table statistic is not currently available for data destruction objects. Here, you can't tell from the logs or statistics whether attachments have also been destroyed.

Let's return to the destruction of SAP ArchiveLink attachments for which you have defined retention periods in SAP ILM (SAP ILM object AL\_DOCUMENTS, documents stored via SAP ArchiveLink) and have propagated these retention periods to the certified SAP ILM store. (It is important that the propagation has taken place.) If you now destroy the related leading object (e.g., a purchase order or a maintenance plan), SAP ILM recognizes that the AL\_DOCUMENTS SAP ILM object (documents stored via SAP ArchiveLink) is being used, and SAP ILM does not destroy the related attachments. You will find corresponding information in the log. As you can see in [Figure 9](#), you still get an **Archivelink Connections** line, but it now indicates 0. The additional **AL\_DOCUMENTS** entry tells you how many connection entries will not be destroyed for this reason.

Type	No.	Description
AL_DOCUMENTS	1	
Archivelink Connections	0	
BDS Connection Entries	0	
DRAD	0	Document-Object Link
EAMWS_SAFETY	0	Safety Data for Objects
SOFM	1	SAPoffice: Folder contents
SRGBTBREL	1	Relationships in GOS Environment

**Figure 9** Destruction of Attachments in ADK Statistics of the Delete Program: Special Features for AL\_DOCUMENTS SAP ILM Object

**Archiving Object and Class ARCHIVELNK**

We recommend that you no longer use the ARCHIVELNK archiving object and archiving class to archive or destroy SAP ArchiveLink attachments. SAP will mark these objects as obsolete in the future.

Let's look at another alternative for the destruction of GOS attachments: the destruction together with the related leading object (e.g., for an order or work plans) from the certified SAP ILM store (and not from the database, as discussed so far).

There is no separate archiving object for the GOS attachments. However, there is the SGOS archiving class, and there are archiving objects that use this archiving class. For this case, the selection screen of the archive write program offers the **Archive GOS Attachments** checkbox, shown in [Figure 10](#).

**Archive Administration: Initial Screen**

Archiving Object: **FI\_MKKDUNN** FI-CA: Dunning History

**Archiving Quantity Selection**

Executed On		to
Identification		to
Contract		to
Business Partner		to
Contract Account		to
Dunn.counter for multip.dunn.		to

**Restrictions**

Archiving Key Date:

**Options**

☐ Archivieren von GOS Anlagen

**Processing Options**

☒ Test mode

**Figure 10** Application of Archiving Class SGOS in Archiving Objects

The related **F1** input field help means that if the checkbox is selected, the GOS attachments are archived together with the business object. Note that depending on the number or file size of the attachments, this may impair system performance during the archiving process.

The subsequent deletion run releases the corresponding storage locations for the GOS attachments. This means that both the related entries in table SRGBT-BREL (Links in the GOS Environment) and the GOS attachment are deleted (because it now resides in the archive file). As a result, you do not need to run the cleanup Transaction SBCS\_REORG (BCS: Reorganization), which corresponds to report RSBCS\_REORG (BCS: Reorganization of Documents and Send Requests).

### Archive GOS Attachments Checkbox in an SAP S/4HANA Cloud System

If such an archiving object is used in SAP S/4HANA Cloud, the **Archive GOS Attachments** checkbox is always selected and not displayed.

Finally, let's provide some information on the cleanup transactions and reports just mentioned. Cleanup reports RSGOSRE01 and RSGOSRE02 have been adjusted to prevent the premature destruction of GOS attachments. (The two reports do essentially the same thing, but one is newer and offers, for example, more selections on the selection screen.)

The task of these reports is to delete links to GOS attachments (table SRGBTBREL) for which the leading object (e.g., for a purchase order) no longer exists in the database.

However, if the leading object has been archived, or if you have stated in SAP ILM that the leading object is to be destroyed from the database using SAP ILM, the link to the attachment must not be deleted by the report. This destruction should take place in the ways discussed earlier.

The reports have been adapted accordingly. In detail: when a BOR object type has been stored in SAP ILM Customizing under **Mapping BOR Object Type for Destroying Attachments**, it is not considered by the cleanup report.

GOS attachments with deleted links (entries in table SRGBTBREL) are finally removed by cleanup Transaction SBSCS\_REORG (BCS: Reorganization).

We recommend that you execute this transaction regularly to prevent the link tables from becoming too large. In SAP S/4HANA Cloud, a corresponding background job is scheduled for you to do this.

Finally, if you are interested in the technical implementation of the functionality just discussed, the answer lies in the enhancement implementations ([Figure 11](#)):

- GOS\_ILM\_DESTRUCTION (destruction of GOS attachments during SAP ILM data destruction)
- ALINK\_ILM\_DESTRUCTION (destruction of SAP ArchiveLink attachments during SAP ILM data destruction)

Both contain implementations for two BAdI methods:

- The `OPERATE_AFTER_FILE_DESTRUCTION` method (postprocessing of the clean-up action in the remote system) of `BADI_ILM_PRE_DESTR_FILE_ACTION` (cleanup action before deleting an archive file)
- The `OPERATE_ON_FILE_DESTRUCTION` method of `BADI_ILM_PRE_DESTR_OBJ_ACTION`

We described these BADIs in more detail in Chapter 2, Section 2.7.6.

