



CUSTOMER

# **Operations Guide**

## SAP ECTR for SAP S/4HANA 1.1

# DOCUMENT HISTORY



## Note

Before you start the implementation, make sure you have the latest version of this document.

This guide is valid as of SAP ECTR for SAP S/4HANA 1.1.7.0

The latest version is available on SAP Help Portal

[https://help.sap.com/docs/SAP\\_ENGINEERING\\_CONTROL\\_CENTER\\_S4HANA](https://help.sap.com/docs/SAP_ENGINEERING_CONTROL_CENTER_S4HANA)

Version	Date	Change
1.0	2019-05-29	Initial creation
2.0	2019-07-26	<ul style="list-style-type: none"> <li>• "Structure of Element &lt;material_create&gt;" section optimized</li> <li>• "Structure of Element &lt;material_master_template&gt;" section added</li> <li>• "Structure of Element &lt;checkout_rules&gt;" section optimized</li> <li>• "Structure of Element &lt;first_level_support&gt;" section optimized</li> <li>• "Parameter Maps" section optimized</li> <li>• "Description of the Attribute Mapping File" section optimized</li> </ul>
3.0	2019-11-21	<ul style="list-style-type: none"> <li>• "Structure of Element &lt;material_master_template&gt;" section optimized</li> <li>• "Description of the Attribute Mapping File" section optimized</li> <li>• "Structure of Element &lt;program options&gt;" section optimized</li> <li>• New "Structure of Element &lt;status_transition&gt;" section added</li> <li>• New "Structure of Element &lt;status_transition_restriction&gt;" section added</li> <li>• "Description of the Attribute Mapping File" section optimized</li> <li>• "Structure of Element &lt;material_create&gt;" section optimized</li> <li>• "Structure of Element &lt;material_master_template&gt;" section optimized</li> <li>• "Default Input for Parameters" section optimized</li> <li>• "MacroState" section added</li> <li>• "Description of the Attribute Mapping File" section optimized</li> <li>• "Structure of Element &lt;program_options&gt;" section optimized</li> </ul>

Version	Date	Change
4.0	2020-07-02	<ul style="list-style-type: none"> <li>• "Smartcontainers and ABAP modules" section optimized</li> <li>• "Using the JPO Repository" section optimized</li> <li>• "Structure of Element &lt;nonmaster_options&gt;" section optimized</li> <li>• "Description of the Attribute Mapping File" section optimized</li> <li>• "Nest Smartcontainers into Each Other" section added</li> <li>• Note added to "Attribute Transfer Between the SAP System and an Application" section</li> <li>• "Keylist Operations" section optimized</li> <li>• "Calling OMF Functions" section optimized</li> <li>• "Editability of Class Types and Characteristics" section added</li> <li>• New "Execute Macro with SAP GUI Transaction" section added</li> <li>• New Note added to "New Search" section</li> </ul>
5.0	2021-04-29	<ul style="list-style-type: none"> <li>• "Structure of Element &lt;document_create&gt;" section optimized</li> <li>• "Structure of Element &lt;primary_application&gt;" section optimized</li> <li>• "Front-End Adjustments" section optimized</li> <li>• "Creating SmartLists" section optimized</li> <li>• "Searching, Filtering" section optimized</li> <li>• "Optimization of RFC Data Transfer" section optimized</li> <li>• "Structure of Element &lt;document_create&gt;" section optimized</li> <li>• "Keylist Operations" section optimized</li> <li>• "Structure of Element &lt;material_create&gt;" section optimized</li> <li>• "Macro Editor" section optimized</li> <li>• New "PLM Observer Sleep Mode" section added</li> <li>• "Supported XML Tags" section optimized</li> <li>• "Rules for Creating New Documents" section updated</li> </ul>
6.0	2021-07-08	<ul style="list-style-type: none"> <li>• "Searching, Filtering" section updated</li> <li>• "Object Properties" section updated</li> <li>• Example in "Supported XML Tags" section optimized</li> <li>• "Structure of the URL" section optimized</li> <li>• "document_create Attribute" section updated</li> </ul>

Version	Date	Change
7.0	2022-07-14	<ul style="list-style-type: none"> <li>• Added a note to chapter „Attribute transfer between SAP system and an application“</li> <li>• „Description of the attribute mapping file“ optimized</li> <li>• „Adjustments in the front end“ optimized</li> <li>• CDESK_BOM_WIZARD deleted</li> <li>• „Calling functions“ optimized</li> <li>• New chapter „Start environment“ added</li> <li>• “DSCSAG/ Business Add Ins” optimized</li> <li>• “DSCSAG/DOCUMENT_02” optimized</li> <li>• “Attribute transfer between SAP system and an application” optimized</li> <li>• New chapter “replace_primary_file attributes”</li> <li>• Chapter “UI Theme” added</li> <li>• „Description of the attribute mapping file“ optimized</li> <li>• „Structure of Element &lt;nm_dtype_change&gt; “ optimized</li> <li>• “Calling functions“ enhanced</li> </ul>
8.0	2022-10-20	<ul style="list-style-type: none"> <li>• "Structure of Element &lt;document_create&gt;" section optimized</li> </ul>
9.0	2022-12-08	<ul style="list-style-type: none"> <li>• Chapter "Vadility check" optimized</li> </ul>
10.0	2023-10-23	<ul style="list-style-type: none"> <li>• "Structure of Element &lt;copy_as&gt;" section optimized</li> <li>• "Structure of Element &lt;new_version&gt;" section optimized</li> <li>• “User entries” section optimized</li> <li>• "Configure UI theme" optimized</li> <li>• "Rules for Part File Names" optimized</li> <li>• “Default Input for Parameters” optimized</li> <li>• "Structure of Element &lt;program_options&gt;" section optimized</li> <li>• Chapter " Response of SAP ECTR for SAP S/4HANA after Starting the Primary Application" opzimized</li> </ul>

# TABLE OF CONTENTS

<b>1</b>	<b>INTRODUCTION</b> .....	<b>9</b>
1.1	ABOUT THIS DOCUMENT .....	9
1.2	GLOBAL DEFINITIONS .....	9
1.3	IMPORTANT SAP NOTES .....	9
<b>2</b>	<b>TECHNICAL SYSTEM LANDSCAPE</b> .....	<b>11</b>
2.1	DOCUMENTATION .....	11
<b>3</b>	<b>FILE HANDLING</b> .....	<b>13</b>
3.1	HANDLING OF THE MAIN ORIGINAL IN AN APPLICATION .....	13
3.1.1	VALIDITY CHECK .....	14
3.1.2	MASTER DOCUMENTS .....	15
3.1.3	CONTENT VERSIONS IN THE SAP SYSTEM AND TROUBLESHOOTING .....	15
3.2	WORKING IN INDIVIDUAL FILES .....	15
3.2.1	ADDITIONAL FILES .....	15
3.2.2	VIEWABLES .....	16
3.3	WORKING WITH MULTIPLE SESSION DIRECTORIES .....	17
3.4	HANDLING OF MAIN ORIGINALS .....	17
<b>4</b>	<b>ATTRIBUTE TRANSFER BETWEEN THE SAP SYSTEM AND AN APPLICATION</b> .....	<b>18</b>
4.1	ATTRIBUTE TRANSFER FROM THE SAP SYSTEM TO AN APPLICATION .....	18
4.1.1	DESCRIPTION OF THE ATTRIBUTE MAPPING FILE .....	18
4.2	ATTRIBUTE TRANSFER FROM AN APPLICATION TO THE SAP SYSTEM .....	35
4.2.1	SUPPORTED XML TAGS .....	35
4.2.2	DRAWING-SPECIFIC ATTRIBUTES .....	38
<b>5</b>	<b>SYSTEM COMPONENT SAP ECTR FOR SAP S/4HANA</b> .....	<b>39</b>
5.1	ROLES IN SAP ECTR FOR SAP S/4HANA .....	39
5.2	RULES FOR PART FILE NAMES .....	39
5.3	FILE HANDLING .....	40
5.4	LOG FILES .....	40
5.4.1	TRACE OUTPUT .....	40
5.4.2	BAPI LOG FILES .....	40
5.5	DICTIONARY .....	40
5.6	MATERIAL .....	41
5.6.1	CREATING A MATERIAL .....	41
5.6.2	CREATING A BILL OF MATERIAL .....	41
5.7	DTYPES .....	42
5.7.1	STRUCTURE OF ELEMENT <DTYPE> .....	43
5.7.2	STRUCTURE OF ELEMENT <ADDITIONAL_FILES> .....	45
5.7.3	STRUCTURE OF ELEMENT <ALLOWED_ACTIONS> .....	46
5.7.4	STRUCTURE OF ELEMENT <ALLOWED_NONMASTER> .....	47
5.7.5	STRUCTURE OF ELEMENT <CHANGE_STATUS> .....	47
5.7.6	STRUCTURE OF ELEMENT <CHECKIN_RULES> .....	48
5.7.7	STRUCTURE OF ELEMENT <CHECKOUT_DIRECT> .....	49
5.7.8	STRUCTURE OF ELEMENT <CHECKOUT_RULES> .....	50
5.7.9	STRUCTURE OF ELEMENT <CONTAINER_FILE> .....	52
5.7.10	STRUCTURE OF ELEMENT <COPY_AS> .....	53
5.7.11	STRUCTURE OF ELEMENT <DELETE_IN_SAP> .....	55
5.7.12	STRUCTURE OF ELEMENT <DELETE_IN_SESSION> .....	55
5.7.13	STRUCTURE OF ELEMENT <DESCRIPTION> .....	55
5.7.14	STRUCTURE OF ELEMENT <DOCUMENT_CREATE> .....	56

5.7.15	STRUCTURE OF ELEMENT <FIRST_LEVEL_FILES>.....	61
5.7.16	STRUCTURE OF ELEMENT <ICON>.....	62
5.7.17	STRUCTURE OF ELEMENT <ICONDEF>.....	62
5.7.18	STRUCTURE OF ELEMENT <IGNORE>.....	63
5.7.19	STRUCTURE OF ELEMENT <IGNORE_FILES>.....	63
5.7.20	STRUCTURE OF ELEMENT <INSERT_ALLOWED>.....	64
5.7.21	STRUCTURE OF ELEMENT <MATERIAL_CREATE>.....	64
5.7.22	STRUCTURE OF ELEMENT <MATERIAL_MASTER_TEMPLATE>.....	65
5.7.23	STRUCTURE OF ELEMENT <NEW_VERSION>.....	66
5.7.24	STRUCTURE OF ELEMENT <NM_DTYPE_CHANGE>.....	68
5.7.25	STRUCTURE OF ELEMENT <NONMASTER_OPTIONS>.....	68
5.7.26	STRUCTURE OF ELEMENT <PARAM>.....	70
5.7.27	STRUCTURE OF ELEMENT <PARAMS>.....	71
5.7.28	STRUCTURE OF ELEMENT <PLMFILE>.....	71
5.7.29	STRUCTURE OF ELEMENT <PRIMARY_APPLICATION>.....	73
5.7.30	STRUCTURE OF ELEMENT <PROGRAM_OPTIONS>.....	75
5.7.31	STRUCTURE OF ELEMENT <REPLACE_PRIMARY_FILE>.....	78
5.7.32	STRUCTURE OF ELEMENT <SAVE_DIRECT>.....	79
5.7.33	STRUCTURE OF ELEMENT <SAVE_LOCAL>.....	79
5.7.34	STRUCTURE OF ELEMENT <SCRIPT>.....	80
5.7.35	STRUCTURE OF ELEMENT <SCRIPTS>.....	81
5.7.36	STRUCTURE OF ELEMENT <SEEDFILE>.....	81
5.7.37	STRUCTURE OF ELEMENT <STATUS_TRANSITION>.....	85
5.7.38	STRUCTURE OF ELEMENT <STATUS_TRANSITION_RESTRICTION>.....	85
5.7.39	STRUCTURE OF ELEMENT <STORE_FILES>.....	86
5.7.40	STRUCTURE OF ELEMENT <VIEW_FILES>.....	86
5.7.41	USING TEMPLATES FOR DTYPE DEFINITIONS.....	87
5.7.42	ACCESS TO ENVIRONMENT AND PREFERENCE VARIABLES.....	89
5.7.43	RULES FOR CREATING NEW DOCUMENTS.....	90
5.8	AUTOMATED CONFIGURATION CHECK.....	91
5.9	EDITABILITY OF CLASS TYPES AND CHARACTERISTICS.....	92
<b>6</b>	<b>MACROS, SMARTLISTS AND SMARTCONTAINER.....</b>	<b>93</b>
6.1	PURPOSE OF USE/AREA OF APPLICATION.....	93
6.2	USING MACROS IN SAP ECTR FOR SAP S/4HANA.....	93
6.2.1	CREATING MACROS.....	93
6.2.2	FUNCTIONS.....	94
6.2.2.1	<i>Macro Editor</i> .....	94
6.2.2.2	<i>Displaying Macros from the Configuration Folder</i> .....	95
6.3	DEFINITIONS OF BASIC TECHNOLOGY.....	95
6.3.1	KEYLIST.....	95
6.3.2	OBJECTSET.....	96
6.3.3	PARAMETER MAPS.....	96
6.4	CREATING SMARTLISTS.....	96
6.4.1	NEW SEARCH.....	96
6.4.2	NEW QUERY WITH SMARTLIST.....	97
6.4.3	SMARTFOLDER.....	98
6.5	CREATING SMARTCONTAINERS.....	98
6.5.1	FRONT-END ADJUSTMENTS.....	99
6.5.2	SMARTCONTAINERS AND ABAP MODULES.....	101
6.5.3	NESTING SMARTCONTAINERS INTO EACH OTHER.....	102
6.6	CREATING FUNCTIONS.....	103
6.7	AVAILABLE FUNCTIONS IN THE MACRO EDITOR.....	103

6.7.1	KEYLIST OPERATIONS.....	103
6.7.2	SET OPERATIONS.....	104
6.7.3	DISPLAY FUNCTIONS.....	106
6.7.4	PARAMETER MAPS.....	107
6.7.4.1	<i>Default Input for Parameters.....</i>	<i>108</i>
6.7.5	USER ENTRIES.....	113
6.7.6	SEARCHING, FILTERING.....	115
6.7.7	CALLING FUNCTIONS.....	117
6.7.8	OBJECT PROPERTIES.....	118
6.7.9	ENVIRONMENT VARIABLES.....	121
6.7.10	FILE OPERATIONS.....	122
6.7.11	DICTIONARY-ENTRIES.....	123
6.7.12	MACROSTATE.....	123
6.8	PROCESSABLE, STATIC VARIABLES IN MACROS.....	124
6.9	EXECUTE MACRO WITH SAP GUI TRANSACTION.....	125
6.10	CREATE SYSTEM SMARTLISTS.....	125
6.10.1	CREATE SYSTEM SMARTLISTS.....	125
6.10.2	CONFIGURE SYSTEM SMARTLISTS.....	125
6.10.3	PREDEFINED PARAMETERS.....	126
6.11	ADDING OBJECTS TO A FOLDER USING A SEARCH MACRO.....	126
<b>7</b>	<b>ADDITIONAL TECHNICAL INFORMATION.....</b>	<b>128</b>
7.1	URL ACCESS TO DOCUMENTS AND MATERIALS IN THE SAP SYSTEM.....	128
7.1.1	STRUCTURE OF THE URL.....	128
7.1.2	CONTEXT MENU FUNCTION FOR URL GENERATION.....	129
7.2	START ENVIRONMENT.....	130
7.3	PROCESSING PREFERENCE VARIABLES.....	131
7.4	ADDITIONAL SETTINGS.....	132
7.4.1	RESPONSE OF SAP ECTR FOR SAP S/4HANA AFTER STARTING THE PRIMARY APPLICATION 132	
7.4.2	STARTING THE PRIMARY APPLICATION.....	133
7.4.3	STARTING THE APPLICATION FOR ADDITIONAL ORIGINALS.....	133
7.5	UI THEME.....	138
<b>8</b>	<b>/DSCSAG/ BUSINESS ADD-INS.....</b>	<b>139</b>
8.1	ADVOBJ.....	139
8.2	CHANGE.....	139
8.3	CLASS.....	139
8.4	/DSCSAG/DOCUMENT_01.....	139
8.5	/DSCSAG/DOCUMENT_02.....	140
8.6	/DSCSAG/FOLDER_01.....	142
8.7	/DSCSAG/MATERIAL_01.....	142
8.8	/DSCSAG/MM_01.....	143
8.9	/DSCSAG/PLM_01.....	144
8.10	/DSCSAG/UTILS.....	144
8.10.1	/DSCSAG/ DOC_CHECK_AUTH.....	144
8.10.2	/DSCSAG/ DOC_SEARCH.....	144
8.11	ENHANCEMENT SPOT /DSCSAG/PLM_ACC.....	144
8.12	/DSCSAG/PLM_ACC_CNTX.....	145
<b>9</b>	<b>SAP TABLES: DESCRIPTION AND CONTEXT.....</b>	<b>146</b>
9.1	DRAW.....	146
9.1.1	DATA MODEL.....	146

9.1.2	RES4 FIELD IN TABLE DRAW .....	146
9.2	CDESK_DRAW .....	147
9.2.1	IMPORTANT FIELDS IN TABLE CDESK_DRAW .....	147
9.2.2	ADDITIONAL FIELDS IN CDESK_DRAW .....	148
9.2.3	FIELDS (PERFORMANCE FIELDS) IN /DSCSAG/DOC_RED.....	148
9.2.4	RELATIONSHIP TYPES/SORT FIELD IN DOCUMENT BOM ITEM .....	150
9.2.5	FOLDER CONTENTS IN THE SAP ECTR FOR SAP S/4HANA DESK IN /DSCSAG/FOL_ALC .....	151
9.3	/DSCSAG/PREF.....	152
9.4	/DSCSAG/CHECK.....	152
9.5	/DSCSAG/DOC_RED_UPDATE .....	152
9.5.1	/DSCSAG/PLM_DOC_VER_BUFFER_FIL.....	153
<b>10</b>	<b>PERFORMANCE OPTIMIZATION .....</b>	<b>154</b>
10.1	BASES FOR OPTIMIZATION IN WAN INSTALLATIONS .....	154
10.1.1	MEASURES FOR OPTIMIZATION .....	154
10.1.2	SAP API CALLS.....	154
10.1.3	OPTIMIZATION OF RFC DATA TRANSFER .....	155
10.1.4	AUTOMATIC PERFORMANCE ANALYSIS FOR ABAP FUNCTION MODULES OR CLASSES.....	156
10.2	CONCRETE MEASURES WITH SAP ECTR FOR SAP S/4HANA .....	156
10.2.1	LOGGING ON TO SAP IN THE BACKGROUND.....	156
10.2.2	PREFERENCE VARIABLE FOR TRANSACTIONS IN THE FOREGROUND .....	156
10.2.3	REPLACING FUNCTION MODULES .....	156
10.2.4	USING THE URL CHECKOUT .....	156
10.2.5	USING THE JCO REPOSITORY .....	157
10.2.6	FOLDER OPERATIONS.....	157
10.2.7	LIMITING THE AUTOMATIC DETAIL READING FOR WORK ITEMS IN THE SAP OFFICE INBOX .....	157
10.2.8	PREVENTING A MEMORY OVERFLOW WHEN CHECKING IN SEVERAL DOCUMENTS .....	157
10.2.9	TESTING THE HTTP CONNECTION.....	157
10.2.10	TESTING THE FTP CONNECTION.....	158
10.2.11	TESTING THE CONTENT SERVER PERFORMANCE.....	158
10.2.12	SAP CONTENT SERVER CACHE STATISTICS.....	158
10.3	IMPROVING THE DATABASE ACCESSES.....	158
10.3.1	TABLE DMS_PHRE_CD1.....	159
10.3.2	TABLE DMS_LODESC_CD1 .....	159
10.3.3	REGULAR UPDATES OF THE OPTIMIZER STATISTICS.....	159
10.3.4	UPDATING THE CLASS STATISTICS .....	159
10.3.5	MAX DB .....	159
10.4	DELETING BOMS WITH TRANSFORMATION MATRICES .....	160
10.5	LONG WAIT TIME WHEN A DOCUMENT BOM CANNOT BE WRITTEN .....	160
10.6	PLM OBSERVER SLEEP MODE.....	161
<b>11</b>	<b>ERROR ANALYSIS .....</b>	<b>162</b>
11.1	PROBLEM ANALYSIS .....	162
11.2	INFORMATION ON THE ERROR.....	162
11.3	ADDITIONAL HELP WITH TROUBLESHOOTING.....	162
<b>12</b>	<b>APPENDIX .....</b>	<b>164</b>
12.1	ADDITIONAL INFORMATION.....	164

# 1 INTRODUCTION



This document does not replace the manual for daily operation. SAP recommends generating this document for your own production operation.

## 1.1 About this Document

Your ability to implement and execute SAP applications around the clock while retaining high performance is essential to the success of your company.

This document contains important information regarding the smooth operation of your SAP applications. It describes the tasks to be performed and the tools to be used as well as the corresponding documentation, for example the Master Guide and the SAP application help.

### ***Target Groups***

- Technical consultants
- System administrators
- Solution consultants
- Persons responsible for business processes
- Support specialists

## 1.2 Global Definitions

### ***SAP application***

SAP software solution for a certain business area, for example ERP, CRM, PLM, SRM and SCM.

### ***Business scenario***

From a micro-economic perspective, a business scenario is a process flow that consists of different logically related processes. A business scenario typically involves several departments of a company and various business partners. From a technical perspective, a business scenario requires at least one SAP application (for example SAP ERP or SAP SCM) for each process flow and external system. A business scenario can be implemented independently of other business scenarios.

### ***Component***

A component is the smallest, separately producible, shippable, installable and maintainable technical software unit.

## 1.3 Important SAP Notes



Inform yourself regularly about updates to the Operations Guide.

## Important SAP Note

SAP Note	Title
810391	Update capability of the KPro DMS
942227	Update capability of the KPRO-DMS
942228	Modified report DMS_KPRO_CHECK1 for KPRO update
1302899	DIR has files with several active content versions
2049249	Release strategy for ABAP add-ons ECTR
2050165	Release strategy for ABAP add-on ECTRWUI

2 TECHNICAL SYSTEM LANDSCAPE

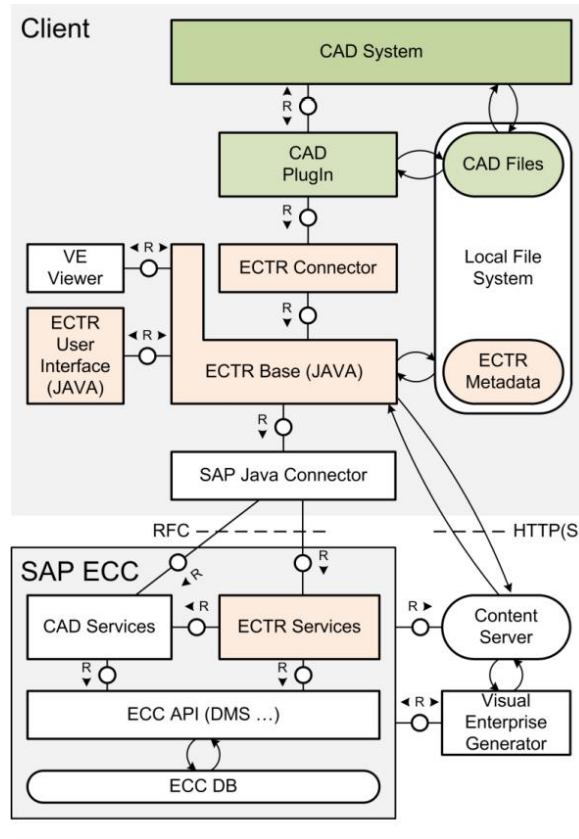


Figure 1: High Level Architecture

SAP ECTR for SAP S/4HANA includes the fields highlighted in orange. SAP ECTR for SAP S/4HANA connects the CAD plug-in with the SAP JavaConnector.

2.1 Documentation

The following table lists all documents that provide additional information regarding the technical system landscape.

Topic	Document	Quick link in SAP Service Marketplace (service.sap.com)
Application and industry-specific components, for example SAP Financial and SAP Retail	Master Guide	<a href="http://service.sap.com/instguides">http://service.sap.com/instguides</a>
Technology components, for example SAP web application server	Master Guide	<a href="http://service.sap.com/instguides">http://service.sap.com/instguides</a>
Technical configuration	Master Guide	<a href="http://service.sap.com/instguides">http://service.sap.com/instguides</a>
Scalability	Master Guide	<a href="http://service.sap.com/instguides">http://service.sap.com/instguides</a>

Topic	Document	Quick link in SAP Service Marketplace (service.sap.com)
High availability	Master Guide	<a href="http://service.sap.com/instguides">http://service.sap.com/instguides</a>
Security	Security Map Application Security Guide	<a href="http://service.sap.com/securityguide">http://service.sap.com/securityguide</a> <a href="http://service.sap.com/instguides">http://service.sap.com/instguides</a>

### 3 FILE HANDLING

This chapter explains how the files of a document are handled within SAP ECTR for SAP S/4HANA. This section also covers the distinction between the original and additional original and explains the differences in the handling of these files.

The following questions are answered in more detail:

What is an original / additional original?

How is an original / additional original edited?

Where are the files stored locally?

#### 3.1 Handling of the Main Original in an Application

In SAP ECTR for SAP S/4HANA, the handling of the original is based on separate file management. This file management uses a local or shared data buffer (cache) to minimize the need to transfer data from the SAP data storage system several times. Each primary application can have additional files that are all saved in the SAP system.

This procedure is intended for managing additional files (assembly instructions, certificates etc.) that are directly related to the primary document. Additional files do not require a separate management data record and are stored in a directory that is saved and opened together with the primary file.

#### Structure of the cache

Path	Description
C:\Users\...\AppData\Roaming\DSCSAG\filespace	Basic directory
<SAP_SYSTEM_NAME>_	System-/client-/user-specific directory
<SAP_CLIENT>_<SAP_USERNAME>\cache\	Cache
<HASHCODE>\	Hash code directory (is calculated from the document number)
<DOKNR>\	Document number directory
<DOKAR>_<DOKTL>_<DOCVR>\	Document type, document part, document version directory
<SAVECOUNTER>\	Save-counter
<DOKNR><DOKAR><DOKTL>_<DOCVR>.	Master file
<EXTENSION>	Enhancement of the master file

#### Additional files in the additional directory

```
C:\...\filespace\  
<SAP_SYSTEM_NAME>_<SAP_CLIENT>_<SAP_USERNAME>\cache\  

```

```
<HASHCODE>\<DOKNR>\<DOKAR>_<DOKTL>_<DOCVR>\
<SAVECOUNTER>\4711UGM000_00\
```

### Example

```
C:\Users\sk\AppData\Roaming\DSCSAG\filesystem\
I11_801_USER\cache\
120\4711\UGM_000_00\
12\4711ugm000_00\.di
12\4711ugm000_00.prt
```

## Structure of the session directory

Path	Description
C:\Users\sk\AppData\Roaming\DSCSAG\filesystem\	Basic directory
<SAP_SYSTEM_NAME>_	System/client /user-specific directory
<SAP_CLIENT>_<SAP_USERNAME>\sess\	Session
<SAP_USERNAME>\	SAP user
<DOKNR><DOKAR><DOKTL>_<DOCVR>.	Master file
<EXTENSION>	Enhancement of the master file

## Additional files in the session directory

```
C:\...\DSCSAG\filesystem\
<SAP_SYSTEM_NAME>_<SAP_CLIENT>_<SAP_USERNAME>\sess\
<SAP_USERNAME>\<DOKNR><DOKAR><DOKTL>_<DOCVR>\
```

### Example

```
C:\Users\sk\AppData\Roaming\DSCSAG\filesystem\
I11_801_USER\sess\
user\4711ugm000_00.prt
user\4711ugm000_00\.di
```

### 3.1.1 Validity check

If you need to determine the validity of locally buffered contents (files), perform a validity check. The validity check takes place in two stages:

1. The system checks whether the current save-counter of the document in SAP corresponds to that in the .di file in the session directory. If the save-counters are identical, the working directory reflects the current SAP status. If the save-counters differ, the files in the session directory are obsolete and must be updated from the cache directory.
2. The system checks whether the cache contains the current SAP status. To make the check efficient, the cache has a special structure: The path to the files contains the save-counter. Hence, the validity check is performed using a simple existence check.

If the following path exists, the current SAP status is in the cache and can be made available in the session directory.

c:\..\filesystem\I11\_801\_USER\cache\UNTER>

If the specified path does not exist, the cache is first updated from SAP. Following that, the working directory is updated from the cache.

If neither the working directory nor the cache are up-to-date, the data must be transferred from SAP.

### 3.1.2 Master Documents

For all master documents that are managed by SAP ECTR for SAP S/4HANA that take part in central (cached) file handling, the current status check is determined to use a save-counter. The save-counter is incremented every time the master original is saved in the BAdI. This requires the relevant original to be identified in ABAP. To do this, load the definitions of the master originals derived from the DTypes into the SAP preference table. If you save the identified files, the save-counter is incremented and stored in the RES5 field. At the same time, the LastModifiedDate field is written in CDESK\_DRAW.

Exception:

If no master original is defined for the DType, both the save-counter and the LastModifiedDate are incremented for each file operation. This procedure makes it possible to also save transactions in which external application viewables are available but no file handling is found.

### 3.1.3 Content Versions in the SAP System and Troubleshooting

In SAP Customizing, you can set it up so that a content version (not to be confused with the document version) is created for each check-in. Here, the latest version of the content is set to active and loaded when the document is opened again. In case of unwanted changes, you can restore an older content version without having to access the last document version.

However, in exceptional cases, communication problems during check-in can cause two content versions to be marked as active. In that case, opening the document is not possible due to the unclear assignment.

You can use the report (SE38) /DSCSAG/DMS\_KPRO\_DOUBLE\_REPAIR to resolve these assignment problems so that the respective incorrect content version no longer has "active" as its status.

In the "Program" field, enter report name /DSCSAG/DMS\_KPRO\_DOUBLE\_REPAIR. Choose [Execute](#) to start the repair transaction.

## 3.2 Working in Individual Files

### 3.2.1 Additional Files

Additional files are managed separately in the file system. They are opened for every access from the SAP system, i.e. they are not buffered. That is why no current status check is necessary for additional files. If an additional file is open for processing, its original is on the hard drive.

In the context of independent editing of additional files, you need further information regarding the status of a document. That is why information is included when the document details are read.

The required information regarding the files stored on the document is determined using a BAdI. In the SAP system, the information is stored as redundant flags and then returned with the document details. The required prerequisites for that are available in the [SAP Preference Upload](#).

The following fields are available:

Abbreviation	Description and values
MOE	No master original exists (1/0).
MOC	The master original is open for editing (1/0)

Abbreviation	Description and values
AOE	An additional original exists (1/0)
AOC	An additional original is open for editing (1/0)
2OE	A preview exists (1/0)
3OE	A 3D viewable exists (1/0)

### 3.2.2 Viewables

A caching mechanism has also been implemented for viewables. Different users can access the drive area.

For each document a (uniquely named) directory is created in which the respective viewables are stored. Different directories are used for 2D and 3D viewables. A simple current status check mechanism has also been implemented here. To avoid opening a file every time this information is needed, a `.ts` file is given a time stamp.

#### Previews

Previews are stored individually in SAP and in certain workstation applications. The corresponding preference variables are:

# Activate the display of previews (thumbnails) in SAP ECTR for SAP S/4HANA

```
plm.preview.enabled = true
```

# Definition of the workstation application and file extension that are used  
# to display 2D previews. This could be  
# a semicolon-separated list.

```
plm.preview.WsAppl = EPG; EPJ; JPG; GIF
plm.preview.extension = gif; jpg; jpeg; gif
```

The validity of previews is determined using the LMD (RES5) date stored for changes to the master original. The previews are changed and saved together with the master original.

If that is not the case (for example if previews are generated in the conversion server), you have to change the date stored in the RES5 field separately.

#### 3D Viewables

On the one hand, 3D viewables are becoming ever more important, on the other hand, they are significantly larger than thumbnails. The files are not handled together. 3D viewables therefore receive separate directories in the preview area, and the current status check takes place separately. This is necessary because it takes a long time to create viewables, which is done preferably by a conversion server.

When 3D viewables are saved in the SAP system, a BAAdI writes the change time (3DLMD) into the RES6 field. The time saved in 3DLMD is stamped to the `.ts-` file and used for the current status check.

To be able to identify the corresponding files, variable `PLM.3DWS` is used in SAP. This variable contains a semicolon-separated list of the WSAPPL with which the 3D viewables are saved.

### 3.3 Working with Multiple Session Directories

You can work with multiple session directories within an application. The user can change the active session directory.

Even if a version is checked out, different versions of a document can be edited. The version edited later and the version checked out earlier are checked out of different session directories. For more information on setting up and using multiple session directories, see SAP's Community WIKI:

<https://wiki.scn.sap.com/wiki/x/18loHQ>

### 3.4 Handling of Main Originals

The two preference variables force the display of all originals in the originals container:

```
plm.sap.original.show.all = true/false (Default)
```

```
plm.sap.original.show.all.{DType} = true/false (Default)
```

The DType settings are ignored. You can use `<ignore_files>` to filter originals.

The preference variable hides the original in the originals container:

```
plm.sap.original.exclude.master = true (Default)/false
```

If the preference variable is set to false, the DType can be used to specify which originals are displayed in the originals container and which are not.

The variable defines additional originals for a certain application type using the file extension:

```
plm.checkout.more.firstlevel.{application_type} = {file extension of additional original}
```

## 4 ATTRIBUTE TRANSFER BETWEEN THE SAP SYSTEM AND AN APPLICATION

A lot of information that is available about a certain document in SAP PLM can be transferred to an application. The attribute transfer is controlled using an XML definition. The attribute transfer is bidirectional. In this case, bidirectional means that multiple attributes from the CAD application can be stored in SAP PLM and vice versa.

### Note

Please do not use semicolons in the characteristic value of single-value characteristics. This character is reserved as a separator for multi-value characteristics. Otherwise, if you use a semicolon, errors may occur when transferring the characteristic values.