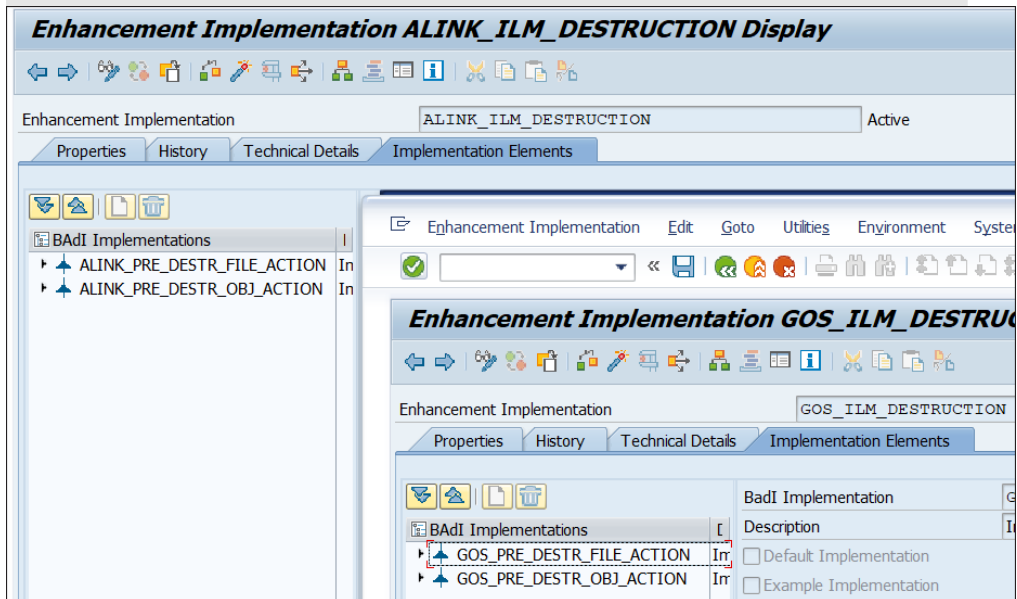


Figure 11 Enhancement Spot ES\_ILM\_DESTRUCTION

## Document Management

Let's now discuss a further new development in the SAP ILM environment: the destruction of DMS attachments.

After a closer look, this is a question of destroying the document info record. The record consists, among other things, of the following:

- The metadata (header data) of the attachments, such as their location or classification data. This information is stored in table DRAW (document info record).
- The attachment itself.
- The link information between the leading business object (e.g., the purchase order) and its attachment. This is stored in the DRAD\* tables, one of which, for example, is table DRAD (document-object link).

The following are available in the DMS environment:

- Archiving object CV\_DVS (document management)
- Archiving class CV\_OBJL (document info record object links)
- SAP ILM object CV\_DVS (document management)

Let's now discuss the area of application of these objects.

The archiving class should be used by all archiving objects that represent a leading object (business object) for which there can be a DMS attachment in the system. The archiving class writes the link information between the leading business object (e.g., the purchase order) and its attachment to the archive. As you will remember, this link information is the third part in our list of document info record components. It is now obvious that only this part of the document info record is written to the archive by the archiving class. At the same time, the DRAD-ARCHFLAG table field (archiving flag) is set in the link information.

What else does the archiving class do? An attachment potentially can have more than one leading object, so more than one business object (e.g., a purchase order, a sales order) can be linked to the same attachment. In this case, there is a document info record that has an entry in the link tables (DRAD\* tables) for each leading object. If the leading object is archived, the DRAD-ARCHFLAG table field is set.

If the last leading object (or the only one if the link is 1:1) is now archived using the leading object, the class also sets another field, this time in the metadata (header data) of the attachment: the DRAW-LOEDK table field (deletion flag).

Now we'll describe the areas of application of the CV\_DVS archiving object (document management) and distinguish among different scenarios. Scenario 1 applies if the following is true:

- The volume of document info records is supposed to be reduced—for example, because they have a lot of classification or descriptive data.
- The related attachments do not have a leading object (or the link to it has been removed), which is the case, for example, in some applications in the pharmaceutical or oil industry.

In this case, we recommend that you use the archiving object with the archiving SAP ILM action.

You can determine the retention period of the archive files created in this way using the related SAP ILM object, CV\_DVS (Document Management). Make



sure that you set the **Linked Object** column correctly—that is, so that the rule applies to attachments without a leading object. The second line in [Figure 12](#) provides an example.

Linked Object	Document Type	To	Authori... Group	Inherit Rule	Min.Retention Period	Max.Ret... Period	Retention Prd Unit	Time Ref.
1	101			<input checked="" type="checkbox"/>				
0				<input type="checkbox"/>	5		Year	End of Year
1	111			<input type="checkbox"/>	5		Year	End of Year

**Figure 12** CV\_DVS SAP ILM Object

The document info records archived in this way can be displayed in SAP S/4HANA (on premise) and SAP S/4HANA Cloud with the SAP\_CV\_DVS\_DMS AS info structure (display DMS documents).

#### Read Access to Archived Document Info Records

Note that at the time of publishing this appendix, only the metadata (header data) and the link information could be displayed using the info structure mentioned here. The attachment itself could not be displayed.

#### DRAD-ARCHFLAG (Archiving Flag) and DRAW-LOEDK (Deletion Flag) Table Fields

The CV\_DVS archiving object (document management) does not set the DRAD-ARCHFLAG table field (archiving flag). This is only set by the archiving class.

The archiving object also does not set the DRAW-LOEDK table field (deletion flag). This is also not necessary. In the case of this table field, this is because you can decide yourself whether the archiving object should process all document info records or only those with the deletion flag. To do this, use the **Only with Deletion Indicator** checkbox, as shown in [Figure 13](#).

**Edit Variants: Report DVSARCH1, Variant**

Attributes

Document Info Record

Document type

Further Selection Criteria

☐ Only with deletion indicator

All Documents

☐ All Documents

**Figure 13** CV\_DVS Archiving Object: Selection Screen

Let's move on to scenario 2, which applies if the following is true:

- The volume of document info records is to be reduced.
- The associated attachments have a leading object (e.g., a purchase order).

It is also recommended here that you use the archiving object with the archiving SAP ILM action.

You can also define the retention period of the archive files created in this way using the corresponding SAP ILM object, `CV_DVS` (Document Management). Make sure that you set the **Linked Object** column correctly—in other words, such that the rule applies to attachments *with* a leading object. The first and third lines in [Figure 12](#) show an example. Whether or not you want to grant the retention period from the leading object is your decision.

#### Two Retention Periods per Document Type

We recommend that you define two retention periods per document type: one for attachments with a leading object and one for attachments without a leading object (see the **Linked Object** column in `CV_DVS` SAP ILM object).

If the attachment has more than one leading object (e.g., a purchase order, a sales order), the longest retention period is taken as the retention period for the document info record.

As described previously, the document info records archived in this way can be displayed in SAP S/4HANA (on premise) and SAP S/4HANA Cloud with the `SAP_CV_DVS_DMS` SAP AS info structure (display DMS documents). The previously mentioned restriction on read access to the archived DMS attachments also applies here.

#### Order

The recommended sequence in this scenario is to first archive the leading object (e.g., the purchase order) using the relevant archiving object, and only then archive the attachment using the `CV_DVS` archiving object (document management).

Let's move on to scenario 3, which applies if the following is true:

- The volume of the document info records did not have to be reduced prematurely with the help of the SAP ILM Archiving action.
- The document info records are to be destroyed after the retention period has expired.

Whether or not the attachments have a leading object (e.g., a purchase order) is not important. Whether the retention period was defined directly or via rule inheritance is also irrelevant.

You may already suspect—correctly—that in this case the Data Destruction SAP ILM action is used. For the data destruction to run correctly here, it is technically necessary that the link information (DRAD\* tables) be located in the database.

#### Side Effects: Deleting Link Information from the Database Using the CV\_OBJL Archiving Class (Document Info Record Object Links)

CV\_DVS archiving object (document management) is SAP ILM-enabled, the developers of the archiving object are asked by the leading object to set a certain parameter for the CV\_OBJL archiving class (document info record object links) so that the link information (DRAD\* tables) is *not* deleted from the database in the delete program (which would otherwise be the typical behavior in a delete program). This is due to the fact that this information is technically necessary to determine the correct retention period of the document info records using the CV\_DVS SAP ILM object (document management). In detail: if the link information is no longer available, only rules can be processed for which the **Linked Object** column is set to a value that indicates that *no* linked object exists, as in line 2 in [Figure 12](#).

One way to avoid these issues as an end user is to follow the recommendation given here and define two retention periods per document type. One should be defined for attachments with a leading object and one for attachments without a leading object (see the **Linked Object** column in the CV\_DVS SAP ILM object).

If the attachment has more than one leading object (e.g., a purchase order or a sales order), the longest retention period is used as the retention period for the document info record.

## Chapter 3

This section will cover topics from Chapter 3 of the book that have been expanded since publication.

### New Change Document Objects

#### Note

This section expands coverage of Section 3.5.2.

Having read Section 3.5.2 about the functions and configuration of the data controller rule framework, you might ask how changes to the Customizing of generated rules and policies are logged. At the beginning of SAP ILM, this was only possible by logging changes to table entries in the corresponding database tables. However, evaluating such logs (table DBTABLOG, Log Records for Table Changes) is cumbersome. Change documents provide a remedy here, and, accordingly, the ILM\_DCRF and ILM\_DCRF\_CONF change document objects (object classes) have been made available for this purpose. You can display the resulting change documents using the new Transaction ILM\_CHDOC\_DCRF (Change Documents: ILM DCRF; see [Figure 14](#)). You can choose between two change objects, as shown in the **[F4]** input help for the first input field. When you select **ILM\_DCRF**, you can evaluate change documents that are created, changed, or deleted when business rules are created, changed, or deleted. When you select **ILM\_DCRF\_CONF**, you can evaluate change documents that were created when the standard audit area, the time reference or time offset, the data controller, and the purpose were defined.

**Figure 14** Transaction ILM\_CHDOC\_DCRF (Change Documents: ILM DCRF)

## Chapter 4

This section will cover topics from Chapter 4 of the book that have been expanded since publication.

### More Complex System Landscapes with Multiple Master Systems and Dependent Systems

The standard EoP check for customers and vendors was initially designed to work only in a system landscape with one master system and several dependent sys-

tems. A more complex system landscape with several master systems and dependent systems was not supported at first. SAP Note 2883431, and the additional SAP Notes mentioned therein, describes how you can register the master system(s) in these more complex system landscapes. The related Section 4.3.1 in our book describes the individual steps for the simple, original case of a system landscape with one master system and several dependent systems. For complex system landscapes, you can skip the passages relating to simple system landscapes and follow the instructions in the mentioned SAP Notes.

## Enhancing EoP Checks

### Note

This section expands coverage of Section 4.3.1.

After reading the first few paragraphs of Section 4.3.1, you now know how to check which applications participate in the EoP check to give an OK or veto for blocking a master record. Let's now add that you can enhance the EoP checks of an application name. In other words, you can add new, additional checks for an application name already entered (provided by SAP). To do this, select one of the following:

- **Register Application Classes for End-of-Purpose Check**, as shown in Figure 4.12 in Chapter 4 of the book. Select this if you are dealing with customers or vendors. The system then displays the screen shown in [Figure 15](#).
- **Define Application FB Registered for the EoP Check**, as shown in Figure 4.13 in Chapter 4 of the book. Select this for business partners. The screen shown in [Figure 16](#) appears.