### Note

If the attribute `Longtext` is used in attribute mapping, the mapping for attribute `Description` is no longer taken into account. The attribute `Longtext` already contains the information from the attribute `Description`.

### Note

If the `attributes-from-sap.xml` or `attributes-to-sap.xml` files exist multiple times, for example in `%PLM_INSTDIR%\customize\config\` and `%PLM_INSTDIR%\addons\customize\config\`, then they must not contain duplicate sections because otherwise they will overwrite each other. For example, a default section may be defined only once.

### 4.1 Attribute Transfer from the SAP System to an Application

The properties of a document and of other dependent objects can be stored to the CAD part as a part attribute. Which attributes are transferred for a document depends on the DType and is specified in the DType definition. A dedicated attribute configuration section can be defined for each DType using the following preference variable: `<primary_application attribute_section="MASTERMODEL"...>`

In the attribute mapping file `%PLM_INSTDIR%\customize\config\attributes-from-sap.xml`, you can use the XML tag `<section name="..." ...>` to define different sections.

Which section is included is controlled using `<section include_section="..." ...>` (insert name of section).

### Note

If an error occurs when accessing the specified section, a corresponding warning is output in SAP ECTR for SAP S/4HANA, and the attribute transfer is canceled. No attributes are then available in the application. The attributes are stored as a part attribute and only updated with "Modifyevent". The CAD components identified in the SAP system as "inconsistent" (X in the DIRTY field of the document) are exceptions. All CAD components with attributes that do not match the SAP attributes are inconsistent. These include new documents created in SAP ECTR for SAP S/4HANA as well as all documents that were created in SAP ECTR for SAP S/4HANA using "Save As".

#### 4.1.1 Description of the Attribute Mapping File

When the XML file is processed, the code page of the local setting is used by default. The code page to be used can also be defined using the following preference variable:



```
plm.xml.metainf.encoding=<codepage>           for example: ISO-8859-2
```

To avoid problems due to syntax errors with umlauts or special characters, you can specify the code page directly in the XML file. Put the following line at the beginning of the XML file:


```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
```

Because the syntax of XML is known, only the tags are explained below.

## XML Tags for Control

XML Tag	Description
<mapping>	Defines the beginning of mapping. In accordance with XML syntax, the final XML instruction is then </mapping>.
<section>	<p>This tag defines the attribute section referenced by the DTypes. The mapping can contain various sections. The DTypes reference the sections. In the simplest cases, the sections: MASTERMODEL, dependent document and default exist. The default section is always processed. You can explicitly deactivate the section. The attributes of the default section exist in the document.</p> <p> <b>Example</b></p> <pre>&lt;section name="MASTERMODEL"       description="mapping for master model"&gt; &lt;/section&gt;</pre> <p><b>XML attributes</b></p> <ul style="list-style-type: none"> <li>• name="..." Name of the attribute section</li> <li>• description="..." Explanation of this attribute section</li> <li>• language="..." Language in which the attributes for the attribute transfer are to be synchronized. The attribute value is returned even if no name exists in the defined language.</li> <li>• include_section="..." (optional) Specifies the section in which the relevant attribute section is included.</li> </ul>
<APPL_ATTRIBUTE>	<p>The name of the attribute is specified here and whether the attribute is transferred to the application or is used only as a temporary auxiliary attribute. If you create an attribute comprising two or more different attributes, you require an auxiliary attribute. Auxiliary attributes have the XML attribute hidden=true.</p> <p><b>General</b></p> <pre>&lt;APPL_ATTRIBUTE name="%Name of the NX attribute%" //required                 prefix=%prefix for NX attributes%" //optional                 hidden="true/false" //optional                 expression="true/false"&gt; //optional &lt;/APPL_ATTRIBUTE&gt;</pre> <p> <b>Example</b></p> <pre>&lt;APPL_ATTRIBUTE name="SAP_CREATED_BY"       hidden="false"&gt;   &lt;STATUSLOG field="user" status="AA"             direction="forward"             stop_at="ZG;AC" /&gt; &lt;/APPL_ATTRIBUTE&gt;</pre>

XML Tag	Description
	<pre>&lt;APPL_ATTRIBUTE prefix="PLM_CLS_"   hidden="false"   expression="true"&gt;   &lt;DIR_CLS_GRP group="GEOM"/&gt; &lt;/APPL_ATTRIBUTE&gt;</pre>
	<p><b>i</b> Note</p> <p>Expressions are supported only by certain CAD applications, such as NX.</p>
	<p>XML attributes</p> <ul style="list-style-type: none"> <li>• name="..." Name of the attribute. Normally, these attribute names are case insensitive. They are converted into capital letters internally. If the application requires the attribute names to be case sensitive, you can set this using the following preference variables in the <code>default.txt</code> file:</li> <li>• <code>plm.attributes.useCaseSensitiveNames = true</code> When using case sensitive attribute names, the spelling of the attribute names in the attribute definition file and in the application must be identical. The attribute transfer can then be performed successfully.</li> <li>• hidden="..." true or false. If the value is true, the attribute is an auxiliary attribute. Auxiliary attributes are not transferred to the application. Auxiliary attributes provide data for other attributes that are transferred to the application.</li> <li>• expression="..." true or false. If the value is true, then the attribute can be used in Expression in the application.</li> <li>• transfer_empty_value="..." true (standard) or false. If the value is false, the system checks whether the value of the attribute is empty or null. If the value is empty or null, the attribute is treated as if hidden were set and the attribute is not transferred to the application.</li> </ul>
	<p><b>i</b> Note</p> <p>Expressions are supported only by certain CAD applications, such as NX.</p>

 **Example**


```
<mapping>
  <section name="default" description="Standard attribute mapping" language="EN">
    <APPL_ATTRIBUTE name="SAP_DESCRIPTION" hidden="false">
      <DIR field="Description"/>
    </APPL_ATTRIBUTE>
    <APPL_ATTRIBUTE name="SAP_APPLICATION" hidden="false">
      <DIR field="Userdefined4"/>
    </APPL_ATTRIBUTE>
    <APPL_ATTRIBUTE name="SAP_LASTSAVED" hidden="false">
      <DIR field="CHECKIN_DATE"/>
    </APPL_ATTRIBUTE>
    <APPL_ATTRIBUTE name="SAP_DOCID" hidden="false">
      <CUTLEADING character="0">
```

```

        <DIR field="Documentnumber"/>
    </CUTLEADING>
</APPL_ATTRIBUTE>
</section>
<section name="STDMOD"
        description="Additional attribute mapping for NX models">
    <APPL_ATTRIBUTE name="SAP_MATERIAL_NO" hidden="false">
        <MA field="MATERIAL"/>
    </APPL_ATTRIBUTE>
    <APPL_ATTRIBUTE name="SAP_MATERIAL_DESC" hidden="false">
        <MA field="MATL_DESC"/>
    </APPL_ATTRIBUTE>
    <APPL_ATTRIBUTE name="SAP_MATERIAL" hidden="false">
        <MA field="BASIC_MATL_NEW"/>
    </APPL_ATTRIBUTE>
    <APPL_ATTRIBUTE name="SAP_MATERIAL_WEIGHT" hidden="false">
        <MA field="NET_WEIGHT"/>
    </APPL_ATTRIBUTE>
</section>
<section name="TDOCS"
        description="Standard attribute mapping for text documents"/>
</mapping>

```

## XML Tags for Status Log Data

XML Tag	Description
<STATUSLOG>	<p>This tag defines that the attribute is an SAP attribute from the status log of the current document.</p> <hr/> <p> <b>Example</b></p> <pre>&lt;STATUSLOG field="date"             status="NP"             direction="backward"             stop_at="IA;AC"/&gt;</pre> <p>In this example, the system checks for the last status "NP" (default check). If status "IA" or "AC" is found, the search terminates.</p> <hr/> <p><b>XML attributes</b></p> <ul style="list-style-type: none"> <li>• <b>field="..."</b> Specifies what information is returned from the status log. Possible values are: user: <b>User</b> date and time: <b>Date</b> reason: <b>Reason</b></li> <li>• <b>status="..."</b> Specifies the status from which you want to have the values.</li> <li>• <b>direction="..."</b> Specifies in which direction the status log is searched. If you want to have the first status, choose <i>forward</i>, otherwise choose <i>backward</i>.</li> <li>• <b>stop_at="..."</b> Specifies at which status the processing of the status log terminates because subsequent statuses are obsolete. These are, for example, all</li> </ul>

XML Tag	Description
	<p>statuses that come after status "IA" (in process) when the status log is read backwards.</p> <p>If <code>stop_at</code> is not specified, the search is terminated at the first status that matches <code>status</code>.</p> <p>If <code>stop_at</code> is set to blank (""), no termination condition is set and the status log is searched completely in the specified direction. The last status that matches <code>status</code> is used.</p>
<DIR_FV>	<p>Here you can use the attributes from &lt;STATUSLOG&gt;; different possible values apply for attribute <code>field</code>. In this way you can also use XML tags &lt;DIR_FV&gt;, &lt;DIR_MM&gt; and &lt;DIR_MM_FV&gt;.</p> <p>XML attribute</p> <ul style="list-style-type: none"> <li><code>field="..."</code> Specifies what information is returned from the status log. Possible values are:  <code>statuslog-user: User name</code>  <code>statuslog-date: Date</code>  <code>statuslog-time: Time</code>  <code>statuslog-reason: Log text</code></li> </ul>

### XML Tags for Document Data

XML Tag	Description
<DIR>	<p>The document data record in XML attribute <code>field</code> is read.</p> <p>General</p> <pre>&lt;APPL_ATTRIBUTE name="%Attribute name in NX-Partfile%"   hidden="false"&gt;   &lt;DIR field="%Field name of document in SAP%" /&gt; &lt;/APPL_ATTRIBUTE&gt;</pre>
<DIR_FV>	The attributes of the current document are determined. The first version is determined and the values are output. The data of the first (or also the 0 version) is also printed in the drawing headers of the newer versions.
<DIR_MM>	The attributes of the current document are determined. The master model is imported and output. For a drawing, the document number of the master model can also be transferred.
<DIR_MM_FV>	From the current document, the attributes are determined, the first version of the master model is determined and its values are output.


The following fields of the document can be addressed in the XML tags for the document data via XML attribute `field="..."`:

Field	Description
<code>_FILE_EXTENSION</code>	File name extension of the source document
<code>_FILENAME</code>	Document name of the source document
<code>_FILENAME_WITHOUT_EXTENSION</code>	Document name of the source document without file name extension

Field	Description
AUTHORITYGROUP	Authorization group
CADINDICATOR	CAD indicator
CMFIXED	Document fixed
CMRELEVANCE	Relevant for configuration management
CREATEDATE	Creation date of the document
DELETEINDICATOR	Deletion flag
DESCRIPTION	Document description
DOCUMENTNUMBER	Document number
DOCUMENTPART	Document part
DOCUMENTTYPE	Document type
DOCUMENTVERSION	Document version
ECNUMBER	Change number
LABORATORY	Laboratory/design office
LONGTEXT	<p>Long text of the document. In addition, the row and language must be specified.</p> <p>Additional XML attributes</p> <ul style="list-style-type: none"> <li>row="..." Row of the long text that is transferred. Row="1" corresponds to row 2 of the long text. If row 2 of the long text does not contain a value, an error message appears. Set the value to "0".</li> <li>language="..." Identifies the language in which the long text is stored in SAP. EN=English, DE=German.</li> </ul>
PREDOCUMENTNUMBER	Document number of the higher-level document
PREDOCUMENTPART	Document part of the higher-level document
PREDOCUMENTTYPE	Document type of the higher-level document
PREDOCUMENTVERSION	Document version of the higher-level document
REFDOCUMENTNUMBER	Document number of the source document
REFDOCUMENTPART	Document part of the source document
REFDOCUMENTVERSION	Document version of the source document
REVLEVEL	<p>The revision level of a document is not recorded in SAP table DRAW, but has to be determined from the change number. This is time-consuming and affects performance. That is why the determination of the revision level from a document in SAP ECTR for SAP S/4HANA can be activated and deactivated using the following preference variable:</p> <pre>plm.document.readRevisionLevel = false(default)/true</pre>

Field	Description
	The revision level is only determined if the preference variable has the value <code>true</code> ; otherwise the attribute stays blank.
STATUSEXTERN	Status of the document (language-specific)
STATUSINTERN	Document status
STATUSLOG	Document management log field
STRUCTUREINDICATOR	Document management indicator
USERDEFINED1	Res field DRAW
USERDEFINED2	Res field DRAW
USERDEFINED3	Res field DRAW
USERDEFINED4	Res field DRAW
USERNAME	Name of the responsible person
VALIDFROMDATE	"Valid from" field
VRLDAT	Due date

### XML Tags for the Material Master


XML Tag	Description
<MA>	This tag outputs the data of the assigned material.   <b>Example</b> <pre>&lt;APPL_ATTRIBUTE name="SAP_MATERIAL_NO"   hidden="false"&gt;   &lt;MA field="MATERIAL"/&gt; &lt;/APPL_ATTRIBUTE&gt;</pre>
<MA_FV>	This tag outputs the data for the first version of the assigned material.
<MA_MM>	This tag outputs the data for the material of the corresponding master model.
<MA_MM_FV>	This tag outputs the data for the material of the first version of the corresponding master model.


The following fields of the material can be addressed in the XML tag for the material data via XML attribute `field="..."`:

XML Tag	Description
BASIC_MATL_NEW	Basic material

XML Tag	Description
GROSS_WT	Gross weight
HEIGHT	Height
LENGTH	Length
MATL_CAT	Material category
MATL_DESC	Material description
NET_WEIGHT	Net weight
PAGEFORMAT	Page format for the production note
UNIT_DIM	Unit of measure for length/width/height
UNIT_OF_WT	Unit of weight
VOLUME	Volume
VOLUMEUNIT	Unit of volume
WIDTH	Width

### XML Tags for the Classification




XML Tag	Description				
<DIR_CLS>	<p>Transfers the characteristic values for the classification of the document. Classification characteristics with different evaluations are written into the attribute as a list of characteristic values. By default, the separator between the list elements is a semicolon. You can use XML attribute separator="..." to redefine the separator.</p>				
	<p> <b>Example</b></p> <pre>&lt;APPL_ATTRIBUTE name="SAP_CLS" hidden="false"&gt;   &lt;DIR_CLS class="CLASS"            classtype="017"            characteristic="CHAR"            neutral="true"/&gt; &lt;/APPL_ATTRIBUTE&gt;</pre> <p>The SAP system can only read one class type per attribute section.</p>				
	<p>XML attributes</p> <table border="1"> <thead> <tr> <th>Field</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>class="..."</td> <td>Name of the class in which the characteristic occurs. If the characteristic can occur in any class, you can also enter a "*" for the class name</td> </tr> </tbody> </table>	Field	Description	class="..."	Name of the class in which the characteristic occurs. If the characteristic can occur in any class, you can also enter a "*" for the class name
Field	Description				
class="..."	Name of the class in which the characteristic occurs. If the characteristic can occur in any class, you can also enter a "*" for the class name				


XML Tag	Description								
	<table border="1"> <tr> <td data-bbox="644 253 943 342">classtype="..."</td> <td data-bbox="944 253 1390 342">SAP class type, for example 017 for document class</td> </tr> <tr> <td data-bbox="644 344 943 405">characteristic="..."</td> <td data-bbox="944 344 1390 405">Name of the characteristic</td> </tr> <tr> <td data-bbox="644 407 943 497">neutral="..."</td> <td data-bbox="944 407 1390 497">true or false, if true, the neutral characteristic value is used</td> </tr> <tr> <td data-bbox="644 499 943 613">separator="..."</td> <td data-bbox="944 499 1390 613">Change of the separator between the list elements for characteristics with multiple values (default = " ; ").</td> </tr> </table>	classtype="..."	SAP class type, for example 017 for document class	characteristic="..."	Name of the characteristic	neutral="..."	true or false, if true, the neutral characteristic value is used	separator="..."	Change of the separator between the list elements for characteristics with multiple values (default = " ; ").
classtype="..."	SAP class type, for example 017 for document class								
characteristic="..."	Name of the characteristic								
neutral="..."	true or false, if true, the neutral characteristic value is used								
separator="..."	Change of the separator between the list elements for characteristics with multiple values (default = " ; ").								
<DIR_FV_CLS>	Transfers the characteristic values for the classification of the first version of the document. The XML attribute description of <DIR_CLS> applies.								
<DIR_MM_CLS>	Transfers the characteristic values for the classification of the master model of the document. The XML attribute description of <DIR_CLS> applies.								
<DIR_MM_FV_CLS>	Transfers the characteristic values for the classification of the material of the first version of the master model of the document. The XML attribute description of <DIR_CLS> applies.								
<MA_CLS>	Transfers the characteristic values for the classification of the material of the document. The XML attribute description of <DIR_CLS> applies.								
<MA_FV_CLS>	Transfers the characteristic values for the classification of the material of the first version of the document. The XML attribute description of <DIR_CLS> applies.								
<MA_MM_CLS>	Transfers the characteristic values for the classification of the material of the master model of the document. The XML attribute description of <DIR_CLS> applies.								
<MA_MM_FV_CLS>	Transfers the characteristic values for the classification of the material of the first version of the material of the document. The XML attribute description of <DIR_CLS> applies.								
<DIR_CLS_GRP>	<p>Specifies a characteristic group of the classification of the document in the SAP system. The characteristic group can be assigned to every characteristic in the characteristic definition in the SAP system via the <a href="#">Characteristic Group</a> input field. All characteristics of the document that belong to the specified characteristic group are transferred to the application as attributes. The names of the transferred attributes then receive the prefix defined using <code>prefix="..."</code>. Instead of <code>group="GEOM"</code> you can also have <code>group="*"</code>; in that case all classification data is copied. Regular expressions or constructs such as <code>group="GEO*"</code> are not supported, <code>"*"</code> is the only supported form.</p>								
	<p> <b>Example</b></p> <pre>&lt;APPL_ATTRIBUTES prefix="PLM_CLS_"   hidden="false"   expressions="true"&gt;   &lt;DIR_CLS_GRP group="GEOM"/&gt; &lt;/APPL_ATTRIBUTES&gt;</pre>								
	XML attributes								

XML Tag	Description								
	<table border="1"> <thead> <tr> <th data-bbox="646 255 927 311">Field</th> <th data-bbox="928 255 1385 311">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="646 313 927 488">prefix="..."</td> <td data-bbox="928 313 1385 488">If all characteristics of a classification are transferred. A prefix is defined, which is placed in front of every characteristic name for example PLM_CLS_&lt;characteristic_name&gt;.</td> </tr> <tr> <td data-bbox="646 490 927 573">hidden="..."</td> <td data-bbox="928 490 1385 573">true or false. If the value is true, the attribute is an auxiliary attribute</td> </tr> <tr> <td data-bbox="646 575 927 658">expressions="..."</td> <td data-bbox="928 575 1385 658">true or false. If the value is true, the attributes can be used in Expressions</td> </tr> </tbody> </table>	Field	Description	prefix="..."	If all characteristics of a classification are transferred. A prefix is defined, which is placed in front of every characteristic name for example PLM_CLS_<characteristic_name>.	hidden="..."	true or false. If the value is true, the attribute is an auxiliary attribute	expressions="..."	true or false. If the value is true, the attributes can be used in Expressions
Field	Description								
prefix="..."	If all characteristics of a classification are transferred. A prefix is defined, which is placed in front of every characteristic name for example PLM_CLS_<characteristic_name>.								
hidden="..."	true or false. If the value is true, the attribute is an auxiliary attribute								
expressions="..."	true or false. If the value is true, the attributes can be used in Expressions								
<DIR_MM_CLS_GRP>	Specifies a characteristic group for the classification of the master model of the document in the SAP system. The characteristic group can be assigned to every characteristic in the characteristic definition in the SAP system via the <a href="#">Characteristic Group</a> input field. All characteristics of the document that belong to the specified characteristic group are transferred to the application as attributes. The XML attribute description of <DIR_CLS_GRP> is valid.								
<DIR_FV_CLS_GRP>	Specifies a characteristic group for the classification of the first version of the document in the SAP system. The characteristic group can be assigned to every characteristic in the characteristic definition in the SAP system via the "Characteristic Group" input field. All characteristics of the document that belong to the specified characteristic group are transferred to the application as attributes. The XML attribute description of <DIR_CLS_GRP> is valid.								
<DIR_MM_FV_CLS_GRP>	Specifies a characteristic group for the classification of the first version of the master model document in the SAP system. The characteristic group can be assigned to every characteristic in the characteristic definition in the SAP system via the "Characteristic Group" input field. All characteristics of the document that belong to the specified characteristic group are transferred to the application as attributes. The XML attribute description of <DIR_CLS_GRP> is valid.								
<MA_CLS_GRP>	Specifies a characteristic group for the classification of the material of the document in the SAP system. The characteristic group can be assigned to every characteristic in the characteristic definition in the SAP system via the "Characteristic Group" input field. All characteristics of the material that belong to the specified characteristic group are transferred to the application as attributes. The XML attribute description of <DIR_CLS_GRP> is valid.								
<MA_MM_CLS_GRP>	Specifies a characteristic group for the classification of the material of the master model of the document in the SAP system. The characteristic group can be assigned to every characteristic in the characteristic definition in the SAP system via the "Characteristic Group" input field. All characteristics of the material that belong to the specified characteristic group are transferred to the application as attributes. The XML attribute description of <DIR_CLS_GRP> is valid.								
<MA_FV_CLS_GRP>	Specifies a characteristic group for the classification of the material of the first version of the document in the SAP system. The characteristic group can be assigned to every characteristic in the characteristic definition in the SAP system via the "Characteristic Group" input field. All characteristics of the material that belong to the specified characteristic								

XML Tag	Description
	group are transferred to the application as attributes. The XML attribute description of <DIR_CLS_GRP> is valid.
<MA_MM_FV_CLS_GRP>	Specifies a characteristic group for the classification of the material of the first version of the master model of the document in the SAP system. The characteristic group can be assigned to every characteristic in the characteristic definition in the SAP system via the "Characteristic Group" input field. All characteristics of the material that belong to the specified characteristic group are transferred to the application as attributes. The XML attribute description of <DIR_CLS_GRP> is valid.

## XML Tags for Editing and Formatting





XML Tag	Description
<MAP>	<p>You can use this tag to replace values from the SAP system with values in a file. The files for the value mapping must be located in the <code>customize/config</code> directory. The attribute mapping file has the following syntax: old value 1 = new value 1</p> <p> <b>Example</b> The value that SAP determines for the "user" is once again exchanged via the <code>user.txt</code> mapping file.</p> <pre>&lt;MAP file="user.txt"&gt;   &lt;STATUSLOG field="user" status="AA"     direction="forward" /&gt; &lt;/MAP&gt;</pre> <p> <b>Example</b> //Indicates what the mapping file is called: &lt;MAP file="document.txt"&gt; //Refers to a characteristic of a document, here //ExampleCharacteristic: &lt;DIR_CLS class="class1"   classtype="017"   characteristic="ExampleCharacteristic"   transfer_empty_value="true"/&gt; &lt;/DIR_CLS&gt; &lt;/MAP&gt; <p>The value mapping in <code>document.txt</code> could be as follows: VALUE1 = One VALUE2 = Two</p> <p>If the characteristic <code>ExampleCharacteristic</code> is inserted into a document with the value <code>VALUE1</code> from the SAP system, then the mapping from <code>document.txt</code> comes into play. This value is displayed as <code>One</code> in the document.</p> </p>
<FIRSTNONEMPTY>	<p>The first set value is taken from a list.</p> <p> <b>Note</b></p> <p>When <code>transfer_empty_value = true</code> is set in the parent node, you can define your default value</p>

XML Tag	Description																
	<p>(defaultvalue) here, which is used when all values in a list are empty.</p> <p> <b>Example</b></p> <pre>&lt;FIRSTNONEMPTY defaultvalue="xy"&gt;   &lt;DIR_CLS class="class1"     classtype="017"     characteristic="char4711"/&gt;   &lt;DIR_CLS class="class2"     classtype="017"     characteristic="char1147"/&gt; &lt;/FIRSTNONEMPTY&gt;</pre> <p>Result: The first available classification is copied.</p>																
<DATETIMEFORMAT>	This tag defines a date format. The date can be converted from the SAP system, into any date format.																
	<p>XML attributes</p> <table border="1" data-bbox="651 846 1394 1630"> <thead> <tr> <th data-bbox="655 853 970 898">Field</th> <th data-bbox="973 853 1390 898">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="655 902 970 981">outputFormat="..."</td> <td data-bbox="973 902 1390 981">Format string for the output format of the date (see symbol table).</td> </tr> <tr> <td data-bbox="655 985 970 1093">inputFormat="..."</td> <td data-bbox="973 985 1390 1093">Format string for the input format of the date. If the date is not available as an object.</td> </tr> <tr> <td data-bbox="655 1097 970 1294">outputTimezone="..."</td> <td data-bbox="973 1097 1390 1294">Specifies the timezone in which the output takes place. If no timezone is specified, the local timezone is used. You can use all valid IDs for timezones.</td> </tr> <tr> <td data-bbox="655 1299 970 1518">inputTimezone="..."</td> <td data-bbox="973 1299 1390 1518">Specifies the timezone in which the input is available. If the date is not an object. If the date is available as an object, its timezone is used and this information is ignored. If no timezone is specified, the local timezone is used.</td> </tr> <tr> <td data-bbox="655 1523 970 1630">outputLocale="..."</td> <td data-bbox="973 1523 1390 1630">Specifies the local language. This ensures that the date is displayed in the correct format automatically.</td> </tr> </tbody> </table>	Field	Description	outputFormat="..."	Format string for the output format of the date (see symbol table).	inputFormat="..."	Format string for the input format of the date. If the date is not available as an object.	outputTimezone="..."	Specifies the timezone in which the output takes place. If no timezone is specified, the local timezone is used. You can use all valid IDs for timezones.	inputTimezone="..."	Specifies the timezone in which the input is available. If the date is not an object. If the date is available as an object, its timezone is used and this information is ignored. If no timezone is specified, the local timezone is used.	outputLocale="..."	Specifies the local language. This ensures that the date is displayed in the correct format automatically.				
Field	Description																
outputFormat="..."	Format string for the output format of the date (see symbol table).																
inputFormat="..."	Format string for the input format of the date. If the date is not available as an object.																
outputTimezone="..."	Specifies the timezone in which the output takes place. If no timezone is specified, the local timezone is used. You can use all valid IDs for timezones.																
inputTimezone="..."	Specifies the timezone in which the input is available. If the date is not an object. If the date is available as an object, its timezone is used and this information is ignored. If no timezone is specified, the local timezone is used.																
outputLocale="..."	Specifies the local language. This ensures that the date is displayed in the correct format automatically.																
	<table border="1" data-bbox="651 1664 1401 1966"> <thead> <tr> <th data-bbox="655 1671 778 1715">Icon</th> <th data-bbox="782 1671 1070 1715">Meaning</th> <th data-bbox="1074 1671 1262 1715">Presentation</th> <th data-bbox="1265 1671 1396 1715">Example</th> </tr> </thead> <tbody> <tr> <td data-bbox="655 1720 778 1794">G</td> <td data-bbox="782 1720 1070 1794">era designator</td> <td data-bbox="1074 1720 1262 1794">(Text)</td> <td data-bbox="1265 1720 1396 1794">AD</td> </tr> <tr> <td data-bbox="655 1798 778 1872">y</td> <td data-bbox="782 1798 1070 1872">year</td> <td data-bbox="1074 1798 1262 1872">(Number)</td> <td data-bbox="1265 1798 1396 1872">1996</td> </tr> <tr> <td data-bbox="655 1877 778 1951">M</td> <td data-bbox="782 1877 1070 1951">month in year</td> <td data-bbox="1074 1877 1262 1951">(Text &amp; Number)</td> <td data-bbox="1265 1877 1396 1951">July &amp; 07</td> </tr> </tbody> </table>	Icon	Meaning	Presentation	Example	G	era designator	(Text)	AD	y	year	(Number)	1996	M	month in year	(Text & Number)	July & 07
Icon	Meaning	Presentation	Example														
G	era designator	(Text)	AD														
y	year	(Number)	1996														
M	month in year	(Text & Number)	July & 07														



XML Tag	Description		
---------	-------------	--	--



d	day in month	(Number)	10
h	hour in am/pm (1~12)	(Number)	12
H	hour in day (0~23)	(Number)	0
m	minute in hour	(Number)	30
s	second in minute	(Number)	55
S	millisecond	(Number)	978
E	day in week	(Text)	Tuesday
D	day in year	(Number)	189
F	day of week in month	(Number)	2
w	week in year	(Number)	27
W	week in month	(Number)	2
a	am/pm marker	(Text)	PM
k	hour in day (1~24)	(Number)	24
K	hour in am/pm (0~11)	(Number)	0
z	time zone	(Text)	PST
`	escape for text		
"	single quote	(Delimiter) (Literal)	'

Presentation	Description
(Text)	> 4 letters – complete form, < 4 abbreviated version, if available.
(Number)	Minimum number of digits. Shorter numbers are filled with zeros. Year is handled separately.
(Text & Number)	3 or more, use the text, if not, use a number.

XML Tag	Description
<p>&lt;SUBSTRING&gt;</p>	<p>With this tag you can use parts of a text. In the example at the bottom, the document number is read from the third to the seventh character and forwarded.</p> <p> <b>Example</b>  <pre>&lt;SUBSTRING begin="3" end="7"&gt;   &lt;DIR field="DOCUMENTNUMBER" /&gt; &lt;/SUBSTRING&gt;</pre></p>
<p>&lt;CUTLEADING&gt;</p>	<p>You can use this tag to remove characters at the start of a string. You can truncate the leading zeros from the document number. You can set truncation to stop when a certain length is reached. To do this, use parameter "length".</p> <p> <b>Example</b>  <pre>&lt;CUTLEADING character="0" length="10"&gt;   &lt;DIR field="DOCUMENTNUMBER" /&gt; &lt;/CUTLEADING&gt;</pre></p>
<p>&lt;CUTTRAILING&gt;</p>	<p>For &lt;CUTTRAILING&gt; the same applies as for &lt;CUTLEADING&gt;, the only difference is that truncation takes place from the back.</p>
<p>&lt;SETLEADING&gt;</p>	<p>If you want to add a certain string of characters to the front until the string reaches a certain length.</p> <p> <b>Example</b>  <pre>&lt;SETLEADING character="0" length="25"&gt;   &lt;FORMAT outputFormat="4711" /&gt; &lt;/SETLEADING&gt;</pre> <p>Result: Document number with leading zeros, for example 4711 → 00000000000004711</p> </p>
<p>&lt;SETTRAILING&gt;</p>	<p>For &lt;SETTRAILING&gt; the same applies as for &lt;SETLEADING&gt;, except the characters are added at the end.</p>
<p>&lt;XFORMAT&gt;</p>	<p>If you want to combine multiple attributes and/or static texts on an attribute, use &lt;XFORMAT&gt;. XFORMAT can also change how numbers are displayed.</p> <p> <b>Example</b>  <pre>&lt;XFORMAT outputFormat="{0}-{1}-{2}-{3}"&gt;   &lt;DIR field="documenttype"/&gt;   &lt;DIR field="documentnumber"/&gt;   &lt;DIR field="documentpart"/&gt;   &lt;DIR field="documentversion"/&gt; &lt;/XFORMAT&gt;</pre> <p>Result: UMK-4711-000-01</p> <p>If you want to convert an attribute to exponential notation, use the format string 0.#####E0.</p> <p>In XFORMAT outputFormat, the preference and environment variables can be referenced or values can simply be used that overwrite the original value. The \$(pref[&lt;abc.def.ghi&gt;]) placeholder is provided for the preference variables and the \$(env[&lt;abc.def.ghi&gt;]) placeholder is provided for the environment variables.</p> </p>

XML Tag	Description						
	<p>plm.om.example is a preference variable with the value "ABC" (plm.om.example = ABC).</p> <p><b>Before replacement:</b> &lt;XFORMAT outputFormat="my output format \$(pref[plm.om.example])"&gt;</p> <p><b>After replacement:</b> &lt;XFORMAT outputFormat="my output format ABC"&gt;</p> <p><b>Format string examples for the different formatting options:</b></p> <p><b>Integer:</b> &lt;XFORMAT outputFormat="{0,number,integer}"&gt;</p> <p><b>Floating-point number with two decimal places:</b> &lt;XFORMAT outputFormat="{0,number,0.##}"&gt;</p> <p><b>Exponential format:</b> &lt;XFORMAT outputFormat="{0,number,0.#####E0}"&gt;</p>						
<REPLACE>	<p>You can use this tag to replace characters in a string.</p> <p> <b>Example</b></p> <p>Replace all commas with periods.</p> <pre>&lt;REPLACE pattern="," replacement="."       global="false" &gt;   &lt;APPL_ATTRIBUTE name="SAP_MATERIAL_WEIGHT"/&gt; &lt;/REPLACE&gt;</pre> <p>Replace all umlauts with a different spelling.</p> <pre>&lt;REPLACE pattern="\u00E4" replacement="\\$a"       global="true"&gt;   &lt;REPLACE pattern="\u00FC" replacement="\\$u"       global="true"&gt;     &lt;REPLACE pattern="\u00F6" replacement="\\$o"       global="true"&gt;       &lt;REPLACE pattern="\u00C4" replacement="\\$A"       global="true"&gt;         &lt;REPLACE pattern="\u00DC" replacement="\\$U"       global="true"&gt;           &lt;REPLACE pattern="\u00D6" replacement="\\$O"       global="true"&gt;             &lt;REPLACE pattern="\u00DF"       replacement="\\$S"       global="true"&gt;               &lt;DIR field="LONGTEXT" row="0"       language="DE"/&gt;             &lt;/REPLACE&gt;           &lt;/REPLACE&gt;         &lt;/REPLACE&gt;       &lt;/REPLACE&gt;     &lt;/REPLACE&gt;   &lt;/REPLACE&gt; &lt;/REPLACE&gt;</pre> <p><b>XML attributes</b></p> <table border="1" data-bbox="651 1664 1393 1944"> <thead> <tr> <th data-bbox="655 1671 900 1715">Presentations</th> <th data-bbox="903 1671 1388 1715">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="655 1720 900 1865">pattern="..."</td> <td data-bbox="903 1720 1388 1865">Character sequence that is searched for in the string as a regular expression. Mark special characters with an introductory "\\".</td> </tr> <tr> <td data-bbox="655 1870 900 1944">replacement="..."</td> <td data-bbox="903 1870 1388 1944">Character sequence after which the found partial string is replaced.</td> </tr> </tbody> </table>	Presentations	Description	pattern="..."	Character sequence that is searched for in the string as a regular expression. Mark special characters with an introductory "\\".	replacement="..."	Character sequence after which the found partial string is replaced.
Presentations	Description						
pattern="..."	Character sequence that is searched for in the string as a regular expression. Mark special characters with an introductory "\\".						
replacement="..."	Character sequence after which the found partial string is replaced.						