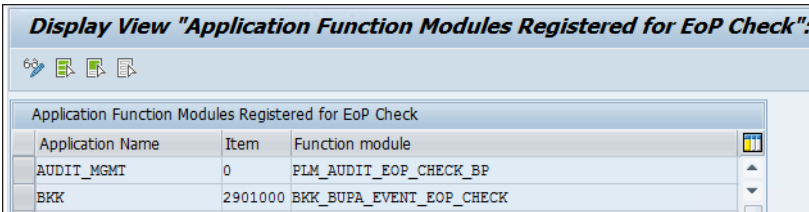
Change View "Register Application classes": Overview							
<div> </div>							
<b>Dialog Structure</b> <ul style="list-style-type: none"> <li>Register Application classes</li> <li>Define priorities</li> </ul>		<b>Register Application classes</b>					
ID	Type	Appl Name	Item Num...	Application Description	Registered class for EoP checks	General	Comp. Code
1	Customer m...	ERP_PP_SFC	0	Production Order	CL_WUC_PP_SFC_EOP_CHECK	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
1	Customer m...	ERP_PROMOTION	0	Promotion	CL_PROMOTION_CV_EOP_CHECK	<input checked="" type="checkbox"/>	<input type="checkbox"/>
1	Customer m...	ERP_PS	0	Project System	CL_NOTIIF_EOP_CHECK	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
1	Customer m...	ERP_PS	1	Project System	CL_CVP_PS_EOP_CHECK	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
1	Customer m...	ERP_QM_QN	0	Quality notification	CL_NOTIIF_EOP_CHECK	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

**Figure 15** Adding Further Checks for Application Name Already Entered (Customer, Vendor)

### Changes to Checks Delivered in the Standard System

We advise you not to delete or modify checks delivered in the standard system.

**Display View "Application Function Modules Registered for EoP Check":**



Application Name	Item	Function module
AUDIT_MGMT	0	PLM_AUDIT_EOP_CHECK_BP
BKK	2901000	BKK_BUFA_EVENT_EOP_CHECK

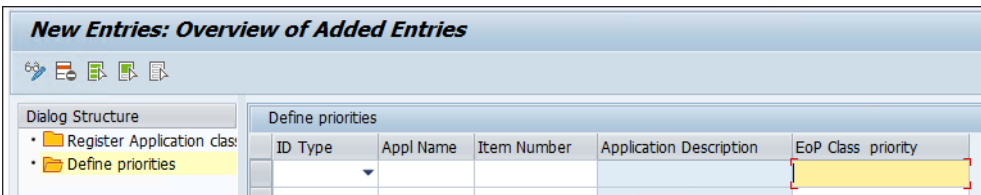
**Figure 16** Adding Further Checks for Application Name Already Entered (Business Partner)

As shown in [Figure 17](#) and [Figure 18](#), you can now create another item number and a corresponding class or function module that performs the checks. In this case, you would define an additional EoP check for an application name delivered by SAP.

You also have the option of prioritizing the sequence in which the individual checks are prompted, so that, for example, a particular customer check is executed first for performance reasons. Choose from the following options:

- For customers or vendors, use the **Define Priority** subitem, as shown in [Figure 15](#), and then the last column for which the **F1** field help contains the description **Priority of the Class Registered for the End of the Purpose**.
- For business partners, use the **Define Application Priority for End-of-Use-Check** entry, as shown in Figure 4.13 in the book.

**New Entries: Overview of Added Entries**



ID Type	Appl Name	Item Number	Application Description	EoP Class priority

**Figure 17** Priority of Further Checks (Customer, Vendor)

## Further Available Condition Fields

### Note

This section expands coverage of Section 4.3.3.

Having made yourself familiar with the preparatory steps for blocking master data in SAP ILM, let's turn back to the available condition fields for the four different types of master data that you can use. All four master data types have in common that you can use the **Application Name** and **Application Rule Variant** condition fields. Which other fields can you use?

As you can see in [Figure 18](#), the **Company Code**, **Account Group**, and **Country Key** condition fields are also available for customers and vendors. You can use the additional fields to adapt your rules more flexibly to your requirements. This applies to rules for both the residence period and the retention period. For example, you could distinguish between different types of master data.

We advise against using other fields, such as **Plant** and **Sales Organization**. These are errors within the scope of delivery and should be deleted.

Field Name	Description
APPL_NAME	Application Name
APPL_RULE_VARIANT	Application Rule Variant
BUKRS	Company Code
KTOKD	Account group
BS_COUNTRY_OF_BUKRS	Country Key

Field Name	Description
APPL_NAME	Application Name
APPL_RULE_VARIANT	Application Rule Variant
BUKRS	Company Code
KTOKK	Account group
BS_COUNTRY_OF_BUKRS	Country Key

**Figure 18** Other Available Condition Fields (Customer and Vendor)

[Figure 19](#) shows the **Business Partner Category** available condition field, which you can also use.

#### FI\_ACCKNVK SAP ILM Object

The FI\_ACCKNVK SAP ILM object for the contact person does not provide any fields besides **application Name** and **application Rule Variant**.

Finally, let's talk about the **Company Code** selectable condition field with customers and vendors. As you saw in detail in Chapter 4, Section 4.4.3, blocking customers and vendors is also possible at the company code level. Figure 4.44 in the book shows an input field with the same name. (This is not possible for the contact person and the central business partner.) Therefore, each application participating in the EoP check must indicate whether it supports this. The application makes this known in Customizing, as shown in [Figure 20](#). If the **BUKRS** column (Relevance for End of Purpose at Bukrs Level) is set, it



means that the application also executes the EoP check on the company code level. In these cases, you have the option of blocking the master data for certain company codes only, if required.

**ILM Policies**

Continue New

\* Policy Category: Residence Rules

\* Object Category: SAP Business Suite

Audit Area: BUPA\_DP Business Partner D...

ILM Object: CA\_BUPA

Policy Category Residence Rules Audit Area Business Partner Data Privacy Object Category SAP Business Suite Object Business Partners

Save Edit Rules Back

Policy Name:

**Available Condition Fields**

Field Name	Description
APPL_NAME	Application Name
APPL_RULE_VARIANT	Application Rule Variant
BP_TYPE	BP Category

**Selected Condition Fields (Maximum of 4)**

Field Name	Description

Figure 19 Other Available Condition Fields (Business Partner)

### Note: Company Code-Dependent and Nondependent Master Data

The central business partner and contact person data is not company code-dependent, so the previous information only applies to customers and vendors.

Blocking and Unblocking of Data

- Business Partner
- Customer Master/Vendor Master Deletion
- Registration of Applications
  - Register Application Names for EoP Check
  - Register Application Classes for EoP Check**
  - Register Application With Different Software Layer

**Change View "Register Application classes": Overview**

New Entries

Dialog Structure

- Register Application classes
- Define priorities

Register Application classes

ID	Type	Appl Name	Item Num...	Application Description	Registered class for EoP checks	General	Comp. Code
Customer mas...	AUDIT_MGMT_ERP_CUS	0	Audit Management - Object Refere	CL_IAM_CUS_VEN_EOP_CHECK	✓	✓	
Customer mas...	CACS_CUSTIVEND	0	ICM Application for customer	CL_CACS_EOP_CHECK_CUST	✓	✓	
Customer mas...	CRM_FEV	0	Period End Valuation	CL_CVP_FEV_EOP_CHECK	✓	✓	
Customer mas...	CRM_IF	0	CRM Middleware	CRM_IF_CMD_EOP_CHECK	✓	✓	
Customer mas...	CVI	0	Customer vendor integration	CVI_CMD_EOP_CHECK	✓	✓	
Customer mas...	EHS_PS_REP	0	EHS Product Safety, Report Ship	CL_EHS_REPSHIP_CUST_EOP_CHECK	✓	✓	
Customer mas...	EHS_WA_MD	0	EHS Waste Management, Master	CL_EHSW_CV_EOP_CHECK_MD	✓	✓	
Customer mas...	ERP_ALLOC	0	Allocation Table	CL_ALLOCATION_CV_EOP_CHECK	✓	✓	
Customer mas...	ERP_CON	0	Consent Management	CL_CM_CVP_EOP_CHECK	✓	✓	
Customer mas...	ERP_CUST	0	ERP Customer Master	CVP_CL_EOP_CHECK_MD_CUST	✓	✓	

Figure 20 Customers and Vendors: Application Execution of EoP Check at Company Code Level

What is the consequence of defining the residence rules for master data in Transaction IRMPOL (SAP ILM rules)? If the application also executes the EoP check on the company code level, you can use the **Company Code** field for rule maintenance for this application, in addition to the fields mentioned so far, such as **Application Rule Variant** and **Application Name**. This applies to the rules related to both the residence period and the retention period. [Figure 21](#) shows an example.

Rules for Policy TEST_BUKRS								
	Application Name	Application Rule Var	Company Code	To	Residence Period	Residence Prd Unit	Time Ref.	Time Offset
	ERP_CUST				6	Month	Start of Retention Period	End of Month
	ERP_FI		1000		2	Year	Start of Retention Period	End of Year
	ERP_FI		2000		1	Year	Start of Retention Period	End of Year

**Figure 21** Defining Residence Rules for Company Code-Dependent Master Data

## Further Notes on Destroying Master Data

### Note

This section expands coverage of Section 4.3.6.

If the ERP\_CVP\_ILM\_1 business function (SAP ILM-based deletion of customer and vendor master data) is active, you can only archive and delete customer/vendor master data if the data has already been blocked. However, there are certain master data types that cannot be blocked (e.g., one-time customer accounts [often referred to as CpD accounts]) or that can only be blocked if the entire master data has been blocked (e.g., sales organization [customer] or purchasing organization [vendor]).

As soon as you have imported SAP Note 2915312 and the SAP Notes mentioned therein, you nevertheless can archive and delete one-time customer accounts and sales organizations or purchasing organizations that include them. This was previously impossible. You can use Transaction CVP\_ARCH\_WHITEL (Define Whitelist for Archiving) to do this. You can include one-time customer accounts in this whitelist, for example. The list is considered during data archiving, and data archiving is enabled.

For more information about the newly introduced process switches that are offered for simplified blocking and deletion of customers and vendors, see, for example, SAP Note 2779658. These switches are referred to in SAP Note 2915312.

## Note on Improved Messages in the Application Log

### Note

This section expands coverage of Section 4.4.3.

The log displayed in scenario 3 of Section 4.4.3 also contains a **Long Text** column, which contains further details related to the message. As of the corrections contained in SAP Note 2908849 (Check for End of Purpose—Improvement of Messages in the Application Log), the long texts have been improved. They now contain details such as the next check date or the start of the retention period for customers, vendors, or contact persons that you want to block. If there is a long text, you will see an icon in the **Long Text** column with a question mark symbol that you can click on.

## Table Fields for Storage of Blocking Information

### Note

This section also expands coverage of Section 4.4.3.

After reading our book up to Chapter 4, Section 4.4.3, you know how master data can be blocked. You could now ask in which table fields this information is stored. We will provide a few examples here.

The following fields are set to the value **X**:

- CVP\_XBLCK (flag for fulfilled business purpose) and LOEVM (central deletion flag for master record) in table LFA1 (vendor master, general part) and KNA1 (customer master, general part)
- XDELE (central archiving flag) and XPCPT (flag for fulfillment of purpose) in table BUT000 (BP: General Data I)

Furthermore, the table fields related to an authorization group are set as follows:

- For the central business partner (BUT000-AUGRP), to the value entered in Figure 4.6 in the book
- For the customer or vendor (KNA1-BEGRU or LFA1-BEGRU), to the value entered in Figure 4.7 in the book

What about the distribution (replication) of these fields in your system landscape?

In SAP Business Suite, the previously mentioned fields are used for the customer and vendor as follows:

- As part of the direct communication within the EoP check. Here the distribution (only) to the directly connected systems takes place. You have specified these in Customizing (as shown in Figure 4.11 in the book).
- Additionally distributed via the “normal” master data synchronization (via ALE) for the customer/vendor via IDoc.