XML Tag	Description										
	<table border="1" data-bbox="651 264 1393 412"> <tr> <td data-bbox="651 264 898 412">global="..."</td> <td data-bbox="901 264 1393 412">true or false; if true, then all found character sequences are replaced. If false, only the first found character sequence is replaced.</td> </tr> </table> <p data-bbox="651 450 1455 645">  <b>Example</b>  <code>&lt;REPLACE pattern="." replacement=","&gt;</code>  This would result in all characters being replaced by ",". Global is not specified here and therefore true. You can prevent this by marking pattern as a special character:  <code>&lt;REPLACE pattern="\." replacement=","&gt;</code> </p>	global="..."	true or false; if true, then all found character sequences are replaced. If false, only the first found character sequence is replaced.								
global="..."	true or false; if true, then all found character sequences are replaced. If false, only the first found character sequence is replaced.										
<CALC>	<p data-bbox="651 678 1455 887">  <b>Example</b>  <code>&lt;REPLACE pattern="." replacement=","&gt;</code>  All characters would be replaced with "," if this were set. Global is not specified here and therefore true. You can prevent this by marking pattern as a special character:  <code>&lt;REPLACE pattern="\." replacement=","&gt;</code> </p> <p data-bbox="651 909 1455 981">You can use this tag to perform calculations with attributes that can be interpreted as numeric values.</p> <p data-bbox="651 1003 1455 1122"> XML attributes:  operation="..."  Calculation operation that is executed. If a specified attribute cannot be interpreted as a numeric value, this attribute is ignored in the calculation. </p> <p data-bbox="651 1144 1455 1171">The following calculation operations have been defined:</p> <table border="1" data-bbox="651 1176 1393 2020"> <thead> <tr> <th data-bbox="651 1176 842 1227">Field</th> <th data-bbox="844 1176 1393 1227">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="651 1232 842 1339">SUM</td> <td data-bbox="844 1232 1393 1339">The specified attributes are added up and the total is transferred to the enclosing element. If no calculation is possible, the value 0 is returned.</td> </tr> <tr> <td data-bbox="651 1344 842 1541">SUB</td> <td data-bbox="844 1344 1393 1541">The first attribute is used as the basis and the other attributes are deducted from this basic value as subtrahends. The result is transferred to the enclosing element. If there is no value for an attribute, the value 0.0 is assumed. Hence, the subtrahends do not contribute to the result.</td> </tr> <tr> <td data-bbox="651 1545 842 1706">MUL</td> <td data-bbox="844 1545 1393 1706">The specified attributes are multiplied and the product is transferred to the enclosing element. If there is no value for an attribute, the value 1.0 is assumed. Hence, missing factors do not contribute to the result.</td> </tr> <tr> <td data-bbox="651 1711 842 2020">DIV</td> <td data-bbox="844 1711 1393 2020">The first attribute is the basis. The first attribute is divided by the second attribute. The result is divided by the next attribute and the final result is transferred to the enclosing element. If there is no value for the first attribute, the value 0.0 is assumed and the result is already determined: 0.0. For missing values from the second attribute onwards, the value 1.0 is assumed. Hence, missing divisors do not contribute to the result.</td> </tr> </tbody> </table>	Field	Description	SUM	The specified attributes are added up and the total is transferred to the enclosing element. If no calculation is possible, the value 0 is returned.	SUB	The first attribute is used as the basis and the other attributes are deducted from this basic value as subtrahends. The result is transferred to the enclosing element. If there is no value for an attribute, the value 0.0 is assumed. Hence, the subtrahends do not contribute to the result.	MUL	The specified attributes are multiplied and the product is transferred to the enclosing element. If there is no value for an attribute, the value 1.0 is assumed. Hence, missing factors do not contribute to the result.	DIV	The first attribute is the basis. The first attribute is divided by the second attribute. The result is divided by the next attribute and the final result is transferred to the enclosing element. If there is no value for the first attribute, the value 0.0 is assumed and the result is already determined: 0.0. For missing values from the second attribute onwards, the value 1.0 is assumed. Hence, missing divisors do not contribute to the result.
Field	Description										
SUM	The specified attributes are added up and the total is transferred to the enclosing element. If no calculation is possible, the value 0 is returned.										
SUB	The first attribute is used as the basis and the other attributes are deducted from this basic value as subtrahends. The result is transferred to the enclosing element. If there is no value for an attribute, the value 0.0 is assumed. Hence, the subtrahends do not contribute to the result.										
MUL	The specified attributes are multiplied and the product is transferred to the enclosing element. If there is no value for an attribute, the value 1.0 is assumed. Hence, missing factors do not contribute to the result.										
DIV	The first attribute is the basis. The first attribute is divided by the second attribute. The result is divided by the next attribute and the final result is transferred to the enclosing element. If there is no value for the first attribute, the value 0.0 is assumed and the result is already determined: 0.0. For missing values from the second attribute onwards, the value 1.0 is assumed. Hence, missing divisors do not contribute to the result.										

XML Tag	Description								
	<table border="1"> <tr> <td data-bbox="646 255 839 405"></td> <td data-bbox="842 255 1390 405">Response in case of a division by zero: Divisors with the value 0.0 are ignored during division and a corresponding entry is output in the SAP ECTR for SAP S/4HANA log file</td> </tr> <tr> <td data-bbox="646 409 839 546">MIN</td> <td data-bbox="842 409 1390 546">The attribute with the lowest numeric value is determined and transferred to the enclosing element. If no calculation is possible, the value 0 is returned.</td> </tr> <tr> <td data-bbox="646 551 839 687">MAX</td> <td data-bbox="842 551 1390 687">The attribute with the highest numeric value is determined and transferred to the enclosing element. If no calculation is possible, the value 0 is returned.</td> </tr> <tr> <td data-bbox="646 692 839 799">COUNT</td> <td data-bbox="842 692 1390 799">Non-empty attributes are counted. The number of non-empty attributes is transferred to the enclosing element.</td> </tr> </table>		Response in case of a division by zero: Divisors with the value 0.0 are ignored during division and a corresponding entry is output in the SAP ECTR for SAP S/4HANA log file	MIN	The attribute with the lowest numeric value is determined and transferred to the enclosing element. If no calculation is possible, the value 0 is returned.	MAX	The attribute with the highest numeric value is determined and transferred to the enclosing element. If no calculation is possible, the value 0 is returned.	COUNT	Non-empty attributes are counted. The number of non-empty attributes is transferred to the enclosing element.
	Response in case of a division by zero: Divisors with the value 0.0 are ignored during division and a corresponding entry is output in the SAP ECTR for SAP S/4HANA log file								
MIN	The attribute with the lowest numeric value is determined and transferred to the enclosing element. If no calculation is possible, the value 0 is returned.								
MAX	The attribute with the highest numeric value is determined and transferred to the enclosing element. If no calculation is possible, the value 0 is returned.								
COUNT	Non-empty attributes are counted. The number of non-empty attributes is transferred to the enclosing element.								
<SCRIPT>	<p>You can use this tag to execute external JavaScript. The script file is expected in the %PLM_INSTDIR%/customize/scripts/attributes/ directory. The resulting value is specified in the JavaScript.</p> <hr/> <p> <b>Example</b></p> <pre>&lt;APPL_ATTRIBUTE name="ENV_XYZ"   &lt;SCRIPT name="getenv.js"&gt; //the script name is specified here.   &lt;XFORMAT string="XYZ"/&gt; //All values that are to be transferred //must be listed within the SCRIPT tags.   &lt;/SCRIPT&gt; &lt;/APPL_ATTRIBUTE&gt;</pre> <hr/> <p>The corresponding JavaScript checkVersion.js:</p> <pre>value=""; //The result is returned with the variable value, //which is why it must always be set to "" at the start. if(argv.length&gt;0) //with argv, the first transferred value is addressed. //with argv2, argv3, and so on, the next transferred //values are addressed {   var x = Packages.java.lang.System.getenv(argv[0]);   if(x!=null)     value = new String(x); //value is returned here }</pre> <hr/> <p><b>Result:</b> If the transferred value is not "0", it is written to the ENV_XYZ attribute.</p> <hr/> <p> <b>Warning</b> If you want to use the &lt;SCRIPT&gt; tag, please contact the system administrator.</p>								

## 4.2 Attribute Transfer from an Application to the SAP System

The attribute transfer from an application to the SAP system is defined in the following attribute mapping file.

```
%PLM_INSTDIR%\customize\config\attributes-to-sap.xml
```

When the file is saved, all attributes are stored in an XML file in the additional directory.

```
%PLM_WORKDIR%\sess\


```

SAP ECTR for SAP S/4HANA reads the file according to the attribute mapping file and sets the configured values in the SAP system.

### 4.2.1 Supported XML Tags


The attribute mapping file is used to set the SAP data listed below via the application:

#### XML Tags for Control

XML Tag	Description
<mapping>	Defines the beginning of mapping. In accordance with XML syntax, the final XML instruction is then </mapping>.
<section>	<p>This tag defines the attribute section referenced by the DTypes. The mapping can contain various sections. The default section is always processed. You can explicitly deactivate the section. The attributes of the default section exist in the document.</p> <p> <b>Example</b></p> <pre>&lt;section name="STDMOD" include_section="STANDARD" description=" Model mapping CAD -&amp;gt; SAP "&gt; &lt;DIR field="Description"&gt;   &lt;APPL_ATTRIBUTE name="SAP_DESCRIPTION"/&gt; &lt;/DIR&gt; &lt;/section&gt;</pre> <p><b>XML attributes</b></p> <ul style="list-style-type: none"> <li>• name="..." Name of the attribute section</li> <li>• description="..." Explanation of this attribute section</li> <li>• include_section="..." (optional) Specifies the section in which the relevant attribute section is included.</li> </ul>


#### XML Tags for Documents

XML Tag	Description
<DIR>	<p>XML attribute field="..." can be used to address the fields of the document that can be changed in the SAP system. For long texts, there is a difference in the transfer of SAP data from the SAP system to the application. The following special rules apply: Attribute "row" is not taken into account. The number and order of definitions in file attributes-to-sap.xml determines the number and order of rows in the long text.</p>


XML Tag	Description
	<p>The first row of the long text overwrites the description of the document (caused by SAP). The first row of the long text contains the description of the document</p> <hr/> <p> <b>Example</b></p> <pre data-bbox="655 436 1249 1245"> &lt;DIR field="LONGTEXT" language="EN"&gt;   &lt;APPL_ATTRIBUTE name="DESCRIPTION_EN"/&gt; &lt;/DIR&gt; &lt;DIR field="LONGTEXT" language="EN"&gt;   &lt;APPL_ATTRIBUTE name="LTEXT_EN_0"/&gt; &lt;/DIR&gt; &lt;DIR field="LONGTEXT" language="EN"&gt;   &lt;APPL_ATTRIBUTE name="LTEXT_EN_1"/&gt; &lt;/DIR&gt; &lt;DIR field="LONGTEXT" language="EN"&gt;   &lt;APPL_ATTRIBUTE name="LTEXT_EN_2"/&gt; &lt;/DIR&gt; &lt;DIR field="LONGTEXT" language="EN"&gt;   &lt;APPL_ATTRIBUTE name="LTEXT_EN_3"/&gt; &lt;/DIR&gt; &lt;DIR field="LONGTEXT" language="DE"&gt;   &lt;APPL_ATTRIBUTE name="DESCRIPTION_DE"/&gt; &lt;/DIR&gt; &lt;DIR field="LONGTEXT" language="DE"&gt;   &lt;APPL_ATTRIBUTE name="LTEXT_DE_0"/&gt; &lt;/DIR&gt; &lt;DIR field="LONGTEXT" language="DE"&gt;   &lt;APPL_ATTRIBUTE name="LTEXT_DE_1"/&gt; &lt;/DIR&gt; &lt;DIR field="LONGTEXT" language="DE"&gt;   &lt;APPL_ATTRIBUTE name="LTEXT_DE_2"/&gt; &lt;/DIR&gt; &lt;DIR field="LONGTEXT" language="DE"&gt;   &lt;APPL_ATTRIBUTE name="LTEXT_DE_3"/&gt; &lt;/DIR&gt; </pre> <hr/> <p><b>XML attributes</b></p> <ul style="list-style-type: none"> <li>• <code>transfer_empty_value="..."</code> true or false (default)</li> </ul> <p>If true, blank characteristic values are also transferred. Make sure that the characteristic value in the SAP system is allowed to be blank. Note that this may not be blank for the LONGTEXT or DESCRIPTION fields, even if this variable is set.</p>

### XML Tag for Materials

XML Tag	Description						
<MA>	<p>For the transfer from the application to the SAP system, you can address the following fields of the material in the XML tags for material data via XML attribute <code>field="..."</code>:</p> <table border="1" data-bbox="655 1854 1393 2024"> <thead> <tr> <th>Field</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>MATL_DESC</td> <td>Material description</td> </tr> <tr> <td>PAGEFORMAT</td> <td>Page format of the production memo</td> </tr> </tbody> </table>	Field	Description	MATL_DESC	Material description	PAGEFORMAT	Page format of the production memo
Field	Description						
MATL_DESC	Material description						
PAGEFORMAT	Page format of the production memo						

XML Tag	Description														
	<table border="1"> <tr> <td>DOC_FORMAT</td> <td>Document format (free text)</td> </tr> <tr> <td>GROSS_WT</td> <td>Gross weight</td> </tr> <tr> <td>NET_WEIGHT</td> <td>Net weight</td> </tr> <tr> <td>VOLUME</td> <td>Volume</td> </tr> <tr> <td>LENGTH</td> <td>Length</td> </tr> <tr> <td>WIDTH</td> <td>Width</td> </tr> <tr> <td>HEIGHT</td> <td>Height</td> </tr> </table>	DOC_FORMAT	Document format (free text)	GROSS_WT	Gross weight	NET_WEIGHT	Net weight	VOLUME	Volume	LENGTH	Length	WIDTH	Width	HEIGHT	Height
DOC_FORMAT	Document format (free text)														
GROSS_WT	Gross weight														
NET_WEIGHT	Net weight														
VOLUME	Volume														
LENGTH	Length														
WIDTH	Width														
HEIGHT	Height														
	<p> <b>Example</b></p> <pre>&lt;MA field="MATL_DESC"&gt;   &lt;APPL_ATTRIBUTE name="SAP_DESCRIPTION"/&gt; &lt;/MA&gt;</pre> <p>Attributes can also be transferred from the application to SAP with SAP ECTR for SAP S/4HANA by providing a BAdI at the end of the check-in transaction. All characteristics from the part are transferred to this BAdI (ABAP user exit in the SAP system). SAP data can be set directly in the ABAP code. To transfer the attributes to the SAP BAdI, the following preference variable must be set in the %PLM_INSTDIR%\customize\config\default.txt file:</p> <pre>plm.attributes.send.onCheckin = true</pre>														
	<p><b>XML attributes</b></p> <ul style="list-style-type: none"> <li>• transfer_empty_value="..." true or false (default)</li> </ul> <p>If true, blank characteristic values are also transferred. Make sure that the characteristic value in the SAP system is allowed to be blank.</p>														

### XML Tag for Classification Data

XML Tag	Description
<DIR_CLS>	<p>You can only change the characteristic value assignment of an existing classification on the document. New classifications are not created and the corresponding entries in the XML file are ignored.</p>
	<p> <b>Example</b></p> <pre>&lt;DIR_CLS class="CLASS-XYZ"   classtype="017"   characteristic="BASIC MATERIALS"   transfer_empty_value="true"&gt;   &lt;APPL_ATTRIBUTE name="NX_MATERIAL"/&gt; &lt;/DIR_CLS&gt;</pre> <p>For XML attribute class="..." you can also use "*" instead of a class name. In this case, the classification of the document is searched for a classification that has the characteristic from characteristic="..." and this classification is used for the characteristic value assignment.</p>

XML Tag	Description
	XML-Attribute <code>transfer_empty_value="..."</code> true or false (default), if true, blank characteristic values are also transferred. Make sure that the characteristic value in SAP is allowed to stay blank.
<MA_CLS>	You can only change the characteristic values of an existing classification on the material. New classifications are not created and the corresponding entries in the XML file are ignored. For XML attribute <code>class="..."</code> you can also use "*" instead of a class name. In this case, the classification of the document is searched for a classification that has the characteristic from <code>characteristic="..."</code> and this classification is used for the characteristic value assignment.

#### 4.2.2 Drawing-Specific Attributes

SAP ECTR for SAP S/4HANA supports drawing-specific attributes. In the SAP system, you can set specific characteristics for each sheet in a CAD file. If you have several drawings in a CAD file, you can assign a different descriptive text to each drawing.

#### Preparing drawing-specific attributes in the drawing template file:

1. Switch to draft mode
2. Open/create sheet for example `SHEET_1`
3. Insert a text (note) into the drawing frame: for example description
4. Select text, then
5. Enter name `SAP_DRW_ATTR`
6. Create attribute `SAP_PART_ATTR` with the value `<Part attribute without sheet information>`

Configure the interface so that a part attribute is created for each sheet. The part attribute consists of two parts: the sheet-neutral name and the sheet number.

#### Example

You have two sheets and want to store the description of each sheet in the SAP system. In the SAP system, define the two characteristics `DESCRIPTION:1` and `DESCRIPTION:2`. Configure the mapping file `attributes.xml` so that these attributes are transferred into an application as `SAP_DESCRIPTION:1` and `SAP_DESCRIPTION:2`. When the attributes are updated in an application, SAP ECTR for SAP S/4HANA searches for the drawing texts with the name `SAP_DRW_ATTR`. If a text is found, the name of the part attribute is read from text attribute `SAP_PART_ATTR` and assigned to the drawing name. The text is then referenced to this part attribute and the last digit of the text name is deleted.

#### Note

Variable `UGII_D_ATTR_DRWATTR` in `default.txt` must then be set to `true`.

### 5.1 Roles in SAP ECTR for SAP S/4HANA

You can assign different roles to users. Do not confuse these roles with the SAP PFCG roles. You can control the scope of functions that are available and the default settings depending on the role. The role is stored in environment variable `%PLM_ROLE%`. SAP ECTR for SAP S/4HANA fetches the role-specific definition files.

You can make the following settings for determining the role:

- As a call parameter for `dsc_StartPlm.exe`:  
The call parameter overwrites all other roles.
- Via the selection of the role in the logon dialog via input field `CAD Group`:  
The selection controls the role via the preference variables `sap.logon.cad_group_<CAD group>` in file `SAPlogon.txt`
- Via the parameter from the SAP system:  
If variable `sap.logon.cad_group_<CAD group>` is not defined in `SAPlogon.txt`, SAP ECTR for SAP S/4HANA uses the default parameters.

#### Note

By default, role `FULL` is assigned.

The definition of the role controls the default files to be executed.

### 5.2 Rules for Part File Names

The file names can be configured freely. The following schema is set by default:

```
Document number document type document part _ document version
```

The settings for the part names are made in the DTypes in section `<document_create>` via the XML attributes `filename_template_base="..."` and `filename_template_version="..."`.

**The following placeholders are available:**

- `$(DOCNUMBER)` for the document number
- `$(DOCTYPE)` for the document type
- `$(DOCPART)` for the document part
- `$(DOCVERSION)` for the document version

#### Rules for upper and lower case of file names

It is possible to define whether and which parts of the file name are written in upper or lower case. The following parameters are available for this purpose:

- `_LC`
- `_UC`

Beispiel:

```
$(DOCNUMBER) $(DOCTYPE_UC) $(DOCPART_LC)
```

### 5.3 File Handling

SAP ECTR for SAP S/4HANA is based on a separate file management. File management uses a local or shared data buffer (cache) to minimize the need to transfer data from the SAP data storage system several times. Each primary application can have additional files that are all saved in the SAP system. This function is used to manage additional files (assembly instructions, certificates etc.). The additional files are directly related to the primary document; they do not need a separate management data record. These additional files are stored in a directory that is saved and opened together with the primary file.

If a document is edited, a complex process is executed. The process ensures that all required files are in the session directory in their current version. The process chain tries to make the required files available. Documents that have caused an error in one step are sorted out; the rest of the documents is transferred to the next process step. This ensures that errors are detected and corrected. The documents are in the session directory in their current version and are not transferred again if the process is repeated.

### 5.4 Log Files

Log files enable you to see exactly what is currently happening in SAP ECTR for SAP S/4HANA. The files can be called up via the [Icon Tray](#) menu.

#### 5.4.1 Trace Output

SAP Support uses this output. If software problems occur, you can trace the program flow and quickly and efficiently find a solution.

The scope of the trace output can be set using the trace level. The trace level can be set; the higher the trace level the more detailed the output. In trace level "Trace" (= 3) the status row output from SAP ECTR for SAP S/4HANA is written to the log file. From trace level "Debug" (= 4) onwards, debug output is generated in an extra log file in addition to the normal trace output. Use the debug output for detailed troubleshooting if an error occurs.

Set a low trace level for normal operations to avoid unnecessary writing of large log files with trace information that is not required during normal operation.

You can use environment variable `PLM_CONSOLE` in `plm_setenv.bat` to specify whether the trace output is to be output on the console or in a log file.

The output on the console can be achieved using the following assignment:

```
set PLM_CONSOLE=true
```

The trace level can be set as follows in the `system-default.txt` file:

```
system.log.trace.level = 1
```

#### 5.4.2 BAPI Log Files

The calls of the BAPIs are logged with all parameters and transferred values in the API log files. If you have software problems, the log files let you trace the exact operations.

API log files that end with the letter h belong to a transaction in the foreground.

In SAP ECTR for SAP S/4HANA, you can use the Options dialog to activate and deactivate the generation of API log files. During normal operation, there is no need to write API log files.

### 5.5 Dictionary

SAP ECTR for SAP S/4HANA supports various languages. The dictionary is responsible for displaying the relevant language.

The language can be set using the logon language in the logon dialog.

When the program is started, the language-specific files are imported from the following directory:

```
%PLM_INSTDIR%\basis\dictionary\<language>
```

You can also integrate customer-specific dictionaries. Store the files in the following directory:


```
%PLM_INSTDIR%\customize\dictionary\<language>
```

Definitions from the standard dictionary `ectr.txt` are overwritten by customer-specific definitions, without changes having to be made to the standard dictionary itself.

## 5.6 Material

### 5.6.1 Creating a Material

When you create a material using the SAP GUI, the fields of the GUI can be prefilled. This can be defined in the `default.txt` file.

Preference Variable	Description	Value
<code>plm.mat.number.converter</code>	Specifies the conversion of the material number.	1: Truncate material number or add leading zeros 2: Internal and external numbers are identical 3: Conversion using a BAPI  <class name>: Conversion using the class name specified   Example <code>plm.mat.number.converter = 1</code>
<code>plm.mat.create.multi.adv.fields.support</code>	Specifies the display in the material creation dialog.	true: <b>Basic Data Fields</b> are displayed in the material creation dialog. false: <b>Basic Data Fields</b> are not displayed in the material creation dialog.

### 5.6.2 Creating a Bill of Material

When you create a Bill of Material using the SAP GUI, the fields of the GUI can be prefilled.

This can be defined in the `default.txt` file.

The following sections describe the most important settings that can be made. For more information, see the `default.txt` file.

Default setting	Description
<code>plm.bom.application</code>	Default setting for the corresponding application.
<code>plm.bom.plant</code>	Default setting for the corresponding plant.
<code>plm.bom.alternative</code>	Specifies an alternative plant.
<code>plm.bom.usage</code>	Default setting for the usage.
<code>plm.bom.useChangeNoFromDoc</code>	Specifies whether the change number of the document is used.
<code>plm.bom.initStatus</code>	Default setting for the initialization status.
<code>plm.bom.baseUnit.DE</code>	Default setting for the base unit in the German system environment.
<code>plm.bom.baseUnit.EN</code>	Default setting for the base unit in the English system environment.

Default setting	Description
plm.bom.create	If no Bill of Material is available in SAP PLM 7.0, a new Bill of Material is created.
plm.bom.update	If a Bill of Material is available in SAP PLM 7.0, the Bill of Material may be updated.
plm.bom.posIsCostRel	Specifies whether the BOM position is relevant for Controlling.
plm.bom.posIsProdRel	Specifies whether the BOM position is relevant for Production.
plm.bom.posIsengiRel	Specifies whether the BOM position is relevant for engineering.
plm.bom.posIsSalesRel	Specifies whether the BOM position is relevant for Sales.
plm.bom.posIsPmainRel	Specifies whether the BOM position is relevant for Maintenance.
plm.bom.insertDocIfNoMatIs Available	If no material is assigned, a document is inserted.
plm.bom.deleteIfEmpty	If entries are available, the Bill of Material is deleted.
plm.bom.mapping	Regulates the attribute mapping.
plm.bom.itemkey	You can use this variable to define how the key for the BOM position is determined. Normally, the document key is the key for the BOM position. The document key can be enhanced. In the Bill of Material, different BOM positions are generated for different keys.
plm.bom.userParameter.plant plm.bom.userParameter.usage plm.bom.userParameter.alternative	You can use these variables to user-specifically control the creation of a Bill of Material.

## 5.7 DTypes

The DType is one of the central elements in SAP ECTR for SAP S/4HANA. DTypes enable you to define different behavior patterns of documents with respect to processes, status networks, standard values, and relationships to other SAP objects. The DType can control entire document groups but also specific document types. Various document types can be used in different DTypes (1:n).

You can use the DType settings to control how SAP ECTR for SAP S/4HANA responds:

- When new documents are created
- The properties of the documents

SAP ECTR for SAP S/4HANA checks whether the workstation application, status, document types, and template files specified in a DType exist.

The file with the DType settings is located in the following directory and is called `dtype.xml`:

```
%PLM_INSTDIR%\customize\config
```

If errors occur during the reading of DType settings, you get an error message and SAP ECTR for SAP S/4HANA is terminated.

The structure of this XML file is defined in definition file `dtype_4110.dtd`, which is located in the following directory:

```
%PLM_INSTDIR%\basis\aux-files
```

For each document type, the XML file contains one or more DType definitions, which define either a master document, a dependent document or a template. The general structure of the DType XML file is therefore as follows:

```

<!DOCTYPE DType_definitions SYSTEM "../..//basis/aux-files/dtype_4110.dtd">
<DType_definitions>
  <DType type="...">
    :
    :
  </DType>
  <DType type="...">
    :
    :
  </DType>
  :
  :
</DType_definitions>

```

### 5.7.1 Structure of Element <DType>

The DType element is a structure element that encloses a set of settings for a document or template.

```

<DType
  type="..."
  identifier="..."
  include_template="..."
  application_type="..."
  application_role="..."
  group="..."
  sort_key"...">
  <description ... />
  <icon> ... </icon>
  <program_options ...> ... </program_options>
  <nonmaster_options .../>
  <document_create ...> ... </document_create>
  <material_create ... />
  <primary_application ...> ... </primary_application>
  <additional_files ...> ... </additional_files>
  <allowed_nonmaster ...> ... </allowed_nonmaster>
</DType>

```

### DType attributes

Attribute	Description	Value
type="..."	Type of DType definition	master: DType definition for a master document (default) nonmaster: DType definition for a dependent document template: DType template definition
identifier="..."	Unique ID string of the DType in the entire XML document. This ID must consist of one to five characters.	DType ID
include_template="..."	Optional: If a DType template is used, this is used to identify the DType template used.	DType template ID

Attribute	Description	Value
application_type="..."	Optional: Label of the application that is responsible for this document. This application is used to open the file from SAP ECTR for SAP S/4HANA. Each application must have a unique application type, which is referenced at various points in the settings files. If there is active communication from the application, the application must register itself with this defined application type.	Defined application types for example: ACD: Autocad M10: ME 10 PPT: PowerPoint UGS: NX WRD: Word XLS: Excel
application_role="..."	Optional: Makes it possible to manually set the object type in the RES4field of SAP table DRAW.	Identification of the object type for example: A: Assembly P: Part D: Drawing etc.
group="..."	Optional: Defines the group assignment for master documents. The group assignment is used to organize DTypes in the <a href="#">Create Document</a> dialog. After the selection of the DType group, all DTypes that belong to this group are displayed automatically. The language-specific group name for display in the document creation dialog is determined using the dictionary key <code>plm.dtype.groupname.&lt;Gruppen Identifier&gt;</code> .	Group ID
sort_key="..."	Optional: Defines the sort key that is used for the alphanumeric sorting of the DType entries in the combo box for DType selection.	Sort key for example "A1000"

## DType Sub-elements

All sub-elements can occur several times.

Sub-element	Description	Reference
<code>&lt;description ... /&gt;</code>	Meaningful description of this set of DType settings.	See: Structure of element <code>&lt;description&gt;</code>
<code>&lt;icon&gt; ... &lt;/icon&gt;</code>	Defines the abbreviation that is used to determine the icons to be used in SAP ECTR for SAP S/4HANA.	See: Structure of Element <code>&lt;icon&gt;</code>
<code>&lt;program_options ...&gt;...&lt;/program_options ...&gt;</code>	Different settings that control the behavior of the document in SAP ECTR for SAP S/4HANA.	See: Structure of Element <code>&lt;program_options&gt;</code>
<code>&lt;nonmaster_options ... /&gt;</code>	This element controls the behavior of the dependent document in relation to the master document.	See: Structure of Element <code>&lt;nonmaster_options&gt;</code>

Sub-element	Description	Reference
<code>&lt;document_create...&gt; ... &lt;/document_create&gt;</code>	Settings that specify how the document is created in the SAP system.	See: Structure of element <code>&lt;document_create&gt;</code>
<code>&lt;material_create ... /&gt;</code>	Settings that determine whether and how a material is created for a new document.	See: Structure of Element <code>&lt;material_create&gt;</code>
<code>&lt;primary_application...&gt;... &lt;/primary_application&gt;</code>	Specifies which application is responsible for this master document and defines the rules for checking in and checking out originals.	See: Structure of Element <code>&lt;primary_application&gt;</code>
<code>&lt;additional_files...&gt; ... &lt;/additional_files&gt;</code>	Specifies which additional originals are displayed and can be edited in the <a href="#">Originals</a> container of a document in SAP ECTR for SAP S/4HANA.	See: Structure of Element <code>&lt;additional_files&gt;</code>
<code>&lt;allowed_nonmaster...&gt;... &lt;/allowed_nonmaster&gt;</code>	The unique ID of the dependent document is inserted as the value here.	See: Structure of Element <code>&lt;allowed_nonmaster&gt;</code>

### 5.7.2 Structure of Element `<additional_files>`

This is used to control which originals are displayed and can be edited in the [Originals](#) container of a document in SAP ECTR for SAP S/4HANA. Without this information, all originals with the exception of the original and the container file are displayed on the document and can be changed there. This can be restricted in relation to the visibility and changeability of the originals.

```
<additional_files
  enable_additional_files="..."
  <view_files> ... </view_files>
  <ignore_files> ... </ignore_files>
  <store_files> ... </store_files>
  <insert_allowed> ... </insert_allowed>
</additional_files>
```

Usage: `<DType>`

#### additional\_files attributes

Attribute	Description	Value
<code>enable_additional_files="..."</code>	Optional: Specifies whether additional originals are allowed for this document.	yes: Additional originals are allowed (default) no: No additional originals permitted

#### additional\_files sub-elements

All sub-elements can occur several times.

Sub-Element	Description	Reference
<code>&lt;view_files&gt;...&lt;/view_files&gt;</code>	Specifies which of the additional originals displayed in the <b>Originals</b> container of the document cannot be changed.	See: Structure of Element <code>&lt;view_files&gt;</code>
<code>&lt;ignore_files&gt;...&lt;/ignore_files&gt;</code>	Specifies which additional originals are not displayed in the <b>Originals</b> container of the document.	See: Structure of element <code>&lt;ignore_files&gt;</code>
<code>&lt;store_files&gt;...&lt;/store_files&gt;</code>	When additional files are added to a document, this defines the assignment of the originals to the workstation application. The additional files can be added using drag & drop or a menu function.	See: Structure of Element <code>&lt;store_files&gt;</code>
<code>&lt;insert_allowed&gt;...&lt;/insert_allowed&gt;</code>	Optional: Specifies which files can be added to the document as additional originals from the file system via drag & drop.	See: Structure of element <code>&lt;insert_allowed&gt;</code>

### 5.7.3 Structure of Element `<allowed_actions>`

If the optional tag `<allowed_actions>` is specified, a DType is displayed for an action only if the action in question is listed below.

`show_for_create` must be `true` for this; otherwise, the DType is hidden in any case.

```
<allowed_actions>
  <action name="application_doc_create"/>
  <action name="application_doc_create_multi"/>
  <action name="standalone_doc_create"/>
  <action name="dnd_doc_create"/>
</allowed_actions>
```

Attribute	Description	Values
<code>action name="..."</code>	The name of the action for which the DType is to be displayed should be specified here.	The following values are possible: <ul style="list-style-type: none"> <li><code>application_doc_create</code> Creation dialog for a master/dependent document in CAD.</li> <li><code>application_doc_create_multi</code> Import on demand from CAD.</li> <li><code>standalone_doc_create</code> Normal creation dialog for a master/dependent document in SAP ECTR.</li> <li><code>dnd_doc_create</code> Document creation triggered by drag &amp; drop.</li> </ul>

If the user only wants to see a DType for import on demand, the DType must be configured as follows:

```
<DType ...>
  <program_options ... show_for_create="yes" ...>
```

```

</program_options ...>
  <document_create ...>
    <allowed_actions>
      <action name="application_doc_create_multi"/>
    </allowed_actions>
  </document_create>
  ...
</DType>

```

#### 5.7.4 Structure of Element <allowed\_nonmaster>

Reference to a certain type of the dependent document.

```

<allowed_nonmaster
  DType="..."
  max_count="..."
  master_status_list ...
/>

```

#### allowed\_nonmaster attributes

Defines the dependent documents that may be created for a master document and how many.