In contrast, table field BUT000-XPCPT for the central business partner is only distributed as part of the direct communication within the EoP check. Here the distribution is (only) to the directly connected systems. You have specified these in Customizing (as shown in Figure 4.10 in the book). There is no additional distribution via the “normal” master data synchronization as in SAP ERP for the customer/vendor via IDoc.

In SAP S/4HANA, the customer and vendor are part of the central business partner. This means that the noted table fields are distributed via the business partner (using CVI). The distribution of these table fields outside the EoP check is no longer necessary.

Finally, let’s discuss the *start of retention time* tables:

- BUTSORT for business partners
- CVP\_SORT for the customer/vendor/contact

These tables contain the start date for the retention period for a blocked master data record. (*SORT* stands for *start of retention time*.) The start of the retention period is the EoB time in the master data. The date is calculated during the blocking of master data.

Why is it necessary to write the EoB date into the tables? In general, we have explained that the start date is calculated using the reference date you enter in a retention rule (e.g., the clearing or change date of a document). However, the situation is slightly different for master data. These may only be destroyed once all the related transactional data have been destroyed. You must determine the retention period of the master data—that is, the longest retention period for the corresponding transactional data—yourself in Transaction IRMPOL using the double maintenance function. (The application rule

variants help you to reduce the effort and increase the consistency of the rules because you can use the rule groups to refer to the same retention periods for the transactional data and the corresponding master data).

Let's now look at the starting date for this retention period. As you will remember, you must enter the **Start of Retention Period** value as a time reference in both the retention and residence rules for master data (shown in Figure 4.23 in the book). The start of the retention period is determined during the EoP check—that is, when a master data record is blocked. The individual associated transactional data record provides its start of the retention period. For each combination of the application name and the application rule variant, the date furthest to the right on the time stream is written to the tables mentioned as the start of the retention period.

In other words, you have to ensure that the retention period for master data is correctly specified in your rules. The correct start date will be calculated for you, as just discussed, as soon as you enter the **Start of the Retention Period** as the time reference.

If you want to know more about the purpose of the tables, we can mention that they also contain information about the application names and application rule variants collected during the EoP check.

It should also be noted that the tables belong to the structure definition of the respective master data archiving object. This means that when the data object is created in the archive file—that is, when data is written to an archive file (regardless of whether the SAP ILM action is Archiving or Data Destruction)—SAP ILM receives all the information it needs to calculate the retention period of the data.

In other words, these tables are the starting point for the Information Retention Manager to determine which rules should be applied for the master data in question—in particular, which application rule variants and which application names should be taken into account. Additional fields also come from the actual master data tables. For example, for the customer, the account group from table KNA1 (customer master, general part). The Information Retention Manager can use all these fields to determine which of the rules you have defined should be processed.

As you now know, the entries in these tables are created when the master data is blocked. What happens if you change your policies after that?

You can change the rules for a rule group at any time. You can also change the assignment of rule groups to application rule variants. However, you should not change the names of the application rule variants because these names have already been entered into the tables. Accordingly, you also should not introduce any further application rule variants that are supposed to apply to master data that has already been blocked because they did not yet exist at the time of blocking and the system cannot take them into account.

#### Unrecommended Changes

A concrete example of this would be if you had rules for the ERP\_CUST (ERP Customer Master) and ERP\_FI (ERP Financial Accounting) application names when you blocked a customer for the FI\_ACCRECV SAP ILM object (customer master data). After blocking, you add rules for the ERP\_SD application name (Sales and Distribution) to the policy. These rules are only considered for master data to be blocked in the future. If they were also considered for data that has already been blocked, you would have to unblock the customer and then block it again.

### Further Consequences of Unblocking Master Data in a Business Process

#### Note

This section expands coverage of Section 4.4.6.

Section 4.4.6 of our book provided information on unblocking master data in a business process. In this section, we will add a few notes about the consequences of unblocking master data.

After you have blocked the central business partner, the table field for the authorization group (BUT000-AUGRP) contains the value entered in Figure 4.6 in the book. After you have blocked customers or vendors, the table field for the authorization group (KNA1-BEGRU or LFA1-BEGRU) contains the value entered in Figure 4.6 in the book.

If you have now unblocked the master data using the information in this section, the table field for the authorization group is reset to the value it had before blocking. This is carried out using the change documents. It is important, therefore, that they are available and have not been archived.

Certain flags that were set when blocking are also reset to blank when unblocking—for example:

- CVP\_XBLCK (flag for fulfilled business purpose) and LOEVM (central deletion flag for master record) in table LFA1 (vendor master, general part) and KNA1 (customer master, general part)
- XDELE (central archiving flag) and XPCPT (flag for fulfillment of purpose) in table BUT000 (BP: General Data I).

In this context, when unblocking from the central business partner, please refer to SAP Notes 2903500 and 2909515.

## Chapter 9

This section will cover topics from Chapter 9 of the book that have been expanded since publication.

### Attachments

#### Note

This section expands coverage of Section 9.2.

In our book, you learned how to enable an archiving object for SAP ILM. Let's discuss here whether you, as the developer of the archiving object, have to take something into account in the archive write or delete program if your business object has attachments and these are supposed to be archived or destroyed correctly. Consequently, this chapter is only relevant for you if such attachments exist.

We'll differentiate between the types of attachments that SAP ILM supports:

- GOS
- SAP ArchiveLink
- Document Management System (DMS)

The handling of these attachments from the perspective of the user who destroys (or in rare cases archives) them is described in [Destruction of Attachments](#).



For the GOS and SAP ArchiveLink attachments there is no work for you as a developer. Only if you want your archiving object to archive the GOS attachments as well—and use archiving class `CV_OBJL` (document info record object links) for this—must you include archiving class `SGOS` in your archiving object.

Let's now discuss DMS. Because archiving object `CV_DVS` (document management) is SAP ILM-enabled, the developers of the archiving object are asked by the leading object (potentially you) to use the `CV_OBJL` archiving class correctly. In particular, when the `CV_OBJL_ARCHIVE_OBJECT` function module is prompted, you must set the `OBJECT_DELETE_FLAG` parameter to `ABAP_FALSE`. This is the only way to ensure that the link information (DRAD\* tables) in the delete program is *not* deleted from the database (which would otherwise be the typical behavior in a delete program). This is important because this information is technically necessary to determine the correct retention period for the document info records using the `CV_DVS` SAP ILM object (document management). If the link information is not (or is no longer) available, only rules can be processed for which the **Linked Object** column is set to a value that indicates that *no* linked object exists.

Finally, let's discuss the `CL_ODATA_CV_ATTACHMENT_API` class and its `IF_ODATA_CV_ATTACHMENT_API~DELETE_ATTACHMENT` interface. The use of such methods is not recommended because they delete data without checking the retention period in SAP ILM. They bypass them, so to speak. The `CV_DVS` archiving object (document management) should be used instead.

## New Object Directory Entry for Data Destruction Objects

### Note

This section also expands coverage of Section 9.2.

Note that, unlike the archiving object, the data destruction object did not have an object directory entry at the beginning (table TADIR [Directory of Repository Objects], Transaction SOBJ [Maintenance Object Attributes]). The settings for a data destruction object were transported by transporting the corresponding table entries. The object directory entry is available as of releases SAP BASIS 7.55 SP 00 and SAP S/4HANA 2020 FPS 0. Its exact name is `R3TR DOBJ<data destruction object>`.

## Destruction of Attachments

### Note

This section expands coverage of Section 9.4.2.

Section 9.4.2 of our book introduced the development of data destruction software. However, a few functions that affect attachments have been added in the meantime:

#### ■ GOS systems

To destroy GOS attachments, you must run the `DELETE_AL_ITEM` method (GOS API: Delete an Attachment) of the `CL_GOS_API` class (GOS API: Access GOS Attachment List for External Users; see [Figure 22](#)).

Class/Interface		CL_GOS_API		Implemented / Active	
Properties	Interfaces	Friends	Attributes	Methods	Events
Parameters of Method		DELETE_AL_ITEM			
←	Methods	Exceptions	Sourcecode	Properties	
Parameter	Type	P...	O...	Associated Type	Description
IS_ATT_KEY	Importing	<input type="checkbox"/>	<input type="checkbox"/>	GOS_S_ATTKEY	GOS API: Key Structure of Attachment
RV_COMMIT	Returning	<input checked="" type="checkbox"/>	<input type="checkbox"/>	FLAG	Commit required

Figure 22 DELETE\_AL\_ITEM Method of CL\_GOS\_API Class

#### ■ SAP ArchiveLink systems

To destroy SAP ArchiveLink attachments, you must run the `ARCHIV_DELETE_META` function module (SAP ArchiveLink: Delete Links and Stored Documents; see [Figure 23](#)).

#### ■ DMS

At present, no application (no business object) is known that has both a data destruction object and DMS attachments. Therefore, no functionality is available here yet.

Function Module		ARCHIV_DELETE_META		active	
Attributes	Import	Export	Changing	Tables	Exceptions
Source Code					
Parameter Name	Typing	Associated Type	Defa...	Opt...	Pas...
ARCHIV_ID	LIKE	TOAAR-ARCHIV_ID	SPACE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
ARC_DOC_ID	LIKE	SAPB-SAPADOKID	SPACE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
AR_OBJECT	LIKE	TOAOM-AR_OBJECT	SPACE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
				Short Text	
				Archive ID	
				Archive document ID	
				Object type to be archived	

Figure 23 ARCHIV\_DELETE\_META Function Module

## New Object Directory Entry for SAP ILM Objects

### Note

This section expands coverage of Section 9.5.2.

Note that, unlike the archiving object, the SAP ILM object did not have an object directory entry at the beginning (table TADIR [Directory of Repository Objects], Transaction SOBJ [Maintenance Object Attributes]). The settings for an SAP ILM object were transported by transporting the corresponding table entries. The object directory entry is available as of releases SAP\_BASIS 7.55 SP 00 and SAP S/4HANA 2020 FPS 00. The exact name for SAP ILM objects from the SAP Business Suite object category is R3TRILMB<SAP ILM object>.

## Settings for the Destruction of Attachments

### Note

This section expands coverage of Section 9.5.9.

The settings we will describe here are necessary for the correct operation of the functionality discussed in [Destruction of Attachments](#). They are not necessary, however, if the users of your SAP ILM object only use the AL\_DOCUMENTS SAP ILM object (documents stored via SAP ArchiveLink) to provide the SAP ArchiveLink attachments with the retention period.

Conversely, the settings are necessary if the leading object for which you create the SAP ILM object has GOS attachments or has SAP ArchiveLink attachments for which the AL\_DOCUMENTS SAP ILM object is not supposed to be used (or not for all attachments).

From a technical point of view, these settings are also only necessary if your SAP ILM object is linked to an archiving object. If it is linked to a data destruction object, the settings are not necessary. However, we recommend them so that end users can see which SAP ILM objects support the destruction of attachments.