Attribute	Description	Value
DType="..."	The unique ID of the definition of the dependent document	ID of the dependent document
max_count="..."	Optional: Number of allowed dependent documents for this master document.	Number of dependent documents. If a number larger than 999 is specified, any number of dependent documents are allowed.
<master_status_list ... />	Optional: Defines the status of the master document for which the dependent document may be created.	One or more status IDs are semicolon-separated for example "AC;IA"

#### 5.7.5 Structure of Element <change\_status>

Element <change\_status> defines a status change from a certain original status to a target status. If you want to force a status change when opening or saving, you can use this definition.

```

<change_status
  from="..."
  to="..."
/>

```

Element <change\_status> does not have any sub-elements.

If no check-out status is defined in the SAP system, the original status must be identical to the target status.

#### change\_status attributes

Attribute	Description	Value
from="..."	Defines the original status of the document. You can set this attribute to change the status to the target status.	Status, for example "ER"

Attribute	Description	Value
to="..."	Defines the target status for the status change.	Status, for example "IA"

### 5.7.6 Structure of Element <checkin\_rules>

This element defines how files in the additional directory and originals in the document are to be handled when saving and whether a forced status change is to be performed.

```
<checkin_rules
  copy_files_before_checkin="..."
  from_application_only="..." >
  >
  <change_status ... />
  <save_direct> ... </save_direct>
  <ignore> ... </ignore>
  <delete_in_sap> ... </delete_in_sap>
  <delete_in_session> ... </delete_in_session>
  <save_local> ... </save_local>
  <replace_primary_file>...</replace_primary_file>
  <scripts> ... </scripts>
</checkin_rules>
```

#### checkin\_rules attributes

Attribute	Description	Value
copy_files_before_checkin="..."	Optional: Specifies whether the files are copied to another directory prior to saving.	yes: The files are copied. no: The files are not copied (default).
from_application_only	Optional: Specifies whether a document may be saved via SAP ECTR.	yes: The document may be saved only via the application. no: The document may be saved via SAP ECTR for SAP S/4HANA. (default)

#### checkin\_rules sub-elements

All sub-elements can occur several times.

Sub-Element	Description	Reference
<change_status ... />	Optional: Specifies the status change that is performed when the document is saved. Always specify an original status. You can define several status change definitions for different original statuses.	See: Structure of element <change_status>
<save_direct>... </save_direct>	Specifies which files in the additional directory are stored as originals on the document when saving.	See: Structure of Element <save_direct>

Sub-Element	Description	Reference
<code>&lt;ignore&gt;...&lt;/ignore&gt;</code>	Specifies which files in the additional directory are ignored during saving. Ignored files only exist in the additional directory and are not available in the SAP system. When the additional directory is cleared, these files are lost.	See: Structure of element <code>&lt;ignore&gt;</code>
<code>&lt;delete_in_sap&gt;...&lt;/delete_in_sap&gt;</code>	Specifies which originals in the document are deleted prior to saving. If the number of originals in document is reduced during check-in, this setting makes sense. A drawing can consist of three sheets. For each sheet, a PDF file has been stored on the document. If you delete a sheet when you change a drawing, the PDF of the deleted sheet remains on the document.	See: Structure of Element <code>&lt;delete_in_sap&gt;</code>
<code>&lt;delete_in_session&gt;...&lt;/delete_in_session&gt;</code>	Specifies which files in the additional directory are deleted after saving. Use this setting for previews that SAP ECTR for SAP S/4HANA accesses via the preview cache after saving. These files no longer need to be available in the additional directory.	See: Structure of Element <code>&lt;delete_in_session&gt;</code>
<code>&lt;save_local&gt;...&lt;/save_local&gt;</code>	Optional: Intended as a subsequent functional enhancement.	
<code>&lt;replace_primary_file&gt;...&lt;/replace_primary_file&gt;</code>	This element defines whether the original of a document can be replaced using drag & drop or a menu function and which files are permissible for this.	See: Structure of element <code>&lt;replace_primary_file&gt;</code>
<code>&lt;scripts&gt; ... &lt;/scripts&gt;</code>	Optional: Execute one or more scripts prior to saving. If no script definition with <code>&lt;scripts&gt;</code> is available, preference variable <code>plm.convert.script.</code> + application type are evaluated prior to saving and the script specified there is executed.	See: Structure of Element <code>&lt;scripts&gt;</code>

### 5.7.7 Structure of Element `<checkout_direct>`

This element defines which of the originals stored directly in the document are transferred into the additional directory for editing/opening the original. The default settings define that, other than the original in the session directory, SAP ECTR for SAP S/4HANA only transfers the originals from the container file into the additional directory. You cannot use this element to transfer the original into the additional directory.

```
<checkout_direct>
  <plmfile.../>
</checkout_direct>
```

## checkout\_direct sub-elements

This sub-element can occur several times.

Sub-Element	Description	Reference
<code>&lt;plmfile ... /&gt;</code>	Specifies the filter conditions for the originals stored in the document that are transferred to the additional directory when the original is opened.	See: Structure of Element <code>&lt;plmfile&gt;</code>

## checkout\_direct additional functions

You can use a corresponding configuration to transfer the additional originals into the session directory.

Comply with the following conditions and make the following additional settings:

- The file name of the additional original must be identical to the file name of the original.
- The file extensions of the original and additional original must differ. This also means that the files receive different workstation applications.
- In the DType definition, `<checkout_direct>` must be defined for the workstation application of the additional original:

```
<checkout_direct>  
  <plmfile workstation_application="{WSAppl}"/>  
</checkout_direct>
```

- In the `default.txt` file, define the following default setting variable:

```
plm.checkout.more.firstlevel.{application_type} =  
{file extension of additional original}  
  
plm.checkout.more.firstlevel.UGS = xls
```

If you define an additional original with these settings, the additional original is handled in a similar fashion to the original. The additional original is stored directly in the session directory and no longer displayed in the [Originals](#) container.

### Note

Originals that are opened for editing using `<checkout_direct>` are hidden in the [Originals](#) container in the SAP ECTR for SAP S/4HANA [Desktop](#) and [Object Browser](#). If SAP ECTR for SAP S/4HANA does not hide the additional originals, the additional originals can still be opened.

The [Originals](#) container normally does not show originals that are copied to the additional directory using the normal [take into editing](#). If the display of all additional originals is explicitly desired, you can force this in the `default.txt` file either DType-specifically or for all documents. To do this, use the following default setting variables:

DType-specific: `plm.sap.original.show.all.{DType} = true`

For all documents: `plm.sap.original.show.all = true`

Remove the [Edit](#) function that opens the additional originals from the context menu for additional originals.

### 5.7.8 Structure of Element `<checkout_rules>`

This element defines how originals of the document are to be handled when opening and whether a forced status change is to be performed.

```
<checkout_rules  
  rename_master_original_at_checkout="..."  
  one_version_in_session_only="..."  
  hardlink_in_session="..."
```

```

setrw_for_checkedin="..."
setrw_for_open="..."
keep_original_extensions="..."
check_references_on_edit="..."
read_structure="..."
>
<change_status.../>
<checkout_direct>...</checkout_direct>
<scripts>...</scripts>
</checkout_rules>

```

### checkout\_rules attributes

Attribute	Description	Value
rename_master_original_at_checkout="..."	Optional: Defines whether the original is renamed according to rules for part names when opened.	yes: Rename the original when opening. no: Retain file name of the original when opening (default).
one_version_in_session_only="..."	Optional: Specifies whether only one version of this document may be in the session directory.	yes: Only one version of the document may be in the session directory (default) no: Several versions of the document are allowed in the session directory.
hardlink_in_session="..."	Optional: Specifies whether a hard link can be used for the original of the document in the session directory or whether a copy must always be generated.	yes: Hard link can be used no: A copy must be generated (default)
setrw_for_checkedin	Optional: Specifies whether a write right is set in the session directory for the original of the document when it is in a secured state. If the write right exists, no hard links may be used even if hardlink_in_session=yes is set.	yes: The write right is set no: The original is read-only (default)
setrw_for_open	Optional: If a document is opened directly from an application, the system checks whether there is a write authorization for the document.	yes: The write authorization is set. (default) no: The original is write protected.
keep_original_extension	Optional: Specifies whether the file extension from the original in the SAP system can be copied or set anew using the PartnameRules.	yes: The file extension is copied from the original in the SAP system (default) no: The PartnameRules can be used to set the file extension anew
check_references_on_edit="..."	Optional: Defines whether the files for all referenced components could be made available in the session directory. The check takes place in	yes: Perform check no: Do not check (default)

Attribute	Description	Value
	the function <code>Open for Editing</code> . If not all required files are located in the session directory, the action is terminated and the document is not opened for editing.	
<code>read_structure</code>	Optional: Specifies whether opening the document results in a structure explosion.	<p><code>auto</code>: Automatic determination whether a structure explosion is executed (default).</p> <p><code>never</code>: Do not perform structure explosion.</p> <p><code>always</code>: Always perform a structure explosion.</p> <p><code>flag</code>: If <code>Structure-Flag</code> is set, perform structure explosion.</p>

### checkout\_rules sub-elements

Sub-element	Description	Reference
<code>&lt;change_status ... /&gt;</code>	Optional: Specifies the status changes that are performed when the document is opened. If the check of the status network for the next reachable status with <code>O</code> flag is deactivated with <code>require_status_with_oflag=no</code> , the settings take effect. For the status change, specify an original status from which this status changes into the target status.	See: Structure of element <code>&lt;change_status&gt;</code>
<code>&lt;checkout_direct&gt;... &lt;/checkout_direct&gt;</code>	Specifies which of the originals stored directly in the document are transferred into the additional directory. In addition to the original, only originals from the container file are transferred into the additional directory.	See: Structure of element <code>&lt;checkout_direct&gt;</code>
<code>&lt;scripts&gt; ... &lt;/scripts&gt;</code>	Optional: Execute one or more scripts during the checkout.	See: Structure of Element <code>&lt;scripts&gt;</code>

### 5.7.9 Structure of Element `<container_file>`

This element defines the properties of the ZIP archive that is stored as a container file in the document.

```
<container_file>
  <plmfile.../>
</container_file>
```

## container\_file sub-elements

Sub-element	Description	Reference
<plmfile ... />	Specifies the properties of the container file. Only the attributes <code>workstation_application</code> and <code>description</code> are relevant for the definition. SAP ECTR for SAP S/4HANA assigns the file name of the container file.	See: Structure of Element <plmfile>

### 5.7.10 Structure of Element <copy\_as>

Different settings that specify how the document responds in case of a `CopyAs` function and how SAP ECTR for SAP S/4HANA responds to this document.

```
<copy_as
  allowed="..."
  ignore_wsappl_list="..."
  ignore_nm_dtype_list="..."
  remove_dirty_flag="..."
  use_initial_version="..."
  copy_material="..."
  alternative="..."
  delete_local_additional="..."
  possible_target_dtype_list="..."
>
  <nm_dtype_change ... />
</copy_as>
```

### copy\_as attributes

Attribute	Description	Value
<code>allowed="..."</code>	Optional: Specifies whether <code>CopyAs</code> is allowed for this DType. The setting also includes the <a href="#">Cloning</a> dialog.	<code>yes</code> : <code>CopyAs</code> is allowed (default) <code>no</code> : <code>CopyAs</code> is not allowed.
<code>ignore_wsappl_list="..."</code>	Optional: Specifies which additional originals in the document are not copied into the copy of the document during copying. The workstation application of the additional originals is used to define this as a semicolon-separated list of IDs for the workstation application.	List with IDs for the workstation application, for example "XML; ZIP"
<code>ignore_nm_dtype_list="..."</code>	Optional: Specifies for a master document, which dependent documents are not copied.	List of DTypes of dependent documents to be ignored. Alternatively, a * can be used to ignore all dependent documents.
<code>remove_dirty_flag="..."</code>	Optional: Specifies whether the dirty flag is removed from the copy or from a cloned part.	<code>yes</code> : The dirty flag is removed. <code>no</code> : The dirty flag is retained (default).

Attribute	Description	Value
<code>use_initial_version="..."</code>	Optional: Specifies the rules for version numbers.	<code>yes</code> : The version number of the copy is set to the initial value (default). <code>no</code> : The version number corresponds to the version number of the source document
<code>copy_material="..."</code>	Optional: Specifies whether the assigned material is also copied when the master document is copied. This material is used as the template for generating a new material. If the dependent document is copied by itself, this also works with a dependent document. This attribute does not affect dependent documents that are copied in the context of the copy transaction to the master document.	<code>yes</code> : Assigned material is copied. <code>no</code> : Material is not copied. (default)
<code>alternative="..."</code>	Optional: Specifies whether <code>CopyAs</code> interprets the new document as an alternative document and only allows the input of the alternative part of the document number.	<code>yes</code> : Interpret as alternative document <code>no</code> : Perform normal copy operation (default)
<code>delete_local_additional="..."</code>	Optional: Specifies whether the additional originals are copied to the new document or deleted during <code>Save Document as...</code> from an application.	<code>yes</code> : The additional originals in the additional directory are not copied. <code>no</code> : The additional originals in the additional directory are copied. (Default)
<code>possible_target_dtype_list="..."</code>	Optional: During copying with a change of DType, this specifies the permissible target DTypes as a semicolon-separated list. If this attribute is not specified, no DType change is possible during copying.	List of permissible DTypes, for example <code>PINT;PEXT;PINTA</code>

### copy\_as sub-elements

This sub-element can occur several times.

Sub-element	Description	Reference
<code>&lt;nm_dtype_change ... /&gt;</code>	For the DType change of the master document, this specifies which DType change is performed for the dependent documents.	See structure of element <code>&lt;nm_dtype_change&gt;</code>

### 5.7.11 Structure of Element <delete\_in\_sap>

This element defines which originals in the document are deleted in the respective transaction. This makes the targeted cleanup of a document possible.

```
<delete_in_sap>
  <plmfile.../>
</delete_in_sap>
```

#### delete\_in\_sap sub-elements

This sub-element can occur several times.

Sub-element	Description	Reference
<plmfile ... />	Specifies the filter condition of the originals to be deleted from the document.	See: Structure of Element <plmfile>

### 5.7.12 Structure of Element <delete\_in\_session>

This element defines which files in the additional directory are deleted in the respective transaction.

```
<delete_in_session>
  <plmfile.../>
</delete_in_session>
```

#### delete\_in\_session sub-elements

This sub-element can occur several times.

Sub-element	Description	Reference
<plmfile ... />	Specifies the filter criteria for the files to be deleted from the additional directory.	See: Structure of Element <plmfile>

### 5.7.13 Structure of Element <description>

You can use <description> to make settings for descriptions for example for the DType or for the template files. You can set any language for the description. The description is displayed in SAP ECTR for SAP S/4HANA according to the logon language. Descriptions can be specified in more than one language. To do so, element <description> is inserted into the file more than once and the corresponding attribute is changed.

```
<description
  language="..."
  text="..."
/>
```

Element <description> does not have any sub-elements.

## description attributes

Attribute	Description	Value
language="..."	Defines the logon language for which the description is valid. The language used for logging on to the SAP system is decisive	Language ID, for example DE: German EN: English
text="..."	Describes the DTypes present.	Any text

### 5.7.14 Structure of Element <document\_create>

Settings that specify how the document is created in the SAP system.

```

<document_create
create_with_SAP_GUI="..."
create_with_SAP_GUI_enabled = "..."
create_multi_allowed = "..."
show_data_pane = "..."
show_classification_pane = "..."
rename_template_filename="..."
characteristic_assignment_master="..."
characteristic_assignment_folder="..."
document_number_assignment="..."
initial_document_number="..."
document_type="..."
initial_document_part="..."
part_editable="..."
initial_document_version="..."
version_editable="..."
document_default_class="..."
document_default_classtype="..."
change_to_after_create_status="..."
filename_template_base="..."
filename_template_version="..."
copy_master_originals="..."
after_create_trigger="..."
attribute_section="..."
cad_flag="..."
characteristic_groups="...">
  <seedfile ...>...</seedfile>
</document_create>

```


## document\_create attributes

Attribute	Description	Value
create_with_SAP_GUI="..."	Optional: The document can be created with or without user interaction. If an user interaction is desired, the SAP GUI will be opened. This is a legacy feature and it is not recommended any more to use this.	yes: SAP GUI is opened upon document creation. no: The document is created without user interaction in the SAP GUI (Default)

Attribute	Description	Value
create_with_SAP_GUI_enabled = "..."	Optional: Activates or deactivates the checkbox "Create with SAP GUI" in the document creation dialog.	yes: The checkbox is active. no: The checkbox is inactive. (Default)
create_multi_allowed = "..."	Optional: Reserved for project-specific mass creation functions for the document.	yes: Mass creation is allowed for the document no: Mass creation is not allowed for this document (Default)
show_data_pane = "..."	Optional: Defines if the metadata panel is displayed in the document creation dialog or not.	yes: The metadata panel is displayed (Default) no: The metadata panel is not displayed
show_classification_pane = "..."	Optional: Defines if the classification panel is displayed in the document creation dialog or not.	yes: The classification panel is displayed (Default) no: The classification panel is not displayed
rename_template_filename = "..."	Optional: Defines if the filename of the main original has to be renamed according to the naming rules when creating a document, or if the filename of the template file should be taken over unchanged for the main original in the document info record. If the value is "no", make sure that the filename of the template file is unique.	yes: The filename of the original is changed when the document is created. (Default) no: The filename of the template file is copied unchanged for the original in the document
characteristic_assignment_master = "..."	Optional: For dependent documents only: Defines whether characteristic values of the master document are to be adopted for the dependent document creation as default values in the input fields of the classification characteristics. All characteristic values for characteristics with the same name are adopted, e.g.: Master: CAD_PRODTYPE Dependent document: CAD_PRODTYPE	yes: Adopt characteristic values no: Do not adopt characteristic values (Default)

Attribute	Description	Value
characteristic_assignment_folder=".. ."	Optional: This setting defines whether characteristic values are to be adopted from the folder, which is preset as target folder when calling the document creation dialog. The characteristic values are entered as default of the input fields for the classification characteristics. All characteristic values will be adopted for characteristics with the same name, e.g.: Folder: CAD_PRODTYPE Document: CAD_PRODTYPE For dependent documents characteristic_assignment_master has priority over this attribute.	yes: Adopt characteristic values no: Do not adopt characteristic values (Default)
document_number_assignment="..."	Defines the number assignment in SAP.	internal: Internal number assignment. SAP assigns the number automatically external: External number assignment. The user has to assign a number exit: The number entered by the user is passed directly to the SAP component. There, the user input can be processed further by a possible user-exit exit_mixed: User pattern is expected and is then processed by the user exit master: The number of the master document is used
initial_document_number="..."	Optional: If the number is assigned internally, the value in this field is irrelevant. If the number is assigned externally, the value is set as prefix in the input field of the document creation dialog. The user is able to edit or delete the prefix.	Prefix of the document number by external number assignment
document_type="..."	Optional: SAP document type for the newly created document. This value can not be changed by the user anymore.	SAP document type
initial_document_part="..."	Optional: SAP part document with a semicolon-separated list of SAP part	SAP part document, e.g.

Attribute	Description	Value
	documents for the newly created document. If a list of document parts has been specified, a part document can then be chosen from a selection list in the document creation dialog. Once it has been created, this part document cannot be changed by the user anymore. In the case of dependent documents the value for the part document of the master document can be taken over with the help of the value "master". If another dependent document is created, the value of the dependent document that was created first will be incremented.	000 or an SAP part document list, e.g. DEU, ENU or master: The part document of the master document is taken over
part_editable="..."	Optional: Defines whether, in the SAP ECTR document creation dialog, the allowed value for the SAP part document of the attribute "initial_document_part" can be changed.	yes: The allowed value can be changed no: The allowed value can not be changed (Default)
initial_document_version="..."	Optional: SAP document version or a semicolon-delimited list of document versions for the newly created document. If a list of document versions has been specified, it is possible to select a document version from the selection list in the SAP ECTR document creation dialog. This value cannot be changed by the user anymore. Values of dependent documents can be adopted using the value "master" for the document version of the master document.	SAP document version, e.g. "00" or list SAP document versions, e.g. "00;AA" master: The document version of the master document is taken
version_editable="..."	Optional: Defines if it is possible to change the attribute "initial_document_version" of the document version in the SAP ECTR document creation dialog	yes: The default value can be changed. no: The default value can not be changed (Default)
document_default_class="..."	Optional: Defines a class which will be assigned to the document in addition to the standard class. If the attribute is specified, the characteristic attribute of this class will be displayed in the document creation dialog.	Language-neutral classname that is to be added to the document, e.g. "EXT_ATTRIB"
document_default_classtype="..."	Optional: Defines which class type of the classification the document_default_class belongs to. Please note: If document_default_class is filled, document_default_classtype must be filled also.	Identifier of the class type in SAP, e.g.: 017 (Default)
change_to_after_create_status="..."	Optional: The newly created document is set to this status. Via the preference variable plm.document.newVersion.changeToAfterCreate Status=false it is possible to suppress this status change when creating a new document	SAP status identifier

Attribute	Description	Value
	<p>version. The initial status of the document is then applied. Only statuses which are defined as follow-up status of the initial status can be set here.</p>	
<p>filename_template_base="..."</p>	<p>Defines how the document version in the filename for the main original is to be composed. In this process, you need to use the following placeholders: \$(DOCNUMBER) for the document number \$(DOCTYPE) for the document type \$(DOCPART) for the part document</p> <p>See also: Chapter 5.2 Rules for Part File Names</p> <p> <b>Warning</b></p> <p>We strongly recommend that all of the above placeholders are used, otherwise you may end up with filenames that are not unique, which can prevent assemblies from loading, for example, or cause other filename conflicts in the session directory.</p>	<p>Composition rule for the filename, e.g.: \$(DOCNUMBER) \$(DOCTYPE) \$(DOCPART)</p>
<p>filename_template_version="..."</p>	<p>Optional: Defines how the document version for the main original is composed in the filename of this document. The following placeholder can be used: \$(DOCVERSION) for the document version. If the name for the main original of this document should be without a version, the attribute stays empty. Please note: If this attribute is set, your CAD system must support loading, otherwise the load rules (latest e.g.) will no longer work.</p>	<p>Composition rule for the document version in the filename</p>
<p>copy_master_originals="..."</p>	<p>Optional, only for dependent documents: Applies originals of master documents to dependent documents. This is a list of workstation applications for the originals of the master document which are taken over to the dependent documents. For this, the originals are copied in an additional directory of the dependent document for being checked in afterwards.</p>	<p>List of workstation applications, e.g. "STP;IGS"</p>
<p>after_create_trigger="..."</p>	<p>Optional: Defines one or more identifiers (semicolon-separated) for an instant conversion that is to be executed directly after the creation of the document. This instant conversion corresponds in a functional sense to the menu function fnc.doc.generate. The key is, as it is the case in the menu function, part of a series of preference variables that determine the conversion process in a more accurate manner. If several identifiers are specified, the related conversion processes are then handled in a successive manner. The configuration of this instant conversion in SAP</p>	<p>Identifier of the conversion process, e. g. "REGEN"</p>

Attribute	Description	Value
	ECTR is described in the administrator's manual in chapter "Configuration of customer-specific instant conversions".	
attribute_section="..."	Optional: Defines which attribute section from the attribute-mapping files are synchronized in the document creation process.	Name of the section
cad_flag="..."	Optional: Defines, if the newly created document has a CAD flag.	yes: The document has a CAD flag. (Default) no: The document has no CAD flag.
characteristic_groups="..."	Optional: A semicolon-separated list of characteristic groups to limit characteristics displayed in the document creation window. This way, the display of characteristics in the document creation window is filtered, and becomes smaller and more accessible. Document creation can be sped up this way.	List of groups, e.g. chrGrp1;chrGrp 2

#### document\_create sub-elements

This sub-element can occur several times.

Sub-element	Description	Reference
<code>&lt;seedfile ...&gt;... &lt;/seedfile&gt;</code>	Settings for the template file. You can assign several template files.	See: Structure of Element <code>&lt;seedfile&gt;</code>

This sub-element can occur several times.

Sub-element	Description	Reference
<code>&lt;allowed_actions &gt;... &lt;/allowed_actions &gt;</code>	Settings for the template file. You can assign several template files.	See: Structure of Element <code>&lt;seedfile&gt;</code>

#### 5.7.15 Structure of Element `<first_level_files>`

For this element to be used, the element `<additional_files>` must be maintained.

This element defines which additional originals are stored alongside the original in the document in the session directory during checkout. During saving, the marked additional originals are fetched from the session directory along with the original and stored as separate additional originals in the document.

```
<first_level_files>
  <plmfile.../>
</first_level_files>
```

## first\_level\_files sub-elements

This sub-element can occur several times.

Sub-element	Description	Reference
<code>&lt;plmfile ... /&gt;</code>	Specifies the properties of the container file. Only the attributes <code>workstation_application</code> and <code>description</code> are relevant in the definition. SAP ECTR for SAP S/4HANA assigns the file name of the container file.	See: Structure of Element <code>&lt;plmfile&gt;</code>



### Example

```
<primary_application primary_workstation_application="SED"  
attribute_section="STDMOD">  
  
<first_level_files>  
  <plmfile extension="cfg"  
    workstation_application="CFG"  
    description="Assembly config file"/>  
</first_level_files>
```

The definition of a workstation application in `<first_level_files>` ensures that files are opened with this workstation application. You do not need to list the workstation application in the `checkout_direkt` list.

### 5.7.16 Structure of Element `<icon>`

Defines the abbreviation that is used to determine the icons to be used in SAP ECTR for SAP S/4HANA.

```
<icon>  
  <icondef ... />  
</icon>
```

The `<icon>` element does not have any attributes.

## icon sub-elements

This sub-element can occur several times.

Sub-element	Description	Reference
<code>&lt;icondef.../&gt;</code>	Settings for the position and display of the icons.	See: Structure of element <code>&lt;icon_def&gt;</code>

### 5.7.17 Structure of Element `<icondef>`

Element `<icondef>` determines the display of icons for a document. Since a document may have icons, `<icondef>` is specified for each icon item.

```
<icondef  
  icon_position = "..."  
  icon_group = "..."  
  use_status_color = "..."  
  icon_identifier = "..."  
  referenced_object = "..."  
>
```

Element `<icondef>` does not have any sub-elements.

## icondef attributes

Attribute	Description	Value
icon_position="..."	Position of the icon in the document entry in the tree.	1 (far left) ... 6 (far right)
icon_group = "..."	Identification of an icon group. You can use different icons depending on the status.	Icon group ID, for example default
use_status_color="..."	Activates (yes) or deactivates (no) the icon color. The attribute enables the correct display in the filter lists.	yes: The icon ID is supplemented with the icon color (default) no: The icon ID is not supplemented.
icon_identifier="..."	Path and name for determining the icon to be used for this icon position.	Icon name with placeholders for the automatic icon determination (see also SAP ECTR for SAP S/4HANA)
referenced_object="..."	Optional: Specifies whether the document status or material status is to be used in the determination of the icon colors. For more detailed explanations, refer to the icon color change for a status change	document: The document status is used material: The material status is used

### 5.7.18 Structure of Element <ignore>

This element defines which files are ignored in the respective transaction.

```
<ignore>
  <plmfile.../>
</ignore>
```

### ignore sub-elements

This sub-element can occur several times.

Sub-element	Description	Reference
<plmfile ... />	Specifies the filter conditions for the files to be ignored.	See: Structure of Element <plmfile>

### 5.7.19 Structure of Element <ignore\_files>

This is used to specify which of the additional originals in the document in the [Originals](#) container of a document are not displayed in SAP ECTR for SAP S/4HANA.

```
<ignore_files>
  <plmfile.../>
</ignore_files>
```

### ignore\_files sub-elements

This sub-element can occur several times.

Sub-element	Description	Reference
<code>&lt;plmfile ... /&gt;</code>	Defines with the help of the workstation application which additional originals are not displayed. Therefore only the filter attribute <code>workstation_application</code> can be used in <code>&lt;plmfile&gt;</code> . Other filter attributes get ignored.	See: Structure of Element <code>&lt;plmfile&gt;</code>

### 5.7.20 Structure of Element `<insert_allowed>`

This is used to specify which files can be added to the document as additional originals from the file system via drag & drop. If a file is not allowed, a corresponding error message is output in the status area of SAP ECTR for SAP S/4HANA.

```
<insert_allowed>
  <plmfile.../>
</insert_allowed>
```

#### Insert\_allowed sub-elements

This sub-element can occur several times.

Sub-element	Description	Reference
<code>&lt;plmfile ... /&gt;</code>	Specifies which files can be added to the document as additional originals from the file system via drag & drop.	See: Structure of Element <code>&lt;plmfile&gt;</code>

### 5.7.21 Structure of Element `<material_create>`

Settings for creating a material for a new document, e.g.:

```
<material_create
  material_creation_method="..."
  material_number_assignment="..."
  material_type="..."
  material_template="..."
  material_plant_assignment="..."
  material_default_class="...">
<material_master_template ... />
</material_create>
```

#### material\_create Attributes

Attribute	Description	Value
<code>material_creation_method="..."</code>	Optional: Settings specifying whether and how a material is created for a new document.	<p><code>no</code>: No material is created for the document. A material cannot be created subsequently for this document using SAP ECTR for SAP S/4HANA.</p> <p><code>gui</code>: Material creation is controlled and executed using the SAP GUI.</p> <p><code>dark</code>: The material is created by SAP ECTR for SAP S/4HANA. The user cannot intervene in this process.</p>

Attribute	Description	Value
		<p><code>deferred</code>: No material is created for the document. A material can be created subsequently for this document using SAP ECTR for SAP S/4HANA.</p> <p><code>linkOnly</code>: No material is created for the document. An object link from the document to the material (which does not yet exist) is created. The value <code>linkOnly</code> only makes sense with the attribute <code>material_number_assignment="master"</code>.</p>
<code>material_number_assignment="..."</code>	Optional: Defines the material number assignment in the SAP system. The values are identical to those for the document number assignment.	<p><code>master</code>: The number is copied from the master document.</p> <p><code>internal</code>: internal number assignment. The SAP system assigns the number automatically.</p> <p><code>external</code>: external number assignment. The user has to specify the material number.</p> <p><code>exit</code>: The number is assigned via a user exit.</p>
<code>material_type"..."</code>	Optional: Defines the material type.	Internal material type from the SAP system.
<code>material_template="..."</code>	Material template, which can be specified. The material is created using this material template. The basic data and classification are copied.	Internal material number of the template material in the SAP system.
<code>material_plant_assignment="..."</code>	Optional: The plant assigned to the material created.	<p>For valid plant values, refer to your SAP system.</p> <p><b>i</b> Note As soon as a template is specified, only basic data 1 and basic data 2 are created. The plant is not maintained. The parameter <code>material_plant_assignment</code> is ignored.</p>
<code>material_default_class = "..."</code>	Optional: The default class for the material.	<p>For valid class values, refer to your SAP system.</p> <p><b>i</b> Note In order to use this DType setting, the variable <code>plm.mat.classify = true</code> must be set in the file <code>%PLM_INSTDIR%\customize\config\default.txt</code>.</p>

### 5.7.22 Structure of Element <material\_master\_template>

Properties of a material that is created for a document.

```
<material_master_template
  material_number_assignment="..."
  material_template="..."
  material_plant_assignment="..."
  material_default_class="..."/>
```

## material\_master\_template Attribute

Attribute	Description	Value
material_number_assignment="..."	Optional: Defines the material number assignment in the SAP system. The values are identical to those for the document number assignment.	<p>master: The number is copied from the master document.</p> <p>internal: internal number assignment. The SAP system assigns the number automatically.</p> <p>external: external number assignment. The user has to specify the material number.</p> <p>exit: The number is assigned via a user exit.</p>
material_template="..."	Material template, which can be specified. The material is then created using this material template. The basic data and classification are copied.	Internal material number of the template material in the SAP system.
material_plant_assignment="..."	Optional: The plant assigned to the material created.	<p>For valid plant values, refer to your SAP system.</p> <p><b>i</b> Note As soon as a template is specified, only basic data 1 and basic data 2 are created. The plant is not maintained. The parameter <code>material_plant_assignment</code> is ignored.</p>
material_default_class = "..."	Optional: The default class for the material	<p>For valid class values, refer to your SAP system.</p> <p><b>i</b> Note In order to use this DType setting, the variable <code>plm.mat.classify = true</code> must be set in the file <code>%PLM_INSTDIR%\customize\config\default.txt</code></p>

### 5.7.23 Structure of Element <new\_version>

Settings for creating new versions of documents:

- In which status is the creation of a new version allowed
- Which originals are copied into the new version
- Which dependent documents receive new versions

```
<new_version
  status_list="..."
  ignore_wsappl_list="..."
  ignore_nm_dtype_list="..."
  remove_dirty_flag="..."
  rename_master_original="..."
  delete_local_additional="..."
  take_instances_for_generic="..."
/>
```

## new\_version attributes

Attribute	Description	Value
<code>status_list="..."</code>	Optional: Creating new versions of a document only makes sense in certain statuses. You can use a semicolon-separated list of status IDs in this attribute to define which statuses are allowed.	List with status IDs, for example "AC; FR"
<code>ignore_wsappl_list="..."</code>	Optional: Specifies which additional originals in the document are not copied into the new version of the document when the new version is created. This is defined as a semicolon-separated list of workstation application IDs using the workstation application of the additional originals.	List with workstation application IDs, for example "XML; ZIP"
<code>ignore_nm_dtype_list="..."</code>	Optional: Specifies which dependent documents are discarded when a new version of the master document is created. The dependent documents to be ignored are specified as a semicolon-separated list of dependent document types.	List of DTypes to be ignored, for example: "DRAW; DRWNV" Alternatively, a * can be used to ignore all dependent documents.
<code>remove_dirty_flag="..."</code>	Optional: Specifies whether the dirty flag is removed from the new version.	<b>yes:</b> The dirty flag is removed from the new version. <b>no:</b> The dirty flag is retained in the new version. (Default)
<code>rename_master_original="..."</code>	Optional: Specifies whether the original receives a file name according to the part rules for this DType or whether the previous file name is retained unchanged when a new version is created. If the file name of the original was assigned freely and the document is used as a component in a document structure, this attribute is used. When the version is created, the file name is formed from the parts of the document key by applying the part rules. The file names of the individual versions only differ in their version numbers.	<b>yes:</b> The file name is formed using the <code>PartNameRules</code> . (Default) <b>no:</b> The file name of the predecessor version is copied unchanged.
<code>delete_local_additional="..."</code>	Optional: Specifies whether the additional originals are copied to the new version of the document or deleted during version creation.	<b>yes:</b> The additional originals in the additional directory are not copied. <b>no:</b> The additional originals in the additional directory are copied. (Default)

Attribute	Description	Value
take_instances_for_generic="..."	Optional: Specifies whether the entire part family group (generic part + variants) is included when a new version of the generic part is created.	yes: The variants are included when a new version of the generic part is created. no: The variants are not included when a new version of the generic part is created. (default)

#### 5.7.24 Structure of Element <nm\_dtype\_change>

When master documents are copied and their DTypes are changed at the same time, this element defines which DType changes are made when the corresponding dependent documents are copied.

#### Attention

If no DType change is performed on the document, then no DType change is performed on the dependent document.

```
<nm_dtype_change
  for_target_master_dtype="..."
  from="..."
  to="..."
/>
```

#### nm\_dtype\_change attributes

Attribute	Description	Value
for_target_master_dtype="..."	Defines the destination DType of the master document for which the DType change of the dependent document is performed.	Destination DType of the master document.
from="..."	Defines the source DTypes of the dependent document during copying.	Source DTypes of the dependent document.
to="..."	Defines the destination DType that the dependent documents receive during copying.	Destination DType of the dependent documents.

#### 5.7.25 Structure of Element <nonmaster\_options>

This element controls the behavior of the dependent document in relation to the master document.

The element <nonmaster\_options> has no sub-elements.

```
<nonmaster_options
  depends_on_master="..."
  link_to_master_material="..."
  create_mastercomp_name="..."
  copy_as_use_master_description="..."
  new_version_nonmaster_alone="..."
  document_part_unique="..."
  new_version_master_version="..."
/>
```

## nonmaster\_options Attribute

Attribute	Description	Value
<code>depends_on_master="..."</code>	Optional: Specifies whether the content of the dependent document depends on the content of the master document. In case of a dependency, SAP ECTR for SAP S/4HANA checks whether the master document has been saved after the dependent documents. If this applies, the content of the dependent document is obsolete and flag <code>NM_OLD</code> is set to 1. This flag is available as a placeholder <code>\$(NM_OLD)</code> for the icon definition.	<code>yes</code> : The content of the dependent document depends on the content of the master document. <code>no</code> : The content of the dependent document does not depend on the content of the master document. (Default)
<code>link_to_master_material="..."</code>	Optional: Specifies whether the dependent document receives a link to the material of the master document.	<code>yes</code> : An object link is created from the material of the master to this dependent document. <code>no</code> : Do not create a link to the material. (default)
<code>create_mastercomp_name="..."</code>	Optional: Specifies the name of the reference to the master document in the template file in the document creation of the dependent document. SAP ECTR for SAP S/4HANA recognizes the reference and distinguishes the reference from existing references. In the template, the reference to the master document is merely a dummy that is changed during the document creation for the actual master document. If this attribute is not set, the value of the preference variable is used.	Name of the reference to the master document in the template file, for example "mastermodel.prt".
<code>copy_as_use_master_description="..."</code>	Optional: Specifies whether the naming of the assigned dependent documents is also changed when a master document is copied with <code>Copy as</code> .	<code>yes</code> : Copy changes to the description of the master document to the dependent documents. <code>no</code> : Do not copy changes to the description of the master document to the dependent documents. (Default)
<code>new_version_nonmaster_alone="..."</code>	Optional: Specifies whether dependent documents may have versions created independently of the master document.	<code>yes</code> : Versions may be created for dependent documents alone. <code>no</code> : Versions of dependent documents are only created together with versions for the corresponding master document. (default)
<code>document_part_unique="..."</code>	Optional: Specifies according to which rule the document part of the dependent document is determined for different versions of the master document. The following two options are available here:	<code>yes</code> : Rule 1 (default). <code>no</code> : Rule 2

Attribute	Description	Value
	<p>1) Incrementation of the document part number under consideration of all the document part numbers of the dependent documents in all versions of the master document. Each new dependent document starts with a unique document part number of its version chain in the version of the master document in which it was generated.</p> <p>2) Incrementation of the document part number under consideration of the document part numbers only on the version of the master document for which the dependent document is created. For each version of the master document, a continuous chain of document part numbers is achieved for the dependent documents.</p>	
<pre>new_version_master_version ="..."</pre>	<p>Optional: Determines whether a new version of a dependent document is attached to the same, possibly obsolete, master document to which the "old" template version of the dependent document is also attached, or whether a different behavior is desired.</p>	<p><code>as_saved</code> (default): A new version of a dependent document is attached to the same, possibly obsolete, master document to which the "old" template version of the dependent document is also attached.</p> <p><code>latest</code>: A new version of a dependent document is attached to the latest version of the master document.</p> <p><code>latest_released</code>: Attaches the dependent document to the latest released version of the master document.</p>

### 5.7.26 Structure of Element `<param>`

This element specifies a transfer parameter for a script.

```
<param
  name="..."
  value="..."
/>
```

Element `<param>` does not have any sub-elements.

#### param attributes

This sub-element can occur several times.

Attribute	Description	Value
<pre>name="..."</pre>	<p>Defines the name of the transfer parameter.</p>	<p>Parameter name, for example "-i"</p>

Attribute	Description	Value
value="..."	Optional: Defines the value of the transfer parameter. You can use the following placeholders here. SAP ECTR for SAP S/4HANA replaces the placeholders with the actual values at the time it is executed: \$(add_dir) = absolute path to the additional data directory \$(masterfile) = absolute path to the master file \$(session) = absolute path to the session directory \$(doc_type) = document type \$(doc_nr) = document number \$(doc_ver) = document version \$(doc_part) = document part \$(apptype) = application type, for example "UGS" \$(dtype) = DType \$mode = Time of JT creation	Parameter value, for example "\$ (add_dir) \report.pdf" points to file report.pdf in the additional data directory Parameter values for "\$mode": <ul style="list-style-type: none"> <li>• checkout</li> <li>• checkin</li> <li>• save</li> <li>• create</li> <li>• import</li> </ul>

### 5.7.27 Structure of Element <params>

This element encloses the set of all definitions of transfer parameters for a script.

```
<params>
  <param ... />
</params>
```

#### params sub-elements

This sub-element can occur several times.

Sub-element	Description	Reference
<param ... />	Specifies a transfer parameter for a script. The transfer parameters are transferred to the script in the order of the <param> definitions.	See: Structure of element <param>

### 5.7.28 Structure of Element <plmfile>

This element defines the filter attributes for originals in the document and in the file system.

```
<plmfile
  extension="..."
  filename_pattern="..."
  filesize_greater_than
  workstation_application="..."
  description="..."
  delete_always="..."

  doctype="..."
/>
```

## plmfile attributes

Attribute	Description	Value
extension="..."	Optional: Filters according to the extensions for file names in the file system. Case sensitivity is ignored.	File name extension for example "gif"
filename_pattern="..."	Optional: Filters according to file names using Regular Expressions. Case sensitivity is taken into account but can be deactivated in the Regex syntax. <b>i</b> Note Either extension or filename_pattern is used.	Regular Expression for file names, for example "plm_metainf-sap.*"
workstation_application="..."	Optional: Filters originals in the document according to the workstation application. Only for element store_file: Specifies the assignment of the workstation application to the additional original.	Identification of a workstation application for example "UGI"
description="..."	Optional: Filters originals in the document according to their description.	Descriptions of originals for example: "container file"
delete_always="..."	Optional: Only for element delete_in_sap: Deletion of the additional originals in the document prior to check-in. SAP ECTR for SAP S/4HANA does not take into account available additional originals that replace the additional original available on the document.	yes: Always delete found additional originals in the document no: Delete originals in the document, SAP ECTR for SAP S/4HANA replaces the originals with corresponding files in the additional directory (default)
filesize_greater_than	Optional: Filters according to files that are larger than the specified file size. To ensure the size information values are easy to read, you can use letters such a K, M, and G after the numeric value as follows, for example: filesize_greater_than="500K" ⇒ 500 KB filesize_greater_than="100M" ⇒ 100 MB filesize_greater_than="1G" ⇒ 1 GB If no letter is specified, counting is done in bytes.	File size in: Bytes Kilobytes Megabytes Gigabytes
doctype="..."	Optional: Nur für Element insert_allowed: Legt die SAP Dokumentart fest, für die diese Einschränkung gilt. Wenn mehrere Dokumentarten in SAP den gleichen DType im SAP ECTR for SAP S/4HANA besitzen, wird dieses Attribut benötigt. Ansonsten kann dieses Attribut entfallen.	Kennung der Dokumentart im SAP-System, z.B. "UGM"

### 5.7.29 Structure of Element <primary\_application>

Specifies which application is responsible for this master document (primary application) and defines the rules for check-in and check-out of originals.

```
<primary_application
  primary_workstation_application="..."
  attribute_section="..."
  update_attributes_for_ro="..."
  update_attributes_for_rw="..."
  update_attributes_for_dirty="..."
  use_relative_path="..."
  description="..."
>
  <container_file>...</container_file>
  <checkout_rules...>...</checkout_rules>
  <checkin_rules>...</checkin_rules>
  <first_level_files>...</first_level_files>
</primary_application>
```

#### primary\_application attributes

Attribute	Description	Value
primary_workstation_application="..."	Optional: SAP workstation application of the original.	Identification of the SAP workstation application for example "NXI"
attribute_section="..."	Optional: Name of a section in the attribute mapping file in which the attribute exchange between the SAP system and the application is defined.	Name of the section
update_attributes_for_ro="..."	Optional: Defines the attribute update for <b>open read-only</b> . With the introduction of these DType settings, the following preference variables are no longer taken into account: plm.attributes.updateAlways plm.attributes.updateAfterCheckOutSet plm.attributes.updateAfterCheckOutView plm.attributes.sendUpdateAttributesForRW . <DTYPE> plm.attributes.sendUpdateAttributesForRO . <DTYPE>	yes: The attribute update is performed. no: The attribute update is not performed (default).
update_attributes_for_rw="..."	Optional: Defines the attribute update for <b>open for editing</b> .	yes: The attribute update is performed. no: The attribute update is not performed (default).

Attribute	Description	Value
<code>update_attributes_for_dirty="..."</code>	Optional: Defines the attribute update in case there is a dirty flag.	yes: The attribute update is performed. (default). no: The attribute update is not performed.
<code>use_relative_path="..."</code>	Optional: Defines whether a relative path is defined for a document with this DType and whether it is to be used upon edit.	yes: A relative path is being used no: No relative path is being used. (default).
<code>description="..."</code>	Optional: Specifies a description of the master original for new creations.	Desired name of the description

 **Warning**

If you use the attribute `update_attributes_for_ro`, performance problems can occur when components are loaded in the authoring system.

 **Warning**

The attribute `update_attributes_for_ro` is not supported by all authoring system integrations. For more information, refer to the documentation of the relevant integration.

 **Warning**

Subsequent modification of the `description` attribute does not change the description of existing documents. Only new documents created after the attribute has been changed are affected.

### primary\_application sub-elements

All sub-elements can occur several times.

Sub-element	Description	Reference
<code>&lt;container_file&gt;...&lt;/container_file&gt;</code>	Specifies the properties of the container file for this document definition.	See: Structure of Element <code>&lt;container_file&gt;</code>
<code>&lt;checkout_rules...&gt;...&lt;/checkout_rules&gt;</code>	Specifies how the original of the document is handled during checkout.	See: Structure of element <code>&lt;checkout_rules&gt;</code>

Sub-element	Description	Reference
<code>&lt;checkin_rules&gt;...&lt;/checkin_rules&gt;</code>	Specifies how files in the additional directory and originals in the document are handled during check-in.	See: Structure of Element <code>&lt;checkin_rules&gt;</code>
<code>&lt;first_level_files&gt;...&lt;/first_level_files&gt;</code>	Specifies which additional originals are stored alongside the original in the document in the session directory during checkout. During saving, the marked additional originals are fetched from the session directory along with the original and stored as separate additional originals in the document.	See: Structure of Element: <code>&lt;first_level_files&gt;</code>

### 5.7.30 Structure of Element `<program_options>`

Different settings that specify how the document responds in SAP ECTR for SAP S/4HANA or how SAP ECTR for SAP S/4HANA responds to this document.

```

<program_options
  show_for_create="..."
  has_primary_application="..."
  dirty_flag="..."
  attempt_status_change_on_checkin="..."
  compare_docbom_on_checkin="..."
  require_status_with_oflag="..."
  function_group="..."
  display_group="..."
  customer_data_section="..."
  document_data_class="..."
  document_data_classtype="..."
  bod_mapping_section="..."
  rename_additional="..."
  checkout_edit_allowed="..."
  dnd_nonmaster_creation_allowed="..."
  auto_adjust_application_role="..."
>
  <copy_as ...>...</copy_as>
  <new_version ... />
  <status_transition.../>
</program_options>

```

#### program\_options attributes

Attribute	Description	Value
<code>show_for_create="..."</code>	Optional: Specifies whether this DType is visible in the combo box of available document types in the <a href="#">Create Document</a> dialog.	yes: The document type is visible. (default) no: The document type is not visible. (default)
<code>has_primary_application="..."</code>	Optional: Specifies whether the document is a normal document with an original and assigned workstation application or a container object for additional originals without an explicit original.	yes: The document has an original with an assigned workstation application no: The document is a container object for

Attribute	Description	Value
		additional originals (default)
dirty_flag="..."	Optional: Specifies whether the newly created document has a dirty flag.	yes: The document has a dirty flag. (default) no: The document does not have a dirty flag.
attempt_status_change_on_checkin="..."	Optional: Specifies whether SAP ECTR for SAP S/4HANA performs a status change during check-in or whether the SAP system performs a status change.	yes: Status changed by SAP ECTR for SAP S/4HANA (default) no: Status change executed by SAP if necessary
compare_docbom_on_checkin="..."	Optional: Specifies whether SAP ECTR for SAP S/4HANA performs a document structure reconciliation during check-in or whether the document structure is always written and completely.	yes: a document structure reconciliation is performed. no: The document structure is always written anew completely. (Default)
require_status_with_oflag="..."	Optional: Specifies whether a check for the original processing status (O flag) is performed when a document is checked out for editing.	yes: The check is performed (default). no: The check is not performed.
function_group="..."	Optional: Specifies the function group to which the document belongs. For documents with a shared function group, for example, separate context menus can be defined.	Identifier of the function group, for example DOC for Word documents
display_group="..."	Optional: Specifies the display group for the document. For documents with the same display group, define a shared tree layout in the <a href="#">Desktop</a> and properties layout in the <a href="#">Object Browser</a> using the layout manager. If the display group is not defined, the <a href="#">Function Group</a> is used as the identifier for the common layout.	Identifier of the display group, for example MSODOCS for Office documents
customer_data_section="..."	Optional: Name of a section in the attribute mapping file in which customer-specific data is defined. The data is displayed in the <a href="#">Object Browser</a> under the <a href="#">Customer Data</a> tab.	Name of the section
document_data_class="..."	Optional: Specifies which classification from the SAP system is used for the	Name of the class in the SAP system

Attribute	Description	Value
	configuration of the following input fields in the document creation dialog: <ul style="list-style-type: none"> <li>* Authorization Group</li> <li>* Lab/Office</li> <li>* Change Number</li> <li>* CM Relevance</li> </ul>	
document_data_classtype="..."	Optional: Defines the class type of the classification from document_default_class.	Identification of the class type in SAP for example: 017 for document classes 001 for material classes
bod_mapping_section="..."	Optional: Defines an attribute section in the attributes-to-sap.xml file. If the attribute section exists in an attributes-to-sap.xml file, the <APPL_ATTRIBUTE/> rules it contains are applied to the attributes of the integration (from the plm_metainf-cad.xml).	Name of the attribute section.
rename_additional="..."	Optional: For an identical part string such as the file name of the original, this specifies whether the file names of additional originals are renamed when the document is copied or a version is created.	yes: File names of additional originals are changed. (Default) no: File names of additional originals remain unchanged.
checkout_edit_allowed="..."	Optional: Specifies whether a document can be edited. This setting can be used for documents that are derived from a source document whose content is kept up-to-date.	yes: The document can be edited. (Default) no: The document cannot be edited.
dnd_nonmaster_creation_allowed="..."	Optional: Specifies whether the creation of dependent documents by means of drag & drop is allowed for this document. If this is set to "yes" there is a check for a suitable dependent document type for the file when a file is dragged & dropped from the file system. If this applies, the document creation dialog is called for the dependent document, rather than storing an additional original in the document. This option is only permissible for master documents.	yes: Dependent documents can be created using drag & drop. no: Drag & drop of a file from the file system generates an additional original on the document. (Default)

Attribute	Description	Value
auto_adjust_application_role="..."	Optional: Activates or deactivates the automatic adaptation of the application role in RES4.	Yes: On (default) Fitting the condition of the assembly / part No: Off

### program\_options sub-elements

Sub-element	Description	Reference
<copy_as ... />	Specifies how SAP ECTR for SAP S/4HANA responds when this DType is copied, which dependent documents are also copied and whether a DType change takes place when copying.	See: Structure of element <copy_as>"
<new_version ... />	Specifies how SAP ECTR for SAP S/4HANA responds when versions of this DType are created and for which dependent documents versions are also created.	See: Structure of element <new_version>
<status_transition...>	Specifies how the post status list can be filtered.	See: Structure of Element <status_transition>

### 5.7.31 Structure of Element <replace\_primary\_file>

This element defines whether the original of a document can be replaced using drag & drop or a menu function and which files are permissible for this. When you replace an original, you always get a confirmation dialog, which has to be confirmed. If element <replace\_primary\_file> is not defined, the original of a document cannot be replaced using drag & drop or a menu function.

```
<replace_primary_file>
dirty_flag="..."
  <plmfile.../>
</replace_primary_file>
```

### replace\_primary\_file Attribute

Attribute	Description	Values
dirty_flag="..."	Optional: Defines whether a dirty flag is set when a file is replaced via drag&drop. This is only relevant when replacing via drag&drop is allowed.	yes: Dirty Flag is set. no: Dirty Flag is not set.(Standard)

## replace\_primary\_file sub-elements

Sub-element	Description	Reference
<code>&lt;plmfile ... /&gt;</code>	Specifies which files are permissible for replacing the original. Attributes <code>extension</code> and <code>filename_pattern</code> are relevant here.	See: Structure of Element <code>&lt;plmfile&gt;</code>



### Example

```
<primary_application primary_workstation_application="UGI"
    attribute_section="STDMOD">
    ...
    <checkin_rules>
        ...
        <replace_primary_file>
            dirty_flag="yes"
            <plmfile filename_pattern=".*\prt"/>
        </replace_primary_file>
        ...
    </checkin_rules>
    ...
```

### 5.7.32 Structure of Element `<save_direct>`

This element defines which files in the additional directory are stored as originals in the document. By default, files in the additional directory are stored within the container file in the document.

```
<save_direct>
  <plmfile.../>
</save_direct>
```

## save\_direct sub-elements

This sub-element can occur several times.

Sub-element	Description	Reference
<code>&lt;plmfile ... /&gt;</code>	Specifies the filter conditions for the files to be stored directly in the document. The two attributes <code>extension</code> and <code>filename_pattern</code> are relevant here.	See: Structure of Element <code>&lt;plmfile&gt;</code>

### 5.7.33 Structure of Element `<save_local>`

```
<save_local>
  <plmfile.../>
</save_local>
```

## save\_local sub-elements

This sub-element can occur several times.

Sub-element	Description	Reference
<code>&lt;plmfile ... /&gt;</code>	Specifies the filter conditions for the files to be stored directly in the document. The two attributes <code>extension</code> and <code>filename_pattern</code> are relevant here.	See: Structure of Element <code>&lt;plmfile&gt;</code>

### 5.7.34 Structure of Element `<script>`

This element defines a script to be executed as well as any error conditions or transfer parameters that must be taken into account.

```
<script
  name="..."
  pre_exit_code="..."
  fail_on_error="..."
  inproc="..."
  >
  <params>...</params>
</script>
```

#### Script attributes

Attribute	Description	Value
<code>name="..."</code>	Specifies the name of the script to be executed.	File name of the script to be executed, for example <code>"report.bat"</code>
<code>pre_exit_code="..."</code>	Optional: If several scripts are executed in a row, you use this attribute to generate a dependency between the execution of this script and the exit code of the previously executed script.	<code>false</code> : If the previously executed script returned an exit code $\neq 0$ , the script is executed. <code>true</code> : If the previously executed script returned an exit code = 0, the script is executed. <code>nn</code> (for example 99): If the previously executed script returned an exit code = 99, the script is executed.
<code>fail_on_error="..."</code>	Optional: In case of an error, this specifies whether additional scripts are executed or the overall processes is terminated completely.	<code>yes</code> : Terminate execution <code>no</code> : Do not terminate execution (default)
<code>inproc="..."</code>	Optional: Defines whether a JavaScript-file is to be executed with Javas in-built ScriptEngine instead of committed to the OS.	<code>yes</code> : JavaScript-file is executed with Javas inbuilt ScriptEngine. <code>no</code> : JavaScript-file is committed to the OS. (default)

#### script sub-elements

This sub-element can occur several times.

Sub-element	Description	Reference
<code>&lt;params&gt;...&lt;/params&gt;</code>	Optional: Specifies the transfer parameters for the script to be executed.	See: Structure of the element <code>&lt;params&gt;</code>

### 5.7.35 Structure of Element `<scripts>`

This element encloses the set of all scripts.

```
<scripts
  phase="..."
  >
  <script>...</script>
</scripts>
```

#### scripts attributes

Attribute	Description	Value
<code>phase="..."</code>	Optional: Only for element <code>checkout_rules</code> : Defines in which phase of the transaction the script is executed.	<code>after_transfer</code> : Executes the script after the files have been transferred from SAP to the download directory. <code>after_session_update</code> : Always executes the script (for each document).

#### scripts sub-elements

This sub-element can occur several times.

Sub-element	Description	Reference
<code>&lt;script&gt;...&lt;/script&gt;</code>	Specifies the script to be executed as well as any error conditions to be taken into account. The scripts are called in the order of the <code>&lt;script&gt;</code> definitions.	See: Structure of Element <code>&lt;script&gt;</code>


### 5.7.36 Structure of Element `<seedfile>`

```
<seedfile
  default="..."
  type="..."
  workstation_application="..."
  filename="..."
  docnumber="..."
  doctype="..."
  docpart="..."
  extension_filter="..."
  filter_description="..."
  filename_pattern="..."
  component_reference="..."
  dictionary_identifier="..."
  appl_environment="..."
  appl_release ="..."
  appl_release_ext="..."
  >
```

```
<description ... />
</seedfile>
```

### seedfile attributes

Attribute	Description	Value
default="..."	Optional: This attribute can be used to set a template file to default, which is preset in the <a href="#">Create Document</a> dialog. You may only set one template file.	yes: This template file is set as the default. no: This template file is not set as the default. (default)
type="..."	Optional: Specifies how the original of the document is generated.	template: The filename attribute is used to specify the template file to be used. select: The filename attribute is used to specify the template file to be used. In the <a href="#">Create Document</a> dialog, you can select a template file using a file selection dialog. The filename attribute and the workstation_application, appl_environment, appl_release, appl_release_ext attributes are ignored. master: For dependent documents only: Use original master as seedfile. The filename attribute and the workstation_application attribute are ignored. Note: To enable documents to be created using drag & drop from the file system, a type of type select must be defined at least once for every DType definition. sap: The original document is used as a template. The document to be used as a template is specified using the seedfile attributes docnumber, doctype, and docpart. A document version is not specified because the most recently released version is always used. If a released version is not available, the most recent version is used. If the above attributes used to define a specific document are not specified, the document search can be used to select a document in the SAP system.
workstation_application="..."	Optional: Workstation application in the SAP system for the template file. This depends on the settings in your SAP system.	Identification of the workstation application for example: UGI: NX WRD: Word document TIF: TIFF file ...
filename="..."	Optional: If you use type="template" this attribute defines the file name of the template file that is used. Here, the name can be specified as relative or absolute. If the file name is specified as relative, the template file has to be located in the directory %PLM_INSTDIR%\applications\ <application&gt;\templates. <img="" alt="at symbol icon" data-bbox="762 1989 785 2020" dtypes="" environment="" if="" in="" references="" the="" to="" use="" variables,="" want="" you=""></application&gt;\templates.>	File name of the template file

Attribute	Description	Value
	<p>character instead of . Here, &lt;application&gt; is the application type that has been defined using <code>application_type</code> in the DType attributes. If the template file is missing, no document can be created. In this case, a corresponding error message is displayed.</p> <p>If additional originals are stored, a directory with the same file name can be created in parallel to the template file. The directory contains all additional originals. These additional originals are stored in the document in accordance with the rules specified in the DType.</p>	
<code>docnumber="..."</code>	Optional: If you select a template file from SAP with <code>type="sap"</code> , this attribute defines the document number of the document whose original is used as the template file.	Document number of the template document
<code>doctype="..."</code>	Optional: If you select a template file from SAP with <code>type="sap"</code> , this attribute defines the document type of the document whose original is used as the template file.	Document type of the template document
<code>docpart="..."</code>	Optional: If you select a template file from SAP with <code>type="sap"</code> , this attribute defines the document part of the document whose original is used as the template file.	Document part of the template document
<code>extension_filter="..."</code>	Optional: If you select a template file using <code>type="select"</code> , this attribute defines the filter for the file extension. The file selection dialog offers files that fit the filter. If <code>filename_pattern</code> is output, this attribute is ignored.	File extension, for example: <code>*.prt</code> or several file extensions for example <code>*.jpg, *.gif, *.png</code>
<code>filter_description="..."</code>	Optional: Specifies a description for the used filter when using the attribute <code>filename_pattern</code> . This description is displayed under file type in the file selection dialog. To fetch the description from the dictionary, you can specify a dictionary key for multi-language use. If no suitable entry is found in the dictionary, the text from this attribute is displayed in the file selection dialog. If this attribute is missing or blank, the regular expression from <code>filename_pattern</code> is displayed.	Description of the filter, for example: <code>"IPW-Parts (*ipw*.prt)"</code> or a dictionary key for example <code>"\$(IPW_FILTER)"</code>
<code>filename_pattern="..."</code>	Optional: Enhanced filter option for restricting the selection of a template file from the file system using a regular expression (Regular Expression, RegExp). If <code>filename_pattern</code> is specified, then attribute	Regular expression for example: <code>".*ipw.*\prt"</code> for filtering on all part files ( <code>*.prt</code> ) where 'ipw' occurs in the file name. or <code>^\.*\.(?!pdf dwg doc xls ppt dwg zwl</code>

Attribute	Description	Value
	extension_filter is ignored. The file selection dialog only offers files that fit the filter.	eml docx xlsx pptx) .*\$ for filtering on files with the listed file extensions.
component_reference="..."	Optional: Only for dependent documents: Specifies whether the dependent document contains a component reference (identified in sort term ' :DC ') or whether the document is a simple dependent document (identifier 'D'). When the master is opened, dependent documents with component references are made available in the same way as any available components in the session directory of the user.	yes: A component reference is generated no: Dependent documents without component reference (default)
dictionary_identifier="..."	Optional: If a description is not defined for the seedfile, a search is run in the dictionary for the key plm.dtype.description.<dtype_id> and the description there is used.	Value: <dictionary_identifier> Entry in dictionary: plm.dtype.seedfilename.<dtype_id>
appl_environment="..."	Optional: Automatically specifies the application environment as an attribute when the document is created. The value of this seedfile attribute is written to the "CAD environment" property.	Name of the application, e.g. NX
appl_release="..."	Optional: Automatically specifies the application version as an attribute when the document is created. The value of this seedfile attribute is written to the "CAD version" property.	Application version, e.g. 10.0
appl_release_ext="..."	Optional: Automatically specifies the last release version of the application as an attribute when the document is created. The value of this seedfile attribute is written to the "CAD version" property.	Last release of the application, e.g. 10.0.1.3

### seedfile sub-elements

This sub-element can occur several times.

Sub-element	Description	Reference
<description ... />	Meaningful description for a template file. The description is displayed according to the logon language in SAP ECTR for SAP S/4HANA when new documents are created. The description can be specified in more than one language. To achieve this, element Seedfile_description is inserted more than once and the value for the language attribute and the description are changed. If this element is defined, dictionary_identifier is ignored.	See: Structure of element <description>

### 5.7.37 Structure of Element <status\_transition>

```
<status_transition>
  < status_transition_restriction... />
</>
```

#### status\_transition sub-elements

This sub-element may be repeated.

Sub-element	Description	Reference
<code>&lt;status_transition_restriction ... /&gt;</code>	Specifies the rules for filtering out the post status.	See: Structure of Element <status_transition_restriction>

### 5.7.38 Structure of Element <status\_transition\_restriction>

```
< status_transition_restriction...
  type="..."
  from="..."
  to="..."
/>
```

#### status\_transition attributes

Attribute	Description	Value
<code>type="..."</code>	Specifies whether the post status is allowed or not	whitelist: Post status is allowed blacklist: Post status is not allowed
<code>from="..."</code>	Specifies the status that the rule relates to.	The status must be entered.
<code>to="..."</code>	Specifies the allowed or forbidden post status.	The post status must be entered.

#### Example

```
<program_options>
<status_transition>
  <status_transition_restriction type="blacklist" from="SW" to="IA"/>
  <status_transition_restriction type="whitelist" from="IW" to="AC;IA;FR"/>
  <status_transition_restriction type="whitelist" from="AC" to="IW;PR"/>
</status_transition>
</program_options>
```

```
// The following post status list is defined in the SAP system:
// SW → Post status: [AC, IW, IA]
// IW → Post status: [AC, IA]
// AC → Post status: [AP, ER, EX, FR, IW, PR, IA]
```

```
// The <status_transition_restriction> rules from the example above
// produce the following result:
// SW → allowed [AC, IW], IA on blacklist
```

```
// IW → allowed [AC, IA], all are on whitelist
// AC → allowed [IW, PR], only these are on whitelist
```

### 5.7.39 Structure of Element <store\_files>

This element defines to which workstation application an additional file is assigned when it is added using drag & drop or a menu function.

If element <store\_files> is not defined, no additional originals can be added using drag & drop or a menu function.

```
<store_files>
  <plmfile.../>
</store_files>
```

#### store\_files sub-elements

This sub-element can occur several times.

Sub-element	Description	Reference
<plmfile ... />	Specifies the assignment between the file and the workstation application. Attributes extension, filename_pattern and workstation_application are relevant here	See: Structure of Element <plmfile>



#### Example

```
<additional_files enable_additional_files="yes" >
...
  <store_files>
    <plmfile extension="txt" workstation_application="TXT"/>
    <plmfile extension="pdf" workstation_application="PDF"/>
    <plmfile filename_pattern="*" workstation_application="ANY"/>
  </store_files>
...
</additional_files>
```

### 5.7.40 Structure of Element <view\_files>

This defines for which of the additional originals displayed in the [Originals](#) container no processing may be performed in the document. These originals can only be displayed.

```
<view_files>
  <plmfile.../>
</view_files>
```

#### view\_files sub-elements

This sub-element can occur several times.

Sub-element	Description	Reference
<plmfile ... />	Specifies which of the displayed additional originals cannot be edited.	See: Structure of Element <plmfile>

### 5.7.41 Using Templates for DType Definitions

When you define DTypes, you often need entire groups of document definitions, which match in regard to their settings and only differ in a few attributes. To avoid maintaining all the settings in the DType definition file several times, you can define DType templates with common settings. You can use this definition several times in the DType definitions for the documents.

#### Defining a DType Template

The definition of a DType template corresponds to that of a master document or a dependent document, with the only difference being that the type of DType is set to `template`. You can use the Attribute ID to assign a unique ID for each DType template. You can use the ID to reference the DType template in the DType definition of a document.