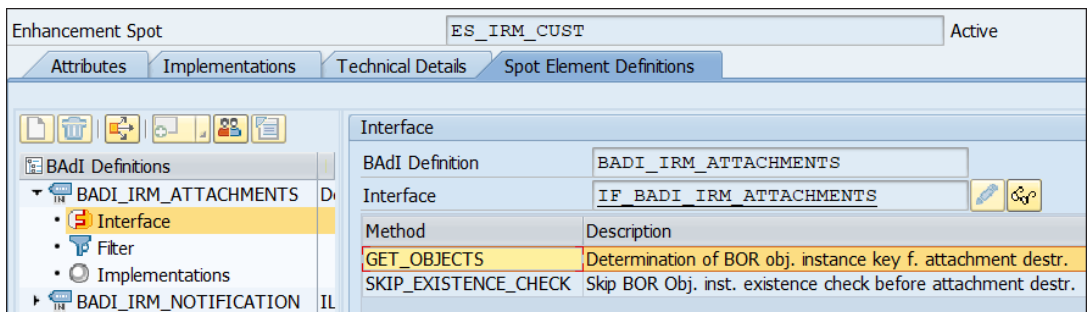
In principle, your task consists of only two steps: you must tell SAP ILM what the associated BOR object type is and how its key is formed.

To make the BOR object type known, you can use the new **Mapping BOR Object Type for Destroying Attachments** entry in Transaction IRM\_CUST\_BS (IRM Customizing, SAP Business Suite; screen already shown in [Figure 7](#)).

The **Determination of Object Instance Key for BOR Obj.** entry below is used to specify the key of the BOR object type. You are already familiar with this entry because it is also used to disclose the key for legal hold.

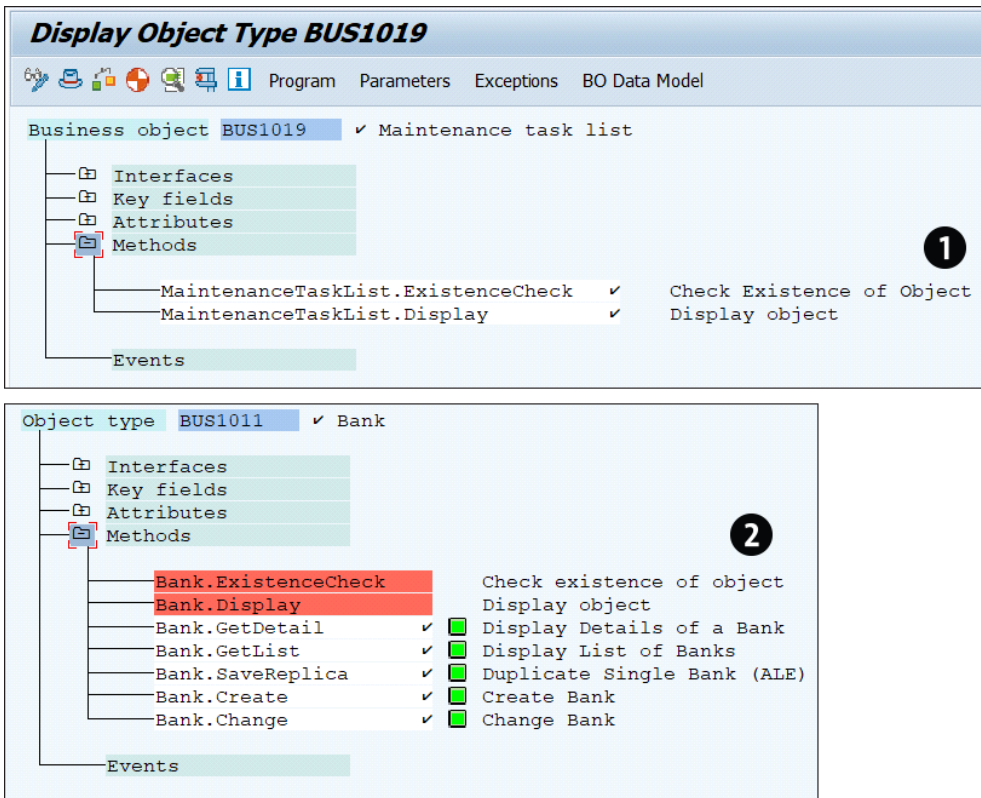
In principle, all necessary settings are already made. Let's now turn our attention to special cases so that you can assess whether they apply to you and need to be taken into account.

Let's start with a case in which several BOR objects are assigned to the SAP ILM object and these have different key definitions. In this case, it is not possible to specify all key definitions for all BOR objects when using **Determination of Object Instance Key for BOR Obj.** In such cases, you as the developer of the SAP ILM object should define the key definition for each BOR object in the implementation of the GET\_OBJECTS method of the BADI\_IRM\_ATTACHMENTS BAdI (Determination of the BOR Object Instance Key for Destroying Attachments). You can see this in [Figure 24](#).



**Figure 24** BADI\_IRM\_ATTACHMENTS BAdI: GET\_OBJECTS and SKIP\_EXISTENCE\_CHECK Methods

Let's look at the second method, SKIP\_EXISTENCE\_CHECK, in the same figure, which represents a second special case. We must first discuss the role of the ExistenceCheck method in the definition of the BOR object type in Transaction SWO1 (Business Object Builder). You can see this in [Figure 25](#), where ❶ shows that the method has been implemented, and ❷ shows that it has not been.



**Figure 25** Definition of the BOR Object Type: ExistenceCheck Method

In this optional method of a BOR object, the GOS framework checks whether the leading object (e.g., a maintenance plan) for an attachment that is currently supposed to be destroyed exists in the database or not. If the method has not been implemented, this means that the GOS attachments have not been destroyed.

Note that the SAP ArchiveLink framework does not execute this check. If your leading object does not have any GOS attachments, you do not need to take into account what has just been discussed. Otherwise, as the developer of the associated SAP ILM object, you must agree with the developer of the BOR object that he or she will implement the `ExistenceCheck` method subsequently. If this is not possible for certain reasons (side effects), you must implement the `SKIP_EXISTENCE_CHECK` method as shown in [Figure 24](#). This stops the GOS framework from destroying the attachment without first checking whether or not the leading object (such as the maintenance plan) for the attachment exists in the database.