Except for XML element <description> a DType template can contain all the settings of a DType.

```
<DType type="template" identifier="UGMDL" application_type="UGS" >
  <icon> <icondef icon_position="1"
    icon_group="default"
    icon_identifier="tree/ugs/
    $(TYPE) $(SUBTYPE) $(FAMILY) "/>

    <icondef icon_position="2"
    icon_group="default"
    icon_identifier="flag/flag_version
    _$(HER) $(HNV) $(HRV) "/>

    <icondef icon_position="3"
    icon_group="default"
    icon_identifier="flag/flag_primaryfiles
    moe$(MOE)_moc$(MOC)_iu$(IN_USE_BY_OTHER) "/>

    <icondef icon_position="4"
    icon_group="default"
    icon_identifier="flag/flag_addfiles_aoe$(
    (AOE)_aoc$(AOC)_iu$(IN_USE_BY_OTHER) "/>

    <icondef icon_position="5"
    icon_group="default"
    icon_identifier="flag/flag_material
    _$(MATX) $(MSTA) "/>

    <icondef icon_position="6"
    icon_group="default"
    icon_identifier="flag/flag_$(DRAW) "/>

  </icon>

  <program_options show_for_create="yes"
    has_primary_application="yes"
    dirty_flag="yes"
    attempt_status_change_on_checkin="yes"
    function_group="STM"
    customer_data_section=""
    document_data_class="DOC_ADDIT_FIELDS"

  <new_version status_list="AC;FR"
    ignore_wsappl_list="ZIP"
    ignore_nmDType_list="" />
```

```

<copy_as
    allowed="yes"
    ignore_wsappl_list="ZIP"
    use_initial_version="yes"
    copy_material="no" />
</program_options>

<document_create
    create_with_SAP_GUI="no"
    create_multi_allowed="no"
    document_number_assignment="external"
    initial_document_number="*"
    document_type=NXM"
    initial_document_part="000"
    part_editable="no"
    initial_document_version="00"
    version_editable="yes"
    change_to_after_create_status=""
    filename_template_base="$(DOCNUMBER)
$(DOCTYPE)$(DOCPART)_"
    filename_template_version="$(DOCVERSION)"/>

<allowed_nonmaster DType="DRWV" max_count="3"
    master_status_list="AC;IA"/>
<allowed_nonmaster DType="NOMA" max_count="3"/>
<allowed_nonmaster DType="SPEC" max_count="3"/>

<material_create
    material_creation_method="deferred"
    material_number_assignment="master"
    material_template=""
    material_plant_assignment="0001"
    material_default_class="" />

<primary_application
    primary_workstation_application=NXI"
    attribute_section="STDMOD">

<container_file>
    <plmfile extension="ecf" workstation_application="ECF"
        description="ECTR container file"/>
</container_file>

<checkout_rules
    rename_master_original_at_checkout="no"
    one_version_in_session_only="yes"
    hardlink_in_session="yes"/>

<checkin_rules>
    <save_direct>
        <plmfile extension="jpg" workstation_application=
            "EPJ" description="ECTR Preview (jpeg)"/>
        <plmfile extension="gif" workstation_application="EPG"
            description="ECTR Preview (gif)"/>
        <plmfile extension="jt" workstation_application="JT"
            description="ECTR 3D Preview"/>
    </save_direct>

    <ignore>
        <plmfile extension="bak"/>
        <plmfile extension="tmp"/>
        <plmfile filename_pattern="plm_metainf-sap.*"/>
    </ignore>

```

```

<delete_in_sap>
  <plmfile workstation_application="EPG"/>
  <plmfile workstation_application="EPJ"/>
  <plmfile workstation_application="JT"/>
</delete_in_sap>

<delete_in_session>
  <plmfile extension="jpg"/>
  <plmfile extension="gif"/>
  <plmfile extension="jt"/>
</delete_in_session>
</checkin_rules>
</primary_application>

<additional_files      enable_additional_files="yes">
  <view_files>
    <plmfile workstation_application="XLS"/>
  </view_files>

  <ignore_files>
    <plmfile workstation_application="EPJ"/>
    <plmfile workstation_application="EPG"/>
    <plmfile workstation_application="JT"/>
  </ignore_files>
</additional_files>
</DType>

```

## Using a DType Template

To use a DType template in a DType definition for a master document or a dependent document, the template is referenced using its unique ID via attribute `include_template`. Hence, the DType definition of the document only contains the XML elements `<description>` and the settings that differ from the template. The settings in the DType definition of the document overwrite the settings of the DType template.

Using a DType template significantly reduces the time required to define the DType of a document.

```

<DType type="master" identifier="PINT" include_template="UGMDL" application_type="UGS">
  <description language="DE" text="NX Part mit generierter Nummer"/>
  <description language="EN" text="NX part with generated number"/>

  <document_create document_number_assignment="internal"
    initial_document_number="*"
    version_editable="no">

    <seedfile filename="model-mm.prt" type="template" default="yes">
      <description language="DE" text="Standard Startmodell" />
      <description language="EN" text="standard base model" />
    </seedfile>

    <seedfile filename="model-mm.prt" type="select" default="no"
      extension_filter="*.prt">
      <description language="DE" text="vorhandenen Part auswaehlen"/>
      <description language="EN" text="select existing partfile"/>
    </seedfile>
  </document_create>
</DType>

```

### 5.7.42 Access to Environment and Preference Variables

Within the DType definition, you can optionally use variables that access the value of environment variables or preference variables to assign XML attributes. The notation of the variables is as follows:

@<Variable\_name>@

The <Variable name> represents the name of the desired environment variables or SAP ECTR for SAP S/4HANA preference variables.

Variable @PLM\_ROLE@ accesses the content of environment variable %PLM\_ROLE%, which SAP ECTR for SAP S/4HANA sets during SAP Logon.

```
<program_options
  ...
  function_group="@PLM_ROLE@"
  ...
/>
```

### 5.7.43 Rules for Creating New Documents

The following sections describe the rules for the creation of new documents in more detail.

#### Creating a Master Document from an Application

When you create a new document from the application, the type of the application is evaluated (for example UGS). This application type is used to determine the valid master DType elements that are displayed for selection in the document creation dialog.

All definitions where the value of attribute `application_type` in XML element <DType> matches the application type determined for the application are valid.

Example for application type UGS:

```
<DType type="master" identifier="PEXT" include_template="UGMDL" application_type="UGS">
```

#### Creating a Dependent Document from an Application

To do this, the master document must be open in the application.

For the master document, the defined allowed DTypes for dependent documents are determined from its DType definition in XML element <allowed\_nonmaster>. The allowed DTypes are displayed for selection in the document creation dialog.

```
<allowed_nonmaster DType="DRWV" max_count="3"
  master_status_list="AC;IA"/>
<allowed_nonmaster DType="NOMA" max_count="3"/>
<allowed_nonmaster DType="SIMPL" max_count="1"/>
<allowed_nonmaster DType="SPEC" max_count="3"/>
<allowed_nonmaster DType="SEEN" max_count="3"/>
```

#### Creating a Master Document from SAP ECTR for SAP S/4HANA

From SAP ECTR for SAP S/4HANA, you can create all master documents for which the value for attribute `show_for_create` was not set to "no" in XML element <program\_options> in the DType definition. If the value of the attribute is set to "no" the DType does not appear in the selection list of DTypes in the document creation dialog of SAP ECTR for SAP S/4HANA.

```
<program_options show_for_create="no"
  has_primary_application="no"
  dirty_flag="no"
  attempt_status_change_on_checkin="no"
  function_group=""
  customer_data_section=""
  document_data_class="DOC_ADDIT_FIELDS"/>
```

## Creating a Dependent Document from SAP ECTR for SAP S/4HANA

To be able to create a dependent document, the corresponding master document must be selected. A new dependent document can only be created with a master document.

For the master document, the defined allowed DTypes for dependent documents are determined from its DType definition in XML element `<allowed_nonmaster>`. The allowed DTypes are displayed for selection in the document creation dialog.

### Hinweis zur Anlage von Ordnern mit eigenem DType

Wenn Sie einen Ordner mit eigenem DType erstellen möchten, achten Sie darauf in der `default.txt` die folgende Preference-Variable zu setzen:

```
PLM.CONTROL.EMPTY.DTYPE.<document_type> = <value for dtype>
```

Hier definieren Sie den Dokumenttyp Ihres Ordners, damit die Zuordnung zum gewählten DType korrekt verläuft.

## 5.8 Automated Configuration Check

SAP ECTR for SAP S/4HANA checks whether the configuration is correct, on start-up or when a configuration file is loaded using the administrator menu.

### Check on Start-Up

When an administrator launches SAP ECTR for SAP S/4HANA, a dialog is displayed. The dialog lists the incorrectly configured DTypes. If the administrator cancels or closes the dialog, the SAP ECTR for SAP S/4HANA application also closes. However, SAP ECTR for SAP S/4HANA can be run despite the error message.

### Check During Upload

Even if errors are discovered when a configuration is uploaded using the administrator menu, the administrator can still execute the upload. The administrator can view the errors or can close the process, thereby terminating the upload.

The automated check is controlled in `default.txt` using the following preference variables:

Preference variable	Description	Value
plm.checker.profile. [profile]	Specifies the check that is to be performed. Both checks can be activated. For [profile], the following are possible: configUpload - Check takes place during upload postlogon - Check takes place at start-up	plm.proc.check.DTypeAudit Checks whether the workstation application, status, document types, and template files specified in a DType exist.
		plm.utils.configtransmitter.ConfigSapAudit Checks which add-ons and applications are included in the installation directory but not included in the configuration directory.
		plm.proc.check.MatEditClassExistsInSap

Preference variable	Description	Value
		Checks whether the MatEditClasses in the preference variables <code>plm.mat.edit.XXX</code> also exist in the SAP system
		<code>plm.proc.check.ApplicationsPreferenceAudit</code> Checks whether application-specific preference variables are in the correct configuration files.

## 5.9 Editability of Class Types and Characteristics

The editability of classes and characteristics in the Object Browser can be restricted using the following preference variables. This restriction does not refer to editability in the SAP system.

```
plm.classification.readonly.pattern =
<classtype1>/<classname1>/<characteristic1>;<classtype2>/<classname2>/<characteristic2>;
```

### Example

```
plm.classification.readonly.pattern = 001/./.*
```

### 6.1 Purpose of Use/Area of Application

SAP ECTR for SAP S/4HANA macros use the Oracle Nashorn JavaScript Engine. This allows potentially complicated tasks to be automated with simple scripts. Besides the control structures and standard functions provided by JavaScript, additional SAP ECTR for SAP S/4HANA -specific functions are provided in the editor.

Macros are saved in the SAP system as a document with an attached source code text file. In this way, all SAP ECTR for SAP S/4HANA functions for maintaining and distributing macros are available to the user.

Alternatively, you can save a macro as a \*.txt macro file in the following SAP ECTR for SAP S/4HANA directories: ... \customize \scripts \macros.

```
... \applications \<Application> \customize \scripts \macros
```

This allows you to define certain macros as part of the controlled SAP ECTR for SAP S/4HANA installation (e.g. in a SmartContainer).

SAP ECTR for SAP S/4HANA macros can be used in three different ways:

- Directly
- In SmartContainers
- In SmartLists or SmartFolders

An SAP ECTR for SAP S/4HANA macro can be run directly. For example, the user can use macros to:

- Find and access selected objects
- Transform, filter
- Output resulting objects in object lists

SmartLists and SmartFolders are essentially object lists with assigned macros. Since they also save input parameters, frequently used queries are made available with a single mouse click. With the Auto Update Function, a SmartList or SmartFolder can be executed automatically with a defined frequency.

With SmartContainers, you can define customer-specific containers for SAP objects. Macros are used to receive the SAP object as reference. The resulting objects are then displayed in the container.

### 6.2 Using Macros in SAP ECTR for SAP S/4HANA

Depending on the desired function, macros can be stored in either the SAP system or the installation directory.

To enable the configuration to access macros, you store those intended for smartcontainers in the macro directory. Macros for SmartLists can be stored in the SAP system, as well. The appropriate storage method depends on whether a given macro is to be available across multiple databases.

#### 6.2.1 Creating Macros

To create a macro in the SAP system using the editor, you create a new macro script in the [Create Document](#) dialog from the DType group [MACRO](#) with DType [Macro Script](#).

The macro script is edited in the SAP system using a .txt file, which is located in the [Originals](#) container.

In the desktop window, the required functions for macro scripts are available in the menu bar and in the context menu.

## 6.2.2 Functions

Function	Description
Edit Macro	Opens the macro script for editing
View Macro	Opens the macro script with read permission only
Run Macro	Runs the current macro script
Document Operation	Opens a flyout menu. The user has the option to check the macro in and out, undo changes, and delete the macro.
New Search with SmartList	Creates a SmartList for the macro
Copy Document	Copies the macro and allows the processor to...
Other Options...	Opens a flyout menu that enables the user to perform a number of operations in SAP GUI and send the macro by SAP e-mail
PLM WEB UI	Opens a flyout menu that gives the user access to a number of operations in PLM WEB UI
Object List Operations...	Opens a list of object list operations that can be executed using the macro
Copy	Copies the macro
Cut	Cuts the macro
Remove	Removes the macro
Update Metadata	Updates the macro's metadata

### 6.2.2.1 Macro Editor

#### Note

Hashtags may not be used to comment in the macro source code. To comment out a single line, it must be started with `//`. If a line is split, the commenting from the previous line is terminated. Ranges that extend over multiple lines can be started with `/*` and commented out with `*/`.

Right-click the macro you have created to bring up the context menu. Choose [Edit Macro](#). The macro editor now opens.

In the macro editor, you can create and edit a macro script. The script contains a menu bar and the input field for the macro.

The menu bar contains the following functions:

- Save (to save the content in the SAP system)
  - After finishing the macro script, choose [Save](#) and check the macro into the SAP system.
- Save as.. (to save the content in the installation folder)
  - Save the macro in the installation folder.

#### Hinweis

Ein Makro, das im Installationsordner gespeichert wurde, ist nicht automatisch im SAP-System eingecheckt. Achten Sie darauf, Makros noch einmal separat im SAP-System zu speichern.

## Note

If macros contain unsaved changes, the macro name in the title bar of the macro editor contains an asterisk (\*). When you close the macro editor without previously saving your changes, a window appears that offers you the following options:

- o Save without checking-in
- o Save and check-in
- o Ignore changes

- Run Macro

- After creating a macro in SAP ECTR for SAP S/4HANA, run it by clicking [Run Macro](#) in this editor or in the [Desktop](#) window.

The syntax is based on JavaScript. When macros are created, the syntax elements are highlighted in various colors. If some attributes and commands do not appear in color, this may be due to the following reasons:

- Input error
- Native JavaScript code

The macro code created in the editor is context-sensitive.

The macro editor can also provide input assistance. Pressing **CTRL+Space** displays a list of all the possible macro operations. Please note that only the functions developed by DSC Software AG are listed here; this does not include the native JavaScript functions.

If you hold down **CTRL** and double-click a function, the system displays related documentation.

For an overview of all the functions that you can use in macros in addition to those defined by JavaScript, please see the [Available Functions](#) section.

When functions are executed, various errors can occur. The macro editor displays information about errors in the status bar or relevant log file.

### 6.2.2.2 Displaying Macros from the Configuration Folder

With the macro menu in the menu bar, you can display all macros ending in `.txt` or `.macro` in the installation directory.

The macros are displayed in an overview window.

If you right-click on a macro, a menu appears:

- Run
  - Opens the [Run Macro](#) dialog. If parameters are defined within a macro, these parameters are displayed here. If documentation exists for a macro, this documentation is displayed here. You create macro documentation within the macro using the tags `<DOK></DOK>`. Start the macro in this window by using the command [Run](#).
- Edit
  - Opens the macro editor. You can edit the macro script.
- Rename
  - Renames the macro. The relevant ending is retained.
- Delete
  - Deletes the macro.

## 6.3 Definitions of Basic Technology

Macros function with the help of objects that need to be defined prior to processing.

### 6.3.1 KeyList

KeyLists are lists of object keys in the front end. They are used for the input and output of many macro functions in the front end.

### 6.3.2 ObjectSet

KeyLists can be converted into ObjectSets. ObjectSets are references to an object set in the back end. They are used for the input and output of macro function modules in the SAP system.

Macro functions with ObjectSets can either be buffered or run directly. In buffered mode, the functions are executed only if the results are required in the front end. This reduces API calls and increases performance. Buffered mode is used by default.

For this reason, you should always convert ObjectSets into KeyLists before the output.

### 6.3.3 Parameter Maps

Parameter maps are used to describe the properties of an object. The resulting list of properties enables you to search for the relevant object within a set of objects.

When you create a parameter map, the editor assists you by displaying the entries you can make. To enable this feature, however, a parameter map first needs to be defined, for example:

```
p = PARAMETER_MAP("mat_fields")
```

p.

Once you enter "p.", the system displays a list of possible entries.

This functionality also applies for parameter maps that contain classification data. In this way, characteristics of a class can also be displayed.

## 6.4 Creating SmartLists

SmartLists are macro-programmed search functions you can run repeatedly or set to run automatically.

Unlike object lists, SmartLists remain in the desktop window beyond the SAP ECTR for SAP S/4HANA session that is currently running. They are neither saved nor managed in the SAP system.

Macros for SmartLists can be stored in both the SAP system and the file system.

### Note

Macros that are only present in the SAP system cannot be used across multiple databases. Please bear this in mind when using multiple databases.

You can link macros to SmartLists in two ways, which will be introduced in the sections below.

### 6.4.1 New Search

#### Note

When creating macros, please note that they must be UTF-8 encoded.

When macros are used with the Smart Wizard, a corresponding macro must first be defined in the file system. This serves, among other things, to give users direct access to macros without all users being given full access to the underlying macro functions.

Macros that can be used with the Smart Wizard are stored under  
%PLM\_INSTDIR%\customize\scripts\macros\wizard.

They can be used across multiple databases.

To create a SmartList using the Smart Wizard, proceed as follows:

- In the desktop window, right-click [SmartLists](#).
- In the menu: choose [New Search](#)
- The [SmartList Wizard](#) dialog appears.
- In the "File Name" field, choose the appropriate macro. Your selection determines what happens next.
- Enter a meaningful name for the SmartList you want to create in the [SmartList Name](#) field.
- Click [OK](#).

- The macro is run.

### **i Note**

All of the macros included in the standard contain references to search screens designed to help the user find materials, documents, and change numbers. The command `GET_MASK_INPUT` (see the [User Input](#) section) can be used to create custom input dialogs, as well. Meanwhile, the Smart Wizard can be used to call up every type of macro, not just those that provide search functions.

#### **6.4.2 New Query with SmartList**

- SmartLists created using [New Search](#) can be adjusted in a flexible manner. Such SmartLists are more adaptable than those created using the [Smart Wizard](#). The [New Search](#) function also enables you to define an advanced search query. A macro can be loaded either directly from the SAP system or from the file system. Macros that can be loaded from the file system are stored here:

- `%PLM_INSTDIR%\customize\scripts\macros\`
- You can also set further relevant parameters, such as a time interval for automated updates.

To create a SmartList using the [New Search](#) function, proceed as follows:

- In the desktop window, right-click [SmartLists](#).
- In the menu that appears, choose [New Search](#).
- The [Configure SmartList](#) dialog appears.
- Enter a meaningful name for the SmartList you want to create in the [SmartList Name](#) field. Unlike searches, SmartLists are visible to multiple users and giving a suitable name makes it easier for everyone.
- Click the binoculars icon.
- The [Select Macro](#) dialog appears. Macros stored in the SAP system can be found on the [Documents](#) tab. Macros stored in the file system can be found on the [Files](#) tab. The search field in the upper-right area of the dialog enables you to filter down to the macro you need.
- Click the macro you want to use.
- Confirm your selection.
- This takes you back to the [Configure SmartList](#) dialog.
- You can use the list button next to the binoculars icon to send the selected macro to the active list.
- Other settings for the macro can be made using the button with the alt text [Settings for auto refresh and others](#) on the right under the documentation field
  - When you define automatic update of the SmartList, you define the corresponding number of seconds for the update in the [Autoupdate](#) field.
  - You can use [Ignore Tree Sorter](#) to ignore the sorting defined for the objects for the desktop.
- [Ignore Tree Sorter](#) enables you to ignore the sorting method defined for the objects in the desktop window.
- If a macro returns certain parameters to SAP ECTR for SAP S/4HANA, the [Parameters](#) field displays the variable in question along with its value.
- If a macro provides corresponding information, this is displayed in the [Documentation](#) field.
- Click [OK](#).
- The SmartList corresponding to the macro now appears in the desktop window.
- To activate the attached macro, expand the SmartList you have just created.

#### **Making follow-up changes to SmartLists**

- You can edit SmartLists after creating them. To do so, right-click the SmartList in question to bring up the menu. If you created the SmartList with the wizard, you can use the following functions:
- [Edit Search Parameters](#). This function is only available for existing search dialogs.
- [Remove macro](#)

If you created the SmartList using [New Search](#), you can use the following functions:

- [Update Metadata](#). This function refreshes the values of all the results displayed in the SmartList.
- [Edit](#) the SmartList
- [Remove](#) the SmartList
- [Display Macro](#)

### 6.4.3 SmartFolder

The functionality of SmartFolders is similar to that of SmartLists, with the difference that SmartFolders can be integrated into the folder structure of SAP ECTR for SAP S/4HANA.

#### Creating and Editing SmartFolders

- Right click on a folder and select [New](#) → [SmartFolder](#).
- The [Configure SmartList](#) dialog opens.
- Name the SmartFolder.
- Click on the binoculars.
- The [Select Macro](#) dialog appears. You can find macros that are stored in the SAP system on the [Documents](#) tab. You can find macros that are stored in the file system on the [Files](#) tab. The search field in the upper right area of the dialog allows you to filter for a macro you are searching for.
- Click on the macro you want to use.
- Confirm.
- You are now back in the [Configure SmartList](#) dialog.
- If a macro returns certain parameters to SAP ECTR for SAP S/4HANA, the variable and its value are displayed in the [Parameters](#) field.
- If the macro provides corresponding information, you can read this information in the [Documentation](#) field. You create macro documentation within the macro using the tags `<DOK></DOK>`.
- You can use the List button next to the binoculars to send the selected macro to the active list, if required.
- Click on [OK](#).
- The SmartFolder with the corresponding macro is displayed in the desktop.

If you want to edit a SmartFolder that has been created, right click on it and select "Edit SmartFolder".

#### Configuring SmartFolders

In `menu.guidedef`, the menu must be adjusted so that it enables a finished SmartFolder to be edited. For this purpose, you create the following entry:

```
om.popup.menu.SMART_FOLDER      = fnc.edit.smart.folder
                                = -----
                                = ?  CLPBRD_OBJ
```

In `menu_macros.txt`, you specify that SmartFolders are displayed in the menu for creating new documents:

```
? FOLDER_NEW
...
= fnc.create.smart.folder
```

### 6.5 Creating Smartcontainers

Smartcontainers enable you to run macros. Macros start from the object to which the relevant smartcontainer is attached or the containers that include said object.

Macros used for smartcontainers can only be stored in the file system. These macros are stored in a single file and can be modified using your preferred editor. This means that these macros can be used across multiple databases in any existing SAP system.



### Example:

The example below involves the creation of a smartcontainer for assemblies. The smartcontainer displays all of the released documents pertaining to the assembly at hand.

```
//Step one. Here, a keylist is created starting from the object under which the smartcontainer in question is located.
```

```
k = KEYLIST_FROM_REFOBJ()
```

```
//Step two. Here, the object set o is created based on keylist k. This is necessary to facilitate the execution of complex operations on the set.
```

```
o = CREATE_SET( k )
```

```
//Step three. This step returns released documents from the object set. The result is also transferred to another object set, t.
```

```
t = TRANSFORM_SET( s, "doc_to_released" )
```

```
//Step four. Object set t is converted back into a keyset for output.
```

```
result = KEYLIST_FROM_SET( t )
```

```
//Step five. The keylist is written as output under the smartcontainer in question.
```

```
WRITE_RESULTLIST( result )
```

Store the macro under the following path:

```
%PLM_INST%\applications\ugs\customize\scripts\macros
```

The folder chosen in the `applications` directory determines the object type to which the macro is attached.

## 6.5.1 Front-End Adjustments

### Changes in `%PLM_INST%\customize\config\default.txt`

To implement a smartcontainer, start by opening the file `%PLM_INST%\customize\config\default.txt`.

Then navigate to the [Smartcontainer Definitions](#) section.

Copy the following lines into the smartcontainer definitions:

```
plm.smart.container.dic.<ContainerName> = <DICTIONARY_ENTRY>
```

Here, you can specify the dictionary key that contains the name of the smartcontainer. The language version is taken into account.

```
plm.smart.container.<ContainerName>.name = <FALLBACK_NAME>
```

If no name exists in the dictionary, this name is used for the container.

```
plm.smart.container.<ContainerName>.functionGroup = <FUNCTION_GROUP>
```

Here, enter the function group to be called up later by `menu.guidef`. You can choose the name of the function group yourself.

```
plm.smart.container.<ContainerName>.macro = <STORAGE_LOCATION>
```

Here, specify the storage location of the macro under the following path:

```
%PLM_INST%\applications\\scripts\macros
```

If you only enter the file name, the program searches through the following directory:

```
%PLM_INST%\customize\scripts\macros
```

```
plm.smart.container.<ContainerName>.icon = <ICON_FOR_SMARTCONTAINER>
```

Here, you can select a symbol for the smartcontainer, whereby the program accesses the compressed images.jar file.

When the macro is used for assemblies, the filled-out preference variables appear as follows:

```
plm.smart.container.dic.MACRO = example
plm.smart.container.MACRO.name = released documents
plm.smart.container.MACRO.functionGroup = EXAMPLE_MACRO
plm.smart.container.MACRO.macro = UGS:BEISPIELMACRO.txt
plm.smart.container.MACRO.icon = {0}/sap/ablage3
```

If the smartcontainers are not displayed on all of the objects in the group you selected, you set the following parameters:

- `plm.smart.container.MACRO.includeFilter = fam_member_filter`
- `plm.smart.container.MACRO.excludeFilter = fam_member_filter`

This function sets a filter. With `includeFilter`, only the objects selected by the filter are added. In the case of `excludeFilter`, the same objects are **not** added.

If you set both of these parameters, `includeFilter` is applied before `excludeFilter`.



#### Example:

The filter accessed by the preference variable is defined in `default.txt`, for example:

```
obr.filter.fam_member_filter = DOK_FILTER
```

By default, entries in the SmartContainer are sorted based on the settings. Using

```
plm.smart.container.BEISPIELMACRO.ignore.tree.sort = true,
```

entries can be displayed depending on the time they were added (`true`) or alphabetically (`false`).

To display dialogs and masks every time a macro is executed, the following parameter can be set:

```
plm.smart.container.BEISPIELMACRO.always.display.searchmask = true
```

You can open an editable search mask with the following parameter:

```
plm.smart.container.BEISPIELMACRO.always.display.searchmask.for.edit = true
```

SmartContainers can be assigned to function groups, which then define separate toolbars and context menus. Every object to which this function group is assigned then displays exactly these toolbars and context menus.

```
plm.smart.container.BEISPIELMACRO.custom.function.group = "GROUP"
```

#### **Changes in %PLM\_INST%\applications\\customize\config\default.txt**

The definition of the preference variable depends on the object to which the smartcontainer is being added.



### Example:

When a macro is added to assemblies, the `default.txt` must be modified in the following folder:

```
%PLM_INST%\applications\ugs\customize\config\
```

If the smartcontainer is added to UGM-type documents, the macro is appended to the corresponding container definition:

```
plm.om.DOC.containers.UGM = SMART_CONT(EXAMPLE_MACRO)
```

### Changes in `%PLM_INST%\customize\config\menu.guidef`

In `menu.guidef`, you can define corresponding functions that then appear in the smartcontainer's context menu.

- Navigate to [Container Operations Menus](#).

The function group matches the one you created in `default.txt`.

- Add the following definition to `menu.guidef`:

```
om.popup.menu.<FUNCTION_GROUP>= <desired function>
```

- Function descriptions can be found in the dictionary.
- Restart SAP ECTR for SAP S/4HANA.

The new smartcontainer has now been implemented.

## 6.5.2 Smartcontainers and ABAP modules

When using ABAP modules in macros, you need to preconfigure a corresponding function module in the back end. This function module then serves as the interface to the macro in question.



### Example:

```
*"-----
*"Local Interface:
*" EXPORTING
*" VALUE(VERSION_ID) TYPE /DSCSAG/VERSION_ID
*" VALUE(RETURN) TYPE BAPIRET2
*" VALUE(RUNTIME) TYPE /DSCSAG/RUNTIME
*" TABLES
*" IT_PARAMETERS STRUCTURE /DSCSAG/SET_NAME_VALUE OPTIONAL
*" IT_CONTAINER_OBJECTS STRUCTURE /DSCSAG/OBJECT OPTIONAL
*" ET_CONTAINER_OBJECTS STRUCTURE /DSCSAG/OBJECT OPTIONAL
*"-----
```

You can define the corresponding module in the following file:

```
%PLM_INST%\customize\config\default.txt
```

Use the following command:

```
plm.smart.container.< ContainerName >.params.0 = FM:<NAME OF FUNCTION MODULE IN BACK END>.
```

You can copy established variables to the module using the following definition:

```
plm.smart.container.< ContainerName >.params.<x>
```

You can call function modules in a macro using the input `PARAMS[0]`, for example:

```
setId = TRANSFORM_SET(setId, PARAMS[0])
```

In `ET_CONTAINER_`, all object types that are known in SAP ECTR for SAP S/4HANA as first level objects are processed. If certain object types are not known, these object types are ignored. In `ET_CONTAINER_OBJECTS`, the `OBJECT_KEY` and `OBJECT_TYPE` properties must always be transferred.

**i Note**

A sample function module for a smartcontainer is included in SAP ECTR for SAP S/4HANA:

`/DSCSAG/ECM_WHERE_USED`

### 6.5.3 Nesting Smartcontainers into Each Other

Smartcontainers can be nested into each other. This change is made in the `default.txt` file.

This function is explained using the following example. In this example, two containers, which themselves contain additional smartcontainers, are subordinated to the object `DOK`.

```
plm.om.DOC.containers = GROUP_CONT(GROUP1);GROUP_CONT(GROUP2)
plm.group.container.GROUP1.name = Group 1
plm.group.container.GROUP1.children= SMART_CONT(GEN_STR);
SMART_CONT(STR);SMART_CONT(ALL_VERSIONS);
plm.group.container.GROUP1.icon = {0}/sap/folder_closed_white

plm.group.container.GROUP2.name = Group 2
plm.group.container.GROUP2.children=
SMART_CONT(ROOT);FAMILY_TEMPLATE;MASTERMODEL;OLINKS;SMART_CONT(ASGND_ECN);SMART_CONT(ECN_
OVERVIEW);FAM_MEMBER;VERSIONS;
plm.group.container.GROUP2.icon = {0}/sap/folder_closed_orange
```

**Step 1:**

```
plm.om.DOC.containers = GROUP_CONT(GROUP1);GROUP_CONT(GROUP2)
```

The container groups mentioned above that are marked with `GROUP_CONT` are attached to objects of type `DOC`.

**Step 2:**

```
plm.group.container.GROUP1.name = Group 1
```

First, the container group is named.

**Step 3:**

```
plm.group.container.GROUP1.children= SMART_CONT(GEN_STR);
SMART_CONT(STR);SMART_CONT(ALL_VERSIONS);
```

Then, all required smartcontainers are subordinated to the group. The containers are defined in more detail in `menu.guidedef`.

To find out how containers are created, see above. This example assumes that the named smartcontainers have been configured correctly.

**Step 4:**

```
plm.group.container.GROUP1.icon = {0}/sap/folder_closed_white
```

An icon is added to the group.

For `Group2` Step 2, Step 3 and Step 4 are repeated.



Function	Purpose	Syntax
	S/4HANA object list.	
KEYLIST_FROM_ACTIVITYLOG	Creates a <b>KeyList</b> for objects that were selected from the SAP ECTR for SAP S/4HANA activity logs (last opened, last edited, last saved...)	<pre>&lt;keylist&gt; = KEYLIST_FROM_ACTIVITYLOG(&lt;parameters&gt;)</pre> <p>See <code>activity_log_fields</code> for parameters</p>
WRITE_OBJECTLIST	Creates / overwrites a designated SAP ECTR for SAP S/4HANA object list with the objects in a <b>KeyList</b>	<pre>WRITE_OBJECTLIST( &lt;keylist&gt;, &lt;objectListName&gt;)</pre>
PLM_OBJECT_KEYS_FROM_KEYLIST	Creates <b>PlmObjectKeys</b> required for OSGi from a <b>KeyList</b> .	<pre>&lt;PlmObjectKeys-Name&gt; = PLM_OBJECT_KEYS_FROM_KEYLIST(&lt;keylist&gt;)</pre>

### Example:

```
//Stores all materials selected from the active context
//in an object listed sorted by material number
keylist = KEYLIST_FROM_CONTEXT("active","selected","MAT")
if(keylist.length > 0)
{
keylist.sort()
WRITE_OBJECTLIST(keylist,"Selected Materials")
}
else REPORT("No material selected")
```

## 6.7.2 Set Operations

Function	Purpose	Syntax
CREATE_SET	Creates (in the SAP system) a set that contains the objects in a keylist.	<pre>&lt;setid&gt; = CREATE_SET(&lt;keylist&gt;)</pre>
OPERATION_SET	Executes a set operation on two object sets.	<pre>&lt;setId&gt; = OPERATION_SET(&lt;operation&gt;, &lt;setId1&gt;, &lt;setId2&gt; )</pre>
		<pre>&lt;operation&gt; Indicates the operation to be executed:</pre>

Function	Purpose	Syntax
		<ul style="list-style-type: none"> <li>• "union" Set union</li> <li>• "intersection" Intersecting set</li> <li>• "complement" Complement (in mathematical terms)</li> <li>• "symmdiff" Difference</li> </ul>
TRANSFORM_SET	Transforms SAP objects by applying certain rules.	<pre>&lt;setId&gt; = TRANSFORM_SET( &lt;setId&gt;, &lt;rule&gt; )</pre> <ul style="list-style-type: none"> <li>• &lt;rule&gt; Specifies the way in which SAP objects are converted:</li> <li>• "olinks" Imports the object links of the relevant object</li> <li>• "doc_to_latest" Returns the newest version</li> <li>• "doc_to_released" Returns the released documents</li> <li>• "doc_to_mat" Returns the linked materials</li> <li>• "mat_to_doc" Returns the linked documents</li> <li>• "doc_to_master" Returns the respective master</li> <li>• "doc_to_nonmaster" Returns the respective dependent document</li> <li>• "doc_to_latest_in_set" Searches for the newest versions in the set</li> <li>• "where_used" Returns the locations where the object is used</li> <li>• &lt;customer function module&gt;: Applies an operation programmed by the customer</li> </ul>
TRANSFORM_SET_WITH_PARAMETERS	Transforms SAP objects by applying certain rules with provided parameters	<pre>&lt;setId&gt; =SET_TRANSFORM_WITH_PARAMETERS( &lt;setId&gt;, &lt;rule&gt;, &lt;parameters&gt; )</pre> <ul style="list-style-type: none"> <li>• "where-used" Indicates where the object in question is used</li> <li>• "expand_structure"</li> <li>• "olinks" Imports the object links of the relevant object</li> </ul>
CALL_CONVERT_FM	Calls a module using a macro, e.g. in a smartcontainer	<pre>&lt;setId&gt; = CALL_CONVERT_FM( &lt;fmName&gt;, &lt;setId&gt;, &lt;[parameters]&gt; )</pre>



### Example: create set test

```

list = KEYLIST_FROM_CONTEXT("active", "selected")
if (list.length == 0) {
  alert("Please, select first an object in SAP ECTR for SAP S/4HANA")
} else {
  list = list.sort();
  setId = CREATE_SET(list)
  list2 = KEYLIST_FROM_SET(setId)
  list2 = list2.sort();
  if (JSON.stringify(list) != JSON.stringify(list2))
    alert("Test failed!" + list + " is not " + list2)
  WRITE_OBJECTLIST(list, "CREATE_SET_TEST")
}

```

### 6.7.3 Display Functions

Function	Purpose	Syntax
WRITE_RESULTLIST	Writes the objects in a <a href="#">KeyList</a> back into the reference container (a <a href="#">SmartList</a> or <a href="#">smartcontainer</a> )	WRITE_RESULTLIST(<keylist>)
DISPLAY_SELECTION	Uses the objects in a <a href="#">KeyList</a> for selecting the corresponding instances in a given SAP ECTR for SAP S/4HANA context	DISPLAY_SELECTION(<context_name>   "active" , <keyList>)
REPORT	Writes a comment line to an SAP ECTR for SAP S/4HANA window. Optionally, an icon can be displayed.	REPORT(<message>.<status_icon>) Possible status icons: Info (I) Warning (W) Error (E) Success (S)
DISPLAY_IN_OBJECT_BROWSER	Displays an object in an <a href="#">Object Browser</a>	DISPLAY_IN_OBJECT_BROWSER(<keyList>)
alert	Produces a notification in a pop-up window	alert("Enter_Message")



### Example: display selection test

```

keylist = KEYLIST_FROM_CONTEXT("active", "selected")
if (keylist.length == 0)
  alert("Please, select object")
else {
  WRITE_OBJECTLIST(keylist, "test_selection")
  DISPLAY_SELECTION("ctx_desk", keylist)
}

```

## 6.7.4 Parameter Maps

The following operators are possible for search parameters: >, <=, >=, !

If multiple parameters are transferred in a list, the operators for each element must be set separately.

Function	Purpose	Syntax
PARAMETER_MAP	Defines a new parameter list object	<pre>&lt;parameters&gt; = PARAMETER_MAP([type])  [type]: "doc_fields" (Use parameters from doc_fields, not from doc_filter_fields, if you want to execute a Search operation.) "mat_fields" "&lt;class&gt;@&lt;class type&gt;" "activity_log_fields" "doc_filter_fields" (Use parameters from doc_filter_fields, not from doc_fields, if you want to execute a Filter operation.) "mat_filter_fields" "olinks" "ecn_fields"</pre>

### Example

```
p = PARAMETER_MAP("doc_fields")
p.DOCUMENTTYPE = "AUX"
p.DESCRPTION = "*"
p.USERNAME = $user
p.MAXROWS = "50"
text = "Search " + p.DOCUMENTTYPE + " Documents from $user"
REPORT(text)
set = SEARCH("doc", p)
list = KEYLIST_FROM_SET(set)
DISPLAY_IN_OBJECT_BROWSER(list)
REPORT("$count Objects found, duration $duration")
```

### Example

```
// Parameter map for documents
P_doc = PARAMETER_MAP( "doc_fields" )
// Parameter map for classification
P_classification = PARAMETER_MAP( "Class1@Classtype1")
P_classification.CONFIGURATION_TYPE = "ST"
// Search for parameters and classification parameters
SET_search = SEARCH( "doc", P_doc, P_classification )
// Keylist from the search
KL_search = KEYLIST_FROM_SET( SET_search )
// Output in object list
WRITE_OBJECTLIST( KL_search, "CL search" )
```

### 6.7.4.1 Default Input for Parameters

Type: "doc\_fields"

Input	Description
AUTHORITYGROUP	Indicates the authorization group
CREATED_ON	Indicates the creation date
DESCRIPTION	Provides a description
DOCUMENTNUMBER	Indicates the document number
DOCUMENTPART	Indicates the document part
DOCUMENTTYPE	Indicates the document type
DOCUMENTVERSION	Indicates the document version
DTYPE	Indicates the DType at hand
ECNUMBER	Indicates the change number
FILENAME	Indicates the file name.
LABORATORY	Indicates the value for "laboratory"
LANGUAGE_ISO	Indicates the language based on its standard ISO code
LOADLATEST	Indicates the most recently edited version
LOADLATESTRELEASED	Indicates the most recently released version
LOEDK	Indicates the deletion flag
MAXROWS	Indicates the maximum number of rows
RES4	Provides the value of the RES4 field
STATUS	Indicates the status.
STATUS_ENDDATE	Specifies the date up to which the document is valid. Warning. Create the date using the following format only: DD, MM, YYYY (e.g. 10, 08, 2016)
STATUS_STARTDATE	Specifies the date from which the document is valid. Warning. Create the date using the following format only: DD, MM, YYYY (e.g. 10, 08, 2016)
USERNAME	Indicates the user name of the person who most recently edited the document

**Type: "mat\_fields"**

Input	Description
DESCRIPTION	Provides a description
GENITEMCAT	Indicates the general object category of the material
GROES	Indicates the size and dimensions
GROSSWEIGHT	Indicates the total weight
LABOR	Indicates the laboratory and engineering office in question
LANGUAGE	Indicates the language
MATKL	Indicates the material group
MATNR	Indicates the material number
MAXROWS	Indicates the maximum number of rows
MTART	Indicates the material type
NETWEIGHT	Indicates the net weight
NORMT	Indicates the standard description
PROD_HIER	Indicates the product hierarchy
SPART	Indicates the division
STATUS	Indicates the status.
VOLUME	Indicates the volume
VOLUMEUNIT	Indicates the volume unit
WEIGHTUNIT	Indicates the weight unit
WRKST	Indicates the basic material

**Type: "activity\_log\_fields"**

Input	Description
LOG_ACTION	Indicates whether a log is maintained
LOG_DAY_COUNT	Indicates how many days the log covers
LOG_OBJECT_KEY	Indicates the log's object key
LOG_OBJECT_TYPE	Indicates the log's object type
LOG_SORT_DESCENDING	Indicates whether the log is sorted by date in ascending or descending order

Input	Description
MAXROWS	Indicates the maximum number of rows

**Type: "doc\_filter\_fields"**

Input	Description
ADATUM	Indicates the change date
AENNR	Indicates the change number
AOC	Indicates whether an additional original is open for editing
AOE	Indicates whether an additional original exists
APPLICATION	Indicates the assigned program
BEGRU	Indicates the authorization group
CAD_ENV	Indicates the preferred CAD environment
CAD_VERSION	Indicates the version of the CAD program
CHECKOUTFILESIZE	Indicates the size of the file
CL_STATUS	Indicates the status of a request
DELFL	Indicates the deletion flag
DKTXT	Indicates the document short text
DOKAR	Indicates the document type
DOKNR	Indicates the document number
DOKST	Indicates the document status
DOKTL	Indicates the document part
DOKVR	Indicates the document version
DTYPE	Indicates the DType
DWNAM	Indicates the user name/processor
FAMILY	Indicates the document family
FLAG2	Provides the value of the FLAG2 field
FLAG4	Provides the value of the FLAG4 field
HASMATERIAL	Indicates whether the object is linked to a material
HASMULTIMATLINK	Indicates whether the object is linked to multiple materials
HASNONMASTER	Indicates whether the object has a dependent document

Input	Description
HNW	Indicates whether the object has newer versions
HRV	Indicates whether the object has released versions
ISASSEMBLY	Indicates whether the object is an assembly
ISDIRTY	Indicates whether the object has a dirty flag
ISDRAWING	Indicates whether the object is an NX object
ISGENERIC	Indicates whether the object is generic
ISINSTANCE	Indicates whether the object is an instance
ISMASTER	Indicates whether the object is a master document
ISPART	Indicates whether the object is a part
LABOR	Indicates the associated laboratory/construction office
LASTMODIFIED	Returns the most recently modified object in a group
MASTERFILENAME	Returns the name of the master
MASTERSTORAGE	Indicates where the master is stored
MATNR_EXT	Indicates the external material number
MATNR_INT	Indicates the internal material number
MAT_MSTAE	Indicates the material status
MAXROWS	Indicates the maximum number of rows
MOC	Indicates whether the original is open for editing
MOE	Indicates whether an original exists
NM_OLD	Lists the dependent documents that are no longer current
REFSET	Indicates the reference set
RES1	Provides the value of the RES1 field
RES2	Provides the value of the RES2 field
RES3	Provides the value of the RES3 field
RES4	Provides the value of the RES4 field
RES8	Provides the value of the RES8 field
RETURNID	Indicates the message class to which the system writes
RETURNMESSAGE	Indicates the message text that the system writes

Input	Description
RETURNNUMBER	Indicates the message number that the system writes
REVLV	Indications the revision level
SAVECOUNTER	Indicates the save counter
STRUCT	Indicates the structure
UNIT	Indicates the unit
VALIDFROM	Indicates validity starting from a certain point in time. For rules on how dates are to be expressed, please refer to the entries for STATUS_ENDDATE and STATUS_STARTDATE.
VALIDTO	Indicates validity up to a certain point in time

**Type: "mat\_filter\_fields"**

See "mat\_fields"

**Type: "olinks"**

Input	Description
LINK_TYPE	Defines the table type of the object displayed

**Type: "ecn\_fields"**

Input	Description
AUTHORIZATION	Specifies the authorization
CHANGED_BY	Indicates who made the last change
CHANGE_DATE	Specifies the change date
CHANGE_REASON	Specifies the change reason
CREATED_BY	Indicates the creator
CREATE_DATE	Specifies the creation date
DESCRIPTION	Specifies the description
FUNCTION	Specifies the function
MAXROWS	Specifies the maximum number of rows
CHANGE_NUMBER	Specifies the change number
STATUS	Specifies the relevant status


Input	Description
TECHV	Specifies the technical status
VALID_FROM	Specifies the validity as from a certain date For date notation rules, observe the entry for STATUS_ENDDATE and STATUS_STARTDATE.

### Search within class

`searchRecursive()` searches for the existing objects within a class, taking into account all subclasses. When the `ParameterMap` is defined, the class type and class for which the `ParameterMap` is valid must be specified. The result is written to the `ParameterMap`.

```
p = PARAMETER_MAP("T0000000@001")
//A ParameterMap is created here. It is defined for which class (here: T00000 - Tool
//Components) and class type (here: 001 - Materials) the ParameterMap is valid.
p.searchRecursive()
//A recursive search within the aforementioned class is carried out. The ParameterMap
//is filled with the objects found.
```

### 6.7.5 User Entries

Function	Purpose	Syntax
ASK_USER	Opens a simple input window for a single value	<pre>&lt;result&gt; = ASK_USER( &lt;fieldId&gt;, &lt;prompt&gt;)</pre> <ul style="list-style-type: none"> <li>• <code>&lt;fieldID&gt;</code> Names the input window</li> <li>• <code>&lt;prompt&gt;</code> Specifies the label of the input window.</li> <li>• <code>[type]</code> Optional: Specifies the type of the input window. Possible values include <ul style="list-style-type: none"> <li>• FILE Displays a selection dialog for a file path. You can also enter the file path yourself.</li> <li>• DATE Displays a selection dialog for a date.</li> <li>• PLMObject Displays a text field for user input during interactive macro execution. The parameter is used in the create window to display the key of PLM objects in a readable manner. Here, a PLM object can be added here by drag&amp;drop into the field.</li> </ul> </li> </ul>
		<p> Note</p>

Function	Purpose	Syntax
		<p>For displaying parameters in the GUI, the source code file is analyzed. The parameters are correctly identified if they are defined as constants and every occurrence of ASK_USER is uniquely identified via fieldID. This restriction is relevant only for displaying parameters in the GUI. When macros are executed, the parameters are correctly evaluated, irrespective of how they are defined.</p>
GET_DATE_AS_STRING	Changes dates of the form DD, MM, YY to a string of the form YYYYMMDD. The value is available on the clipboard.	<DD, MM, YYYY>
GET_MASK_INPUT	Outputs data from the input window to a parameter list.	<pre>&lt;parameters&gt; = GET_MASK_INPUT( &lt;maskName&gt;, &lt;dialogTitle&gt;, &lt;objectType&gt;, [queryName] )</pre> <p>&lt;maskName&gt; The name of a search screen. You can create search screens for specific customers and call them up by name. Search screens for materials, documents, and change numbers are available by default: STANDARD_MAT STANDARD_DOC STANDARD_ECM</p> <p>&lt;dialogTitle&gt; Defines the name of the screen</p> <p>&lt;objectType&gt; Defines the object type of the output objects</p> <p>[queryName] The search ID</p>
confirm	Opens a dialog that asks the user to confirm or reject something. Outputs True or False, depending on whether a confirmation or rejection was entered.	<pre>&lt;result&gt; = confirm(&lt;prompt&gt;)</pre> <p>&lt;prompt&gt; Indicates the text of the dialog.</p>



#### Example: ask user test

```
result = ASK_USER("field", "Input 5")
var messageif (result != 5)
    message = "you input " + result + " instead of 5"
else
    message = "Test passed!"
alert(message)
Get mask input test
p = GET_MASK_INPUT("STANDARD_DOC", "Document mask", "DOC", "someId1")
```

```

REPORT("----Document mask fields----");
for ( var key in p ) {
    if ( p.hasOwnProperty(key) && !key.startsWith("_+DSC")) {
        REPORT(key + "=" + p[key]);
    }
}
p = GET_MASK_INPUT("STANDARD_MAT", "", "MAT", "someId2")
REPORT("---Material mask fields----");
for ( var key in p ) {
    if ( p.hasOwnProperty(key) && !key.startsWith("_+DSC")) {
        REPORT(key + "=" + p[key]);
    }
}
}

```

### 6.7.6 Searching, Filtering

Function	Purpose	Syntax
GET_SEARCHHELP_INPUT	The command is used to display the search help GUI. The command receives the name of the search help as a parameter. You can also search for functional locations or equipment in this way.	<pre>p = GET_SEARCHHELP_INPUT('&lt;Elementary Search Help&gt;')</pre> <p>The search command with "SH" or "SEARCHHELP" as the object type parameter is used for the search:</p> <pre>setId = SEARCH( 'SH', p )</pre>
FILTER	Executes a filter operation on an <i>ObjectSet</i> .	<pre>&lt;setId&gt; = FILTER( &lt;setId&gt;, &lt;objectType&gt;, &lt;parameters&gt;, [&lt;classificationParameters&gt; ])</pre>
SEARCH	Executes a search operation in the SAP system.	<pre>&lt;setId&gt; = SEARCH(&lt;objectType&gt;, &lt;parameters&gt;, [&lt;classificationParameters&gt; ] )</pre> <p>&lt;objecttype&gt; Possible values are: doc, mat, ecn, work_item</p>
SEARCH_AND_RETURN	Executes a search operation in an SAP system and returns the values immediately.	<pre>&lt;setId&gt; = SEARCH_AND_RETURN (&lt;objectType&gt;, &lt;parameters&gt;, [&lt;classificationParameters&gt; ] )</pre> <p>&lt;objecttype&gt; Possible values are: doc, mat, ecn, work_item</p>
SEARCH_BY_FILENAME	Search function by file name	<pre>&lt;setId&gt; = SEARCH_BY_FILENAME( &lt;parameters&gt; )</pre>
FILTER_BY_TYPE	Executes a search operation in the SAP system and returns a set with a specified object type.	<pre>&lt;setId&gt; = FILTER_BY_TYPE(&lt;setId&gt; , &lt;objectType&gt; )</pre> <pre>resultSetId = FILTER_BY_TYPE(setId , "DOC" ) // returns a set that only contains documents with the setId</pre> <pre>resultSetId = FILTER_BY_TYPE(setId , ["DOC", "MAT"] ) // returns a set that</pre>

Function	Purpose	Syntax
		only contains documents and materials with the setId
SEARCH_BY_CHARACTERISTIC	Performs a search operation for a characteristic. The specification of the class type is mandatory.	setId = SEARCH_BY_CHARACTERISTIC ([characteristic name without quotation marks])



### Example - Search help

```
p = GET_SEARCHHELP_INPUT( 'ZVKO_CHANGE_REC_SEARCH_BY_USER' )
setId = SEARCH( 'SH', p )
keylist = KEYLIST_FROM_SET( setId )
WRITE_OBJECTLIST( keylist, 'Change Records' )
```



### Example: search test

```
p = PARAMETER_MAP("mat_fields")
p.MTART = "HALB"
setId = SEARCH("mat", p)
keylist = KEYLIST_FROM_SET(setId)
WRITE_OBJECTLIST(keylist, "Materials:HALB")
Search and return test
p = PARAMETER_MAP("doc_fields")
p.DESCRPTION = "*Test*"
SEARCH_AND_RETURN("doc", p)
// result in Objectlist -> My Test List
filter test
p = PARAMETER_MAP("mat_fields")
p.MTART = "FERT;HALB"
searchSet = SEARCH("mat", p)
fp = PARAMETER_MAP("mat_filter_fields")
fp.MTART = "HALB"
setId = FILTER(searchSet, "mat", fp)
keylist = KEYLIST_FROM_SET(setId)
WRITE_OBJECTLIST(keylist, "Materials:HALB")
```



### Example - SEARCH\_BY\_CHARACTERISTIC

```
p = PARAMETER_MAP()
p.CLASS_TYPE = "017"
p.MAXROWS = 3
p.U_SUBTYPE = "TEST"
p.DRO_RESPONSIBLE = "07"
setId = SEARCH_BY_CHARACTERISTIC(p)
keylist = KEYLIST_FROM_SET(setId)
WRITE_OBJECTLIST(keylist, "Standard-Liste")
```

### 6.7.7 Calling functions

Function	Purpose	Syntax
CALL_OMF	Calls an SAP ECTR for SAP S/4HANA function. Only functions which do not need a tree selection and that are executed in the desktop window work can be used here.	CALL_OMF( <omfName>, [<keylist>] ) <omfName> Name of the function to be called
CALL_OMF_EXT	Calls a SAP ECTR for SAP S/4HANA-function. How the program deals with the function can be specified via different options.	<pre>CALL_OMF_EXT({ 'omfName': '&lt;FunctionName&gt;', 'context' : '&lt;ContextName&gt;', 'waitForExecution': &lt;true/false&gt;, 'contextSelection': '&lt;NONE/PROVIDED/SELECTED/ALL&gt;', 'keylist': '&lt;keylistName&gt;', })</pre> <p>omfName Specifies name of the function.</p> <p>'context' Specifies the context key for the OMF. If the parameter is missing or has ACTIVE as a value, then the key is used by the current context. The parameter is optional, 'ACTIVE' is used as default value.</p> <p>'waitForExecution' Specifies whether the macro should wait until the OMF is executed. The parameter is optional, default value is false.</p> <p>'contextSelection' Specifies how the selection for the function is to be determined. The following values are supported: SELECTED, ALL, NONE, PROVIDED NONE - empty ObrTreeSelection object. The value is optional  PROVIDED - objects for selection are passed along with the keylist parameter. The value is optional. The value is set automatically if a keylist parameter is defined.  SELECTED - ObrTreeSelection contains selected objects and TreeNode from the context.  ALL - ObrTreeSelection contains all objects</p> <p>Keylist Specifies objects used in the function. The parameter is used if the contextSelection is missing or has PROVIDED as a value.</p>
GET_SERVICE	Calls a service interface class and returns a service implementation or, if the service is not available, 0.	GET_SERVICE(<Service Interface Class>)



### Example: call OMF

```
CALL_OMF("fnc.tmp.fld.create")
```



### Beispiel - call OMF EXT

```
CALL_OMF_EXT({
  'omfName': 'fnc.doc.viewer.show',
  'context' : 'ctx_desk',
  'waitForExecution': true,
  'contextSelection': 'SELECTED'
})
```



### Example: call OMF with single object

```
list = KEYLIST_FROM_CONTEXT("active", "selected")
if (list.length == 0)
  alert("Please, select document to show in viewer")
else
  CALL_OMF("omf.doc.viewer.show", list)
```



### Example: call OMF with multiple objects

```
list = KEYLIST_FROM_CONTEXT("active", "selected")
list = KEYLIST_FROM_CONTEXT("active", "selected")
if (list.length <= 1)
  alert("Please, select more than one object ")
else
  CALL_OMF("fnc.tmp.fld.handle.set", list)
```

## 6.7.8 Object Properties

Function	Purpose	Syntax
GET_PROPERTY	A specified property of an object. If properties of an object are not set, the object returns <i>undefined</i> .	<pre>&lt;propertyValue&gt; = GET_PROPERTY(   &lt;propertyName&gt;/&lt;propertyNames&gt;, &lt;keylist&gt; )</pre> <pre>&lt;propertyName&gt;/&lt;propertyNames&gt;</pre> <p>Indicates the respective preferred property. For a list of the objects in question, please refer to the examples involving object properties.</p>
GET_CHARACTERISTIC	Returns the values of the SAP classification. The values must be available in a keylist.	<pre>&lt;CharacteristicValue&gt; = GET_CHARACTERISTIC   (&lt;characteristicName&gt;/&lt;characteristicNames&gt;,   &lt;keylist&gt;)</pre> <pre>&lt;characteristicName&gt;/&lt;characteristicNames&gt;</pre> <p>Indicates the SAP property that is to be modeled by the value.</p> <pre>&lt;keylist&gt;</pre> <p>Indicates the keylist from which the values are transferred.</p>
CREATE_PROPERTY_COLLECTOR	Collects properties of objects. The property is	<p>Read:</p> <pre>collector =   CREATE_PROPERTY_COLLECTOR(resultList)</pre>

Function	Purpose	Syntax
	<p>transferred as a keylist. Returns collector object. The collector object supplies the methods to retrieve the properties from the objects in the keylist.</p>	<p>Specifies the property value of individual elements:  <code>collector.value(&lt;element&gt;, &lt;property&gt;)</code></p> <p>Collects the values as displayed in the GUI:  <code>collector.guiValue(&lt;element&gt;, &lt;property&gt;)</code></p> <p>The value container collects all non-unique values. If a value appears twice, it is still left in the list. The value container has a field for an internal value (e.g. status AC) that is not displayed in the GUI and an external value (e.g. status Accessible) that is displayed for the user in the GUI.  <code>ValueContainer = valueContainer(&lt;element&gt;, &lt;property&gt;)</code>  <code>ValueContainer.internal</code></p> <p>The unique values of the objects from the list are transferred here.  <b>Unique internal values that are not displayed in the GUI:</b>  <code>collector.distinctValues(&lt;propertyKey&gt;)</code></p> <p><b>Unique external values as displayed in the GUI:</b>  <code>collector.distinctGuiValues(&lt;propertyKey&gt;)</code></p> <p><b>All unique values</b>  <code>collector.distinctValueContainers(&lt;propertyKey&gt;)</code></p> <p><b>Static grouping, returns map in the form &lt;Key, List&lt;KEYLIST_ITEMS&gt;&gt;.</b>  <code>collector.groupByPropertyValue()</code>  <code>collector.groupByValueContainer()</code></p>

### Example – CREATE\_PROPERTY\_COLLECTOR

```

keylist = KEYLIST_FROM_CONTEXT('ACTIVE', 'SELECTED')
REPORT(`Selected ${keylist.length} elements`)

collector = CREATE_PROPERTY_COLLECTOR(keylist)

REPORT('Loop through keylist')
for (item of keylist) {
    value = collector.value(item, 'USER')
    REPORT(`Property USER for object ${item} is ${value}`)
}

```



### Example: get document property test

```
list = KEYLIST_FROM_CONTEXT("active", "selected", "doc")
if (list.length == 0) {
    alert("Please, select document in active context")
} else {
    properties = [ "object_type", "number", "type", "part", "version",
                  "description", "status_description", "user",
                  "authorization_group",
                  "laboratory", "change_number", "revision",
                  "last_sap_error_message",
                  "dtype", "dtype_description",
                  "lastmodified", "master", "family", "dirty", "res1",
                  "res2",
                  "res3", "res4", "status", "assembly", "appl_type",
                  "material_number", "valid_from", "valid_to", "create_date",
                  "replace_by", "storage_category", "master_filename",
                  "drawing",
                  "cad_system", "cad_version", "cad_environment",
                  "cad_version_and_environment", "dtype_description_and_id",
                  "valid_from_to", "status_id_and_description",
                  "checkout_size",
                  "cad_indicator", "acc_owned", "sapkey", "normpart" ]
    allPropertiesString = ""
    properties.forEach(function(property) {
        allPropertiesString += property.toUpperCase() + "="
            + GET_PROPERTY(property, list) + "\n"
    })
    alert(allPropertiesString)
}
```



### Example: get change number property test

```
list = KEYLIST_FROM_CONTEXT("active", "selected", "ecm")
if (list.length == 0) {
    alert("Please, select document in active context")
} else {
    properties = [
        "change_number", "status", "status_description",
        "release_key", "authorization_group", "valid_from",
        "description", "change_reason", "created_on", "created_by",
        "changed_on", "changed_by",
        "technically_released", "function",
        "last_sap_error_message", "long_text", "acc_owned",
        "change_number_type"
    ]
    allPropertiesString = ""
    properties.forEach(function(property) {
        allPropertiesString += property.toUpperCase() + "="
            + GET_PROPERTY(property, list) + "\n"
    })
    alert(allPropertiesString)
}
```



### Example: get work item property test

```

list = KEYLIST_FROM_CONTEXT("active", "selected", "obj_wi")
if (list.length == 0) {
    alert("Please, select document in active context")
} else {
    properties = [
        "object_type", "handle", "id", "type", "technical_status",
        "status", "description", "created_at", "latest_end_date",
        "created_by", "note_count", "superordinate_item", "client",
        "language", "end_message", "missed_deadline",
        "task", "actual_processor", "task_id", "priority",
        "forwarder", "type_text", "task_classification",
        "deadline_status", "status_description", "processor_id",
        "remote_id", "system", "definition_type", "overdue",
        "designated_start_date", "latest_start_date",
        "designated_end_date", "processed_from", "processed_to",
        "open_requests"
    ]
    allPropertiesString = ""
    properties.forEach(function(property) {
        allPropertiesString += property.toUpperCase() + "="
            + GET_PROPERTY(property, list) + "\n"
    })
    alert(allPropertiesString)
}

```

### 6.7.9 Environment Variables

Function	Purpose	Syntax
READ_ENVVAR	Reads the value of a Windows environment variable defined at runtime. To view the environment variables specific to SAP ECTR for SAP S/4HANA that are defined at runtime, please refer to the console log file (for example). You can access this log file through the start menu using the command <a href="#">Track ECTR Log Files</a> . You can also define Windows-specific environment variables in your system settings.	<varValue> = READ_ENVVAR(<varName>)
READ_PREFERENCE	Reads an SAP ECTR for SAP S/4HANA standard preference value.	<prefValue> = READ_PREFERENCE( <prefName> )
WRITE_PREFERENCE	Writes a string to an SAP ECTR for SAP S/4HANA preference variable.	WRITE_PREFERENCE( <prefName>, <prefValue> )



### Example: read environment variables test

```

plmBuild = READ_ENVVAR("PLM_BUILD")
alert(plmBuild)

```



### Example: read/write preference test

```
WRITE_PREFERENCE("plm.test." + $user, "4711")
result = READ_PREFERENCE("plm.test." + $user)
if ("4711" != result)
    alert("Test was not passed. 4711 is not " + result)
```

### 6.7.10 File Operations

Paths in macros are mapped as follows:

```
filepath = "C:" + "\\\" + "test_folder" + "\\\" + "textfile.txt"
```

Paths can still be imported using the READ\_ENVVAR function.

Function	Purpose	Syntax
FILE_CLOSE	Closes an open file. The file must be closed in order to save changes.	CLOSE_FILE( <filehandle> )
FILE_OPEN	Opens a text file for reading or writing. A file must be open before operations can be carried out on it.	<pre>&lt;filehandle&gt; = FILE_OPEN( &lt;filepath&gt;, &lt;mode&gt; )</pre> <p>&lt;mode&gt; Indicates the permissions with which the file can be opened. "R" - read permissions "W" overwrites all of the file's content with the next line inserted using FILE_WRITELINE. Every additional FILE_WRITELINE produces a new line. If the file is opened with w and does not yet exist, the file is created. A leaves the file's existing content unchanged and adds a new line to the end of the file the first time FILE_WRITELINE is used. If the file is opened with A and does not yet exist, an error occurs.</p>
FILE_READLINE	Writes a line from an open text file	<line_content> = FILE_READLINE( <filehandle> )
FILE_WRITELINE	Writes a line to an open text file	FILE_WRITELINE( <filehandle>, <line> )
FILE_EXISTS	Tests whether a file or a folder exists (True/false).	<exists> = FILE_EXISTS( <filename> )



### Example: file operations test

```
tmp = READ_ENVVAR("TEMP")
filepath = tmp + "\\\" + "my_test_file.txt"
h = FILE_OPEN(filepath, "W")
isExists = FILE_EXISTS(tmp)
if (!isExists)
    alert("Test failed. FILE_EXISTS for " + tmp + " should be true")
isExists = FILE_EXISTS(filepath)
if (!isExists)
    alert("Test failed. FILE_EXISTS for " + filepath + " should be true")
```

```

FILE_WRITELINE(h, "first line")
FILE_WRITELINE(h, "second line")
FILE_CLOSE(h)
h = FILE_OPEN(filepath, "A")
FILE_WRITELINE(h, "append line")
FILE_CLOSE(h)
h = FILE_OPEN(filepath, "R")
line = FILE_READLINE(h)
if ("first line" != line)
    alert("Test failed. first line is not " + line)
else
    REPORT(line)
line = FILE_READLINE(h)
if ("second line" != line)
    alert("Test failed. second line is not " + line)
else
    REPORT(line)
line = FILE_READLINE(h)
if ("append line" != line)
    alert("Test failed. append line is not " + line)
else
    REPORT(line)
FILE_CLOSE(h)

```

### 6.7.11 Dictionary-Entries

Function	Purpose	Syntax
READ_DICTIONARY_TEXT	The user may access dictionary entries that are stored in the Basis directory and in the Customize directory.	<pre>text = READ_DICTIONARY_TEXT(&lt;dictionaryKey&gt;, [&lt;defaultValue&gt;], [&lt;argumentArray&gt;])</pre> <p>&lt;dictionaryKey&gt; Indicates the dictionary key that is to be accessed.</p> <p>[&lt;defaultValue&gt;] Indicates the standard value that is used if the key cannot be found.</p> <p>[&lt;argumentArray&gt;] Enables the user to transfer arrays.</p>

### 6.7.12 MacroState

The status of a macro can be queried in the script. This only works if the script is currently in a front-end process. Therefore, the query does not work during operations on ObjectSets whose content has not yet been returned to a KeyList.

Function	Purpose
MacroState.isAborted()	<p>Returns the value <code>true</code> if the execution of the macro has been interrupted with <code>ESC</code>. Can be used to interrupt the process with <code>ESC</code> using a loop in the script.</p> <p>If the script is currently in the back end, this function is not necessary since the execution of the macro is interrupted automatically with <code>ESC</code>.</p>

Function	Purpose
MacroState.runtime()	Returns the runtime of a macro (in nanoseconds).
MacroState.objectCount()	Returns the number of objects processed during the last operation.



### Example: Macro.State

```
//use only as example as CPU is under heavy load when executing this function
function sleep(delay) {
    var start = new Date().getTime();
    while (new Date().getTime() < start + delay);
}

function main() {
    REPORT( "start delay1" )
    sleep(3000)
    if(MacroState.isAborted() == false){
        REPORT( "start delay2" )
        sleep(3000)
    }
    REPORT("isAborted=" + MacroState.isAborted())
}
// Execute the function main
main()
```

## 6.8 Processable, Static Variables in Macros

The following static variables can be used in macros:

Variable	Purpose
\$user	Contains the relevant user.
\$os-user	Contains the relevant user of the operating system.
\$client	Contains the relevant client.
\$language	Contains the relevant language.
\$system	Contains the relevant system.
\$count	Contains the number of objects in a keylist. Observe the note below.
\$duration	Contains the duration of an operation in seconds. Observe the note below.
\$execution_environment	Contains the relevant usage type for the macro. The following can be output: SMARTLIST, SMARTCONTAINER, EDITOR, OMF, UNKNOWN.

### **i** Note

By default, the environment variables \$count and \$duration deliver results only after conversion of a set to a keylist and can currently be used in the function REPORT() only.

To receive valid results before the conversion, you need to run macros in sequential mode. To do this, enter the following command in the macro file:

```
$SET_EXECUTION_OPTION("BUFFERED", "false")
```

The buffer mode is activated by default. To switch to this mode, enter the command:

```
$SET_EXECUTION_OPTION("BUFFERED", "true")
```

## 6.9 Execute Macro with SAP GUI Transaction

If the parameter `$SET_EXECUTION_OPTION('use_gui_transaction', 'true')` is set, the macro can be executed with an SAP GUI transaction. The parameter is not set by default. This is interpreted as `use_gui_transaction=false`.

## 6.10 Create System SmartLists

System SmartLists are SmartLists that are always available for all instances.

### 6.10.1 Create System SmartLists

SmartLists are created in the user directory in `smartlists.xml`.

They are defined as System SmartLists using the following command:

```
custom.function.group="SYSTEM_SMARTLIST"
```

### 6.10.2 Configure System SmartLists

This approach means that SmartLists do not have to be adjusted for every individual instance. System SmartLists cannot be adjusted from individual instances; instead, they must be centrally adjusted using `default.txt`.

The System SmartLists are created using the following variable:

```
plm.system.smartlists = <smartlist_1>;<smartlist_2>;...;<smartlist_n>
```

The label displayed for the System SmartLists is maintained in `default.txt`. As a result, the definitions in the dictionary of the relevant language version are used.

```
plm.smartlist.dic.<SmartList_Name> = <label>
```

System SmartLists can be renewed automatically. The delay between the automatic renewals is specified in milliseconds as follows:

```
plm.system.smartlist.<Smartlist_Name>.delay = <time in ms>
```

Entries in the SmartList are sorted based on the SAP ECTR for SAP S/4HANA settings by default. Entries can be sorted using the following variables.

- Depending on the time when they were added (`true`)
- Displayed alphabetically (`false`)

```
plm.system.smartlist.<SmartList_Name>.ignore.tree.sort.
```

In addition, you must define the icons in the tree for the System SmartLists. You can find the relevant icons in the `icons.jar` in `<PLM_INSTDIR>/basis/classes`.

If you create your own icons, these icons can be placed in the following directory:

```
<PLM_INSTDIR>/customize/classes/patch
```

The name of the icon is its path from the icon source directory, for example:

```
plm.system.smartlist.LAST_EDIT.icon = {0}/sap/edit
```

### 6.10.3 Predefined Parameters

If the value of the relevant parameter is not editable, the PARAMS array is used:

In `default.txt`:

```
plm.system.smartlist.<SmartList_Name>.params.0 = DRAW
```

In the macro, everything is called using the following call:

```
otype = PARAMS[0]
```

If a parameter is to be defined by the user, the `ASK-USER` command can be used here.

The values for `ASK_USER` can be prefilled in `default.txt`.

```
plm.system.smartlist.<Smartlist_Name>.userkey.0 = OTYPE  
plm.system.smartlist.<Smartlist_Name>.uservalue.0 = DRAW
```

In the macro itself, the values can be called using the following call:

```
ASK_USER("OTYPE", "Object type")
```

If there are no prefilled parameters, an input dialog is displayed on initial start.

At the same time, the corresponding OMFs `fnc.edit.smart.list.search.input` and `fnc.edit.smart.list` must be changed for editing by the administrator.

Existing System SmartLists without user entries can be changed by the administrator.

If an existing System SmartList has user entries, the name of the System SmartList is changed when changes are made. The previous SmartList is then removed from the `smartlist.xml` file.

To display dialogs and masks every time a macro is executed,

```
plm.system.smartlist.<Smartlist_Name>.always.display.searchmask.
```

can be set to `true`.

### 6.11 Adding Objects to a Folder Using a Search Macro

Documents can be added to a folder using a search defined in a macro. The associated function is configured in a menu (for example, in `menu_macros.txt`).

When configuring a folder, you can specify a macro. This macro is executed whenever the folder is opened or updated.

Macros can be referenced from various paths.

```
? FOLDER_ADD  
  
= fnc.add.with.macro(macro.txt)
```

You use the above syntax if the macro (`macro.txt`) is saved under the following path:

```
ECTR_INST/customize/scripts/macros/wizard/
```

```
? FOLDER_ADD  
  
= fnc.add.with.macro(wrd:macro.txt)
```

You use the above syntax if the macro (`macro.txt`) is saved in the application folder "`wrd`". The same syntax also works for other application folders.

```
ECTR_INST/applications/wrd/customize/scripts/macros/wizard/
```

```
? FOLDER_ADD
```

```
= fnc.add.with.macro(mso:macro.txt)
```

You use the above syntax if the macro (`macro.txt`) is saved in the add-on folder "`mso`". The same syntax also works for other add-on folders.

```
ECTR_INST/addons/mso/customize/scripts/macros/wizard/
```

If you do not specify a macro during configuration, users can select a macro after the function has been executed.

Example:

```
? FOLDER_ADD
```

```
= fnc.add.with.macro
```

**7.1 URL Access to Documents and Materials in the SAP System**

SAP ECTR for SAP S/4HANA provides the option to access documents and materials in the SAP system via a URL. To do this, SAP ECTR for SAP S/4HANA installs a URL handler for the protocol in the operating system.

If a link with this protocol is clicked, the application registered for it is called. The application transfers the URL to SAP ECTR for SAP S/4HANA for further evaluation. SAP ECTR for SAP S/4HANA uses the currently logged on user to execute the action described in the URL. If SAP ECTR for SAP S/4HANA is not running already, it is started.

**7.1.1 Structure of the URL**

A URL for protocol `sapplm://` has the following structure:

`sapplm://ectr?<Parameter list>`

The following parameters have been defined:

Parameter	Description
<code>system=&lt;SAP system&gt;</code>	Name of the SAP system in which the object is located, for example D11
<code>client=&lt;SAP client&gt;</code>	Number of the SAP client, for example 100
<code>type=&lt;Object type&gt;</code>	Object type: mat = material doc = document
<code>sapkey=&lt;Key&gt;</code>	SapKey of the object
<code>filename=&lt;file name&gt;</code>	File name of the original in the document
<code>wsappl=&lt;WSAPPL&gt;</code>	Workstation application of the original in the document
<code>action=&lt;action&gt;</code>	Action to be executed: open = open as read-only ob = display in Object Browser edit = open for editing adol = Add object from the active list and select the object in the active list (this also results in it being displayed in the Object Browser) If no action is specified, open is assumed as default.
<code>loadrule=&lt;load rule&gt;</code>	The load rule to be used for opening documents: assaved = opens the version of the document that is specified in the URL. If the parameter is skipped, this load rule is the default. latest = opens the latest version of the document. released = opens the latest released version of the document. If there is no released version of the document, an error message is output.

## Examples of URL Access to Documents:

### **Open document (via fnc.doc.open.fast)**

- `sapplm://ectr?type=doc&sapkey=UGM000000000000001000001120602000&action=open`
- `sapplm://ectr?type=doc&sapkey=UGM000000000000001000001120602000&action=open&loadrule=assaved`

Opens the version of the document that is specified in the URL.

### **Open SAP original via file name**

- `sapplm://ectr?type=doc&sapkey=UGM000000000000001000001120602000&filename=test.txt&action=open`

### **Open SAP original via WSAPPL**

- `sapplm://ectr?type=doc&sapkey=UGM000000000000001000001120602000&wsappl=PDF&action=open`
- `sapplm://ectr?type=doc&sapkey=UGM000000000000001000001120602000&wsappl=PDF&action=open&loadrule=assaved`

The first PDF file found in the document is used.

### **Open the latest document version**

- `sapplm://ectr?type=doc&sapkey=UGM000000000000001000001120602000&action=open&loadrule=latest`

Opens the latest version of the document.

### **Open the latest released document version**

- `sapplm://ectr?type=doc&sapkey=UGM000000000000001000001120602000&action=open&loadrule=released`

Opens the latest released version of the document. If there is no released version of the document, then an error message is output.

### **Open SAP original of the latest document version**

- `sapplm://ectr?type=doc&sapkey=UGM000000000000001000001120602000&filename=test.txt&action=open&loadrule=latest`
- `sapplm://ectr?type=doc&sapkey=UGM000000000000001000001120602000&wsappl=PDF&action=open&loadrule=latest`

Opens the latest version of the document.

### **Open SAP original of the latest released document version**

`sapplm://ectr?type=doc&sapkey=UGM000000000000001000001120602000&filename=test.txt&action=open&loadrule=released`

`sapplm://ectr?type=doc&sapkey=UGM000000000000001000001120602000&wsappl=PDF&action=open&loadrule=released`

Opens the latest released version of the document. If there is no released version of the document, then an error message is output.

## **7.1.2 Context Menu Function for URL Generation**

For the selected document or material, the context menu function `fnc.create.url` generates a URL for the protocol `sapplm://`. The URL is stored in the Windows clipboard for further use.

The function is defined as follows:

```
fnc.create.url (<Action> [, <LoadRule>])
```

<Action> specifies the action to be executed:

- `ob` = display in [Object Browser](#)

- `open` = open in assigned application

**The optional parameter <LoadRule> is used to specify the load rule to be used:**

- `assaved` = opens the version of the document that is specified in the URL. If the parameter is skipped, this load rule is the default.
- `latest` = opens the latest version of the document.
- `released` = opens the latest released version of the document.

## 7.2 Start environment

SAP ECTR for SAP S/4HANA processes three types `plm_initialize.*` / `plm_setenv.*` files on start, in the mentioned sequence. (The first existing file is being processed, other enhancements are then skipped, even if the files exist).

- `plm_initialize.bat` / `plm_setenv.bat`
- `plm_initialize.js` / `plm_setenv.js`
- `plm_initialize.env` / `plm_setenv.env`

Along the known `.bat`-files in SAP ECTR for SAP S/4HANA, the start environment can also be defined via `env`- or `.js`-files. Simple text files to set or enhance environment variables.

### **.env files**

Simple text files to set and enhance environment variables.

- It is possible to link to other variables: `%TEMP%\SAP\ECTR\temp`
- `if` / `else` commands are not possible
- UTF-8 coding

### **Example:**

```
# SET
ABC = xyz
QWE=%TEMP%\data

# APPEND
ABC += val1
ABC += val2
ABC += val3
# Result: ABC = val1;val2;val3

# PREPEND
ABC += val1
ABC += val2
ABC += val3
# Result: ABC = val3;val2;val1
```

### **.js files**

Simple JavaScript files to set environment variables.

- It is possible to link to other variables: `"%TEMP%\SAP\ECTRtemp"`

- if / else commands are possible
- UTF-8 coding

The following functions are possible:

```
void    setEnv(envName, envValue)
string  getEnv(envName)
void    appendEnv(envName, envValue, separator = ";")
void    prependEnv(envName, envValue, separator = ";")
boolean exists(filename)
boolean createDirs(dirName)
```

### Example:

```
/**
 * Define the location where the SNC library is located. The correct
 * entry depends on the type of security mechanism your SAP system uses.
 * Please note that this is only an example, the correct type depends on
 * your system setup (SapCrypto | Keberos | NTLM)
 */

setEnv("SSO_TYPE", "")
if(getEnv("SSO_TYPE")== "")
    setEnv("SSO_TYPE", "SapCrypto")

if(getEnv("SSO_TYPE")== "SapCrypto")
    setEnv("PLM_SNC_LIB", "%SNC_LIB_64%")
else if(getEnv("SSO_TYPE")== "Kerberos")
    setEnv("PLM_SNC_LIB", "%SystemRoot%\SysWOW64\gx64krb5.dll")
else if(getEnv("SSO_TYPE")== "NTLM")
    setEnv("PLM_SNC_LIB", "%SystemRoot%\SysWOW64\gx64ntlm.dll")
```

## 7.3 Processing Preference Variables

Preference variables are processed in a certain order. You can define the order in the list `system.start.custpref`.



### Note

Files processed at a later point in time can overwrite set preference variables.

When started, SAP ECTR for SAP S/4HANA automatically searches for the preference files. The files are written to the list in the following order:

1. <INST>\basis\config\system-default.txt
2. <INST>\customize\config\SAPlogon.txt (or Content of PLM\_LOGON\_SETTINGS)

After logging on to the SAP system, additional preference files are determined. The preference files are also written to the list `system.start.custpref`. If an existing file is found, the first entry is removed and the file is added anew at the end.

In this definition, <APP> is the alphabetically sorted list of applications, which is the result of a listing of directory <INST>\applications. SAP ECTR for SAP S/4HANA only takes into account names that have three characters. If a file called <INST>\applications\<APP>\DISABLED exists, the application is ignored.

In this definition, <ADDON> is the alphabetically sorted list of add-ons, which is the result of a listing of directory <INST>\addons. If a file called <INST>\addons\<ADDON>\DISABLED exists, the add-on is ignored.

SAP ECTR for SAP S/4HANA only takes into account existing files. This results in the following processing order:

- <INST>\customize\config\SAPlogon.txt (or content of PLM\_LOGON\_SETTINGS)
- <INST>\basis\config\\*default.\*
- <INST>\customize\config\default.txt
- <INST>\customize\config\ default\_<plm\_location>.txt
- <INST>\customize\config\ default\_<plm\_role>.txt
- <INST>\customize\config\ menu.guidef
- <INST>\customize\config\ menu\_<plm\_location>.guidef
- <INST>\customize\config\ menu\_<plm\_role>.guidef
- <INST>\applications\<APP>\customize\config\default.txt
- <INST>\applications\<APP>\customize\config\ default\_<plm\_location>.txt
- <INST>\applications\<APP>\customize\config\ default\_<plm\_role>.txt
- <INST>\applications\<APP>\customize\config\ menu.guidef
- <INST>\applications\<APP>\customize\config\ menu\_<plm\_location>.guidef
- <INST>\applications\<APP>\customize\config\ menu\_<plm\_role>.guidef
- <INST>\applications\<ADDON>\customize\config\default.txt
- <INST>\applications\< ADDON >\customize\config\ default\_<plm\_location>.txt
- <INST>\applications\< ADDON >\customize\config\ default\_<plm\_role>.txt
- <INST>\applications\< ADDON >\customize\config\ menu.guidef
- <INST>\applications\< ADDON >\customize\config\ menu\_<plm\_location>.guidef
- <INST>\applications\< ADDON >\customize\config\ menu\_<plm\_role>.guidef

You can define additional files in environment variable PLM\_PREEFILES. To do this, create a comma or semicolon-separated list in plm\_setenv.bat. The files from this list are inserted at the end of system.start.custpref. If the definition contains references to environment variables, these references are expanded. If no path is specified, SAP ECTR for SAP S/4HANA inserts <INST>\customize\config by default.

All .guidef files are converted to preference files in the specified order and replaced in the list. You can reference macros in the .guidef files. You can distribute the macros to several definition files. SAP ECTR for SAP S/4HANA writes the macro files into a list and processes the files in the specified order. You can nest the macro references and reference one macro to another macro.

### Note

Subsequently processed macros can overwrite defined macros. In the conversion of the menu files, the order of the macros no longer matters.

SAP ECTR for SAP S/4HANA only takes into account existing macros. This results in the following processing order:

```
<INST>\basis\config\menu_macros.txt
<INST>\applications\<APP>\customize\config\ menu_macros.txt
<INST>\ addons \< ADDON >\customize\config\ menu_macros.txt
```

## 7.4 Additional Settings

### 7.4.1 Response of SAP ECTR for SAP S/4HANA after Starting the Primary Application

The response of SAP ECTR for SAP S/4HANA after starting the primary application can be specified using preference variable

- plm.appl.start.omAction.default

or application-specifically via preference variable

- plm.appl.start.omAction.<application\_type>

This is located in the default.txt file.

#### Available values are

Value	Description
STAY	SAP ECTR for SAP S/4HANA stays open and in front.
HIDE	SAP ECTR for SAP S/4HANA starts up without visible main window. It is accessible via application it started with or tray-icon.
MINIMIZE	SAP ECTR for SAP S/4HANA will be minimized to the system toolbar.

### 7.4.2 Starting the Primary Application

To be able to ensure the starting of the primary application for a document for which no explicit assignment to an application exists, you should make sure that variable

```
plm.appl.startCommand.default = LaunchRegisteredApplication.exe "$(filename) "
```

is set in the default.txt file.

It is also possible to define special commands for the respective primary application.

```
plm.appl.startCommand.<application type>
```

Since this depends on the application type, you should keep in mind that this is not possible if a supported application file has been assigned as an additional original to the document of a document of another application.



#### Example

```
plm.appl.startCommand.UGS = ugs_router.exe -ug "$(filename) " -opts -view
```

### 7.4.3 Starting the Application for Additional Originals

You can use the DType definition to specify whether an additional original can be opened for display or editing. This is done using the application that is assigned to the additional original in the operating system. However, a different application can also be defined for opening the additional original, which can also be dependent on the operating system.

The following priority applies to preference variables:

- 1: system.appl.view.<extension>.<os>                    system.appl.edit.<extension>.<os>
- 2: system.appl.view.<extension>                        system.appl.edit.<extension>
- 3: system.appl.view.default.<os>                      system.appl.edit.default.<os>



#### Example

```
system.appl.edit.default.windows = LaunchRegisteredApplication.exe "$(filename) " -  
edit  
system.appl.view.default.windows = LaunchRegisteredApplication.exe "$(filename) " -  
view
```

### Default Settings for Basic Data in the Object Browser

For documents, materials and change numbers, the display of basic data can be configured in the layout manager of the [Object Browser](#) via the property layout.

To avoid the display of basic data being empty if no layout has been assigned to the affected object yet, it is possible to define default settings for the basic data layout for documents, materials and change numbers.

The following preference variables are used for this purpose:

`obr.propertypane.doc.default`

`obr.propertypane.mat.default`

`obr.propertypane.ecm.default`

`obr.propertypane.omr.default`

### Definition of the Default Layout for Documents

The basic data to be displayed is specified as a semicolon-separated list:

`obr.propertypane.doc.default = 15;16;19;24;6;7;9;8;12;30;31;28;32;17;77.7`

The following data is available:

Identifier	Name
1	Document number
2	Document type
3	Document part
4	Version
6	Status
24	Status (short form)
7	Editor
9	Laboratory/office
12	Change number
13	Revision
14	Last SAP error message
15	DType
16	DType description
17	Last saved
18.2	Master
18.3	Family
18.4	Dirty
19	CAD system ID
28	Material number
25	Assembly

Identifier	Name
27	Application type
77.7	Number of save transactions
30	Valid from
31	Valid to
32	Creation date
ISDRAWING	Is drawing
CHAR%MODELTYPE	CHAR%MODELTYPE
CHAR%ERZEUGERSYSTEM	Generated by system
CHAR%ZEIGE_RES4	Reserve field DRAW

### Definition of the Default Layout for Materials:

The basic data to be displayed is specified as a semicolon-separated list.

`obr.propertypane.mat.default = MARA-MTART;MARA-MSTAE;MARA-MBRSH;MARA-SPART;MARA-MEINS;MARA-MEABM;MARA-BSTME;MARA-MATKL;MARA-DISST;MARA-BRGEW;MARA-LABOR;MARA-ZZCADSYSTEM;MARA-BEGRU`

### The following data is available

Identifier	Name
SORT	Sorting
NUMBER	Material
SAPINFO	Last SAP error message
NTGEW	Net weight
MBRSH	Industry
BFLME	Logistics variants
MAKTX	Material short text
DISST	MRP level
BEGRU	Authorization group
BSTME	Order quantity
MTART	Material type
MSTAE	Cross-plant material status

Identifier	Name
MEABM	Measurement unit
MEINS	Base unit of measure
LABOR	Laboratory/office
KZKUP	Co-product
SPART	Division
INHBR	Gross content
MATKL	Goods group
AESZN	Change number of the document
ZEIFO	DIN format
/DSD/VC_GROUP	DSD grouping
ZZCADSYSTEM	CAD system
WRKST	Basic material
LVORM	LV client level
BREIT	Width
CADKZ	CAD ind.
FORMT	DIN format
XCHPF	Batch management requirement
BRGEW	Gross weight

### Definition of the Default Layout for Change Numbers:

The basic data to be displayed is specified as a semicolon-separated list.

```
obr.propertypane.ecm.default =
reasonChn;FUNCTION;createdBy;authGroup:4;createdOn;changedBy;
changedOn;validFrom;LONGTEXT;STATUS;TECH_REL
```

### The following data is available

Identifier	Name
SORT	Sorting
CHANGENUMBER	Number
description	Description

Identifier	Name
STATUS	Status
authGroup	Authorization group
validFrom	Valid from
reasonChn	Change reason
createdOn	Creation date
createdBy	Created by
changedOn	Changed on
changedBy	Changed by
FUNCTION	Function
TECH_REL	Technically free
SAPINFO	Last SAP error message
LONGTEXT	Long text

### Definition of the Default Layout for Object Management Records:

The basic data to be displayed is specified as a semicolon-separated list.

```
obr.propertypane.omr.default =
DOC_NUMBER;DOC_TYPE;DOC_PART;DOC_VERS;DOC_DESC;DOC_STAT;
DESCR_OBJ;DATUV;ALT_DATE;OBJ_CHGLOCK;LTXSP
```

The basic data to be displayed is specified as a semicolon-separated list.

### *The following data is available*

Identifier	Name
SORT	Sorting
ALT_DATE	Alternative date
DATUV	Valid from
LTXSP	Language of the long text
DESCR_OBJ	Object management record description
OBJ_CHGLOCK	Locked for changes
DOC_NUMBER	Document
DOC_TYPE	Type

Identifier	Name
DOC_PART	Document part
DOC_VERS	Document version
DOC_STAT	Status description
DOC_DESC	Description

## 7.5 UI theme

You can set the colors of the UI theme back to versions 5.2.5.4 and earlier.

Set the following preference variable in `default.txt`:

```
plm.ui.supported.themes = com.sun.java.swing.plaf.windows.WindowsLookAndFeel
```

## 8 /DSCSAG/ BUSINESS ADD-INS

SAP ECTR for SAP S/4HANA provides these BAdI methods.

### 8.1 ADVOBJ

BAdI method	Description
ADV_OBJ_GET_F4_AFTER	After pressing <b>F4</b> on certain fields

### 8.2 CHANGE

BAdI method	Description
ECM RFC_CREATE_BEFORE	Prior to calling transaction CC01, DSCSAG/ECM RFC_CREATE
ECM_STATUS_SET_SINGLE_AFTER	Call at the end of /DSCSAG/ECM_STATUS_SET_SINGLE
ECM_CREATE_BEFORE	Before generating an ECM.
ECM_CREATE_AFTER	Before generating an ECM.
ECM_CREATE_OMR_BEFORE_MAINT	Call before CCAP_ECN_MAINTAIN

### 8.3 CLASS

BAdI method	Description
CHAR_VALUE_VERIFY_BEFORE	Call at the start of /DSCSAG/CHAR_VALUE_VERIFY
CHAR_VALUE_GET_F4_VALUES	Call at the start of /DSCSAG/CLASS_GETDETAIL
CHAR_VALUE_GET_F4_VALUES	Call at the end of /DSCSAG/CHARACT_GET_CHECKTABLE
CLASS_GETDETAIL_AFTER	Call at the end of /DSCSAG/CLASS_GETDETAIL
CHECKTABLE_GET_BEFORE	Call at the start of /DSCSAG/CHARACT_GET_CHECKTABLE
CLASS_SELECT_OBJECTS_BEFORE	Before execution of /DSCSAG/CLASS_SELECT_OBJECTS
CHECK_FOR_OBSERVER_EXIT	Called by EXIT_SAPLCLFM_002
OBJ_CLASSIF_SET_SINGLE_AFTER	Call in /DSCSAG/OBJ_CLASSIF_SET_SINGLE
CLASS_ALLOC_MAINT_BEFORE	Call in /DSCSAG/CLASS_ALLOC_MAINT_MUL2

### 8.4 /DSCSAG/DOCUMENT\_01

BAdI for function group /DSCSAG/DOCUMENT

BAdI method	Description
DOC_CLONE_CLASSIF_BEFORE	Document is cloned prior to classification
WHEREUSED_AFTER	After the material consumption document structure
WHEREUSED_BEFORE	Before the material consumption API is called
DOC_AFTER_GET_CLASS_SINGLE	After reading the <code>classdata</code> of a document
DOC_RFC_CHANGE_AFTER	After changing the document with RFC GUI
DOC_RFC_CHANGE_BEFORE	Before changing the document with RFC GUI
DOC_BEFORE_CLONE	Prior to generating the document with <code>/DSCSAG/DOC_CLONE</code>
DOC_BEFORE_CLONE_LOOP	Before the loop in <code>/DSCSAG/DOC_CLONE</code>
DOC_AFTER_CLONE	After <code>/DSCSAG/DOC_CLONE</code>
DOCSTR_AFTER_MAINTAIN	After maintaining the document structure
DOC_BEFORE_DELETE_HARD	Before starting the deletion transaction
DOC_BEFORE_DELETE_HARD_INT	Before calling <code>document delete hard</code> (API)
DOC_AFTER_DELETE_HARD_INT	After calling <code>document delete hard</code> (API)
DOCVERSION_GET_FIRST	Determining the first version
DOC_BEFORE_CHANGE	Before calling <code>BAPI_DOCUMENT_CHANGE2</code>
DOC_AFTER_CHANGE	After calling <code>BAPI_DOCUMENT_CHANGE2</code>
DOC_BEFORE_GET_LATEST	Before the current version of a document is specified
DOC_AFTER_CHECKIN_MULTI	After execution of the check-in of multiple documents
DOC_PLOT_START	Start plot of documents
DOCSTR_BEFORE_MAINTAIN	Before the document structure is maintained
DOC_CONV_CHECKIN_BEFORE	Before saving a document (conversion server API)
DOC_CONV_CHECKIN_AFTER	After saving a document (conversion server API)
DOC_CONV_CHANGE_BEFORE	Before changing a document (conversion server API)
DOC_CONV_CHANGE_AFTER	After changing a document (conversion server API)

## 8.5 /DSCSAG/DOCUMENT\_02

BAdI for function group `/DSCSAG/DOCUMENT`

BAdI method	Description
TRANSLATE_TO_CODEPAGE	Translation to code page
DOC_CHECKIN_MULTI_BEFORE_CV	Before executing CVAPI_DOC_CHECKIN
DOC_CREATE_FROM_SRC_BEFORE	Before creating a document with /DSCSAG/DOC_CREATE_FROM_SRC2
DOC_CREATE_AFTER	After creating a document with /DSCSAG/DOC_CREATE2
DOC_CREATE_BEFORE	Before creating a document with /DSCSAG/DOC_CREATE2
DOC_CHECKOUT_MULTI_AFTER	After execution of the check-in of multiple documents
DOC_CHANGE_AFTER	After calling BAPI_DOCUMENT_CHANGE2
DOC_CHANGE_BEFORE	Before calling BAPI_DOCUMENT_CHANGE2
DOC_CHECKOUT_MULTI_BEFORE	Before executing the check-out of multiple documents
DOC_GETDETAIL_AFTER	After receiving the details
DOC_CREATE_FROM_SRC_AFTER	After creating a document with /DSCSAG/DOC_CREATE_FROM_SRC2
DOC_VERSION_BUFFER	Changes the current version (and others) prior to saving
DOC_GET_LATEST_STR_BEFORE	Before doc.structure explosion is perf. (get_latest_str3)
DOC_GET_LATEST_STR_AFTER	After doc.structure explosion is perf. (get_latest_str3)
DOC_CANCEL_MULTI_BEFORE	Before executing /DSCSAG/DOC_CANCEL_MULTI2
DOC_CANCEL_MULTI_AFTER	After executing /DSCSAG/DOC_CANCEL_MULTI2
DOC_GETLIST2_BEFORE	Before executing /DSCSAG/DOC_GETLIST2
DOC_GETLIST2_AFTER	After executing /DSCSAG/DOC_GETLIST2
DOC_VERSION_BUFFER_BEFORE	Before generating a new entry in table /DSCSAG/DOC_VERS
DOC_GET_VERSION_FROM_BUFFER	Determine the version of the buffer
DOC_LONGTEXT_SAVE_BEFORE	Before saving a long text
DOC_LONGTEXT_SAVE_AFTER	After saving a long text
DOC_CHECK_AUTH_AT_EXPLODE	BAdI for the optional execution of an authorization check
DOC_CHANGE_MULTI3_BEFORE	Before calling multi3 document change
DOC_CHANGE_MULTI3_AFTER	After calling multi3 document change
CHANGE_SEARCH_ORDER	Changing the search folder
SELECT_LEVEL_DATA_BEFORE	Before the call of the BAdI for the top assembly (select_level_data)

BAdI method	Description
SELECT_LEVEL_DATA_AFTER_LATEST	After the BAdI for select_level_data after GET_LATES_A1

## 8.6 /DSCSAG/FOLDER\_01

BAdIs for function group /DSCSAG/FOLDER

BAdI method	Description
FOL_OPENDETAILS_DOC	While reading the document and open details
FOL_OPENDETAILS_AFTER	After FOL_OPENDETAILS
FOL_OPENDETAILS_BEFORE	Before executing FOL_OPENDETAILS
FOL_ALLOC_MAINT_BEFORE	Before executing /DSCSAG/FOL_ALLOC_MAINTAIN2
FOL_ALLOC_MAINT_AFTER	After executing /DSCSAG/FOL_ALLOC_MAINTAIN2
FOL3_OPEN_BEFORE	Before executing /DSCSAG/FOL3_OPEN
FOL3_OPEN_AFTER	After executing /DSCSAG/FOL3_OPEN
FOL_CLN_OBJ_ASSIGN_BEFORE	Before assigning objects to a new folder and during the cloning transaction
FOL_CLN_DOC_KEY_CREATE_BEFORE	Before executing create Document Key

## 8.7 /DSCSAG/MATERIAL\_01

BAdIs for function group /DSCSAG/MATERIAL

BAdI method	Description
MAT_REV_CREATE_BEFORE_INT	Before creating a revision for a material
MATBOM RFC_CHANGE_BEFORE	Before changing the Bill of Material by RFC
MATBOM RFC_CREATE_BEFORE	Before creating the Bill of Material by RFC
MATBOM_AFTER_MAINTAIN	After maintaining the Bill of Material
MATBOM_BEFORE_MAINTAIN	Before maintaining the Bill of Material
MAT_BEFORE_CREATE RFC	Creation of a new material by RFC
MAT_AFTER_GETDETAIL	After reading the material details
MATBOM_BEFORE_READ	Before reading a Bill of Material
MATBOM_AFTER_READ	After reading a Bill of Material
MATBOM_AFTER_DELETE	After deleting a Bill of Material

BAdI method	Description
MATBOM_BEFORE_DELETE	Before deleting a Bill of Material
MATBOM RFC_CHANGE_AFTER	After changing the Bill of Material by RFC
MATBOM RFC_CREATE_AFTER	After creating the Bill of Material by RFC
MAT_BEFORE_DISPLAY RFC	Creation of a new material by RFC
MAT_SAVE_BEFORE	Before executing save material
MAT_SAVE_AFTER	After executing save material
MAT_CHANGE RFC_BEFORE	Before executing change a material via RFC
MAT_CHANGE RFC_AFTER	After executing change a material via RFC
MAT_AFTER_CREATE RFC	After executing the creation of a new material via RFC
MAT_GETLIST_BEFORE	Before executing the material search
MAT_SETALL_BEFORE	Before executing ADV_MAT_SETALL
MAT_SETALL_AFTER	After executing ADV_MAT_SETALL
MAT_REV_CREATE_BEFORE_MULTI	Before executing /DSCSAG/MAT_REV_CREATE_MULTI
MAT_COPY_BEFORE_INT	Before calling BAPI /DSCSAG/ADV_MAT_COPY
MAT_WHERE_USED_AFTER	After calling MAT_WHERE_USED

## 8.8 /DSCSAG/MM\_01

BADIs for function group /DSCSAG/MASTERMODELL

BAdI method	Description
MM_NEWVERSION_CHECKDIRTY	Before checking the dirty flag of a dependent document
MM_ECM_CREATE_OMR_BEFORE	Before the ECM generation for a master model group
MM_COPYAS_AFTER2	After master model Copy as
MM_NEWVERSION_AFTER	After master model new version
MM_SETMASTER	Setting up the document structure and object links for NM in an MM
MM_COPYAS_BEFORE2	Before master model Copy as
MM_COPYAS_NM_BEFORE2	Before the creation of the dependent document in Copy as
MM_GETNONMASTER2_BEFORE	@ECTR: before get nonmasters

## 8.9 /DSCSAG/PLM\_01

BAdIs for function group /DSCSAG/PLM

BAdI method	Description
CHECK_LOGON_DATA_BEFORE	Before logon data check
OBJ_PERFORM_SELECTION	Execution of actions on objects
CHECK_ADMIN	Check whether the user is a PLM administrator
INIT_BEFORE	Before the initialization
PLM_WRITEMESSAGE_BEFORE	Before creating the log file

## 8.10 /DSCSAG/UTILS

BAdI definition for /DSCSAG/UTILS FMs

BAdI method	Description
BEFORE_BASKET_SELECT	
BEFORE_BASKET_SET	
F4_VALUES_GET_BEFORE	
OBJ_LONGTEXT_SAVE_BEFORE	Before save Longtext at object.
OBJ_LONGTEXT_SAVE_AFTER	After save Longtext at object.

Enhancement spot /DSCSAG/PLM\_DOCUMENT (package /DSCSAG/PLM++\_Modules)

### 8.10.1 /DSCSAG/DOC\_CHECK\_AUTH

BAdI definition for checks

BAdI method	Description
DOC_CHECK_SINGLE	Checking an individual document

### 8.10.2 /DSCSAG/DOC\_SEARCH

BAdI definition for FG /DSCSAG/DOC\_SEARCH

BAdI method	Description
DOKNR_RANGE_CONVERSION_EXIT	Document number

## 8.11 Enhancement Spot /DSCSAG/PLM\_ACC

BAdI definition for /DSCSAG/PLM\_ACC\_ASSIGN

<b>BAdI method</b>	<b>Description</b>
ACC_ASSIGN_BEFORE	Before assign ACC
ACC_ASSIGN_AFTER	After assign ACC

### 8.12 /DSCSAG/PLM\_ACC\_CNTX

BAdI definition for Create/Change Contexts

<b>BAdI method</b>	<b>Description</b>
ACC_CNTX_CREATE_BEFORE	Before create ACC context
ACC_CNTX_CREATE_AFTER	After create ACC context

**9.1 DRAW**

This table contains the default entries for documents. All basic data of a document is stored here. To get more detailed information on the individual fields, use transaction SE11 to access table DRAW.

**9.1.1 Data Model**

The following sections explain some data fields that describe a document in more detail. Field RES4 from table DRAW and additional fields from table CDESK-DRAW describe the document, the sort field on the document structure item and the relations between two documents.

**9.1.2 Res4 Field in Table DRAW**

The object type of the document (BAPI structure fields DRAW-RES4 or BAPI\_DOC\_DRAW2-USERDEFINED4) is stored in this field. The object type has controlling functions and specifies with which CAD application the document is opened/edited.

The first three characters define the CAD system; these are followed by a ":" The following abbreviations for CAD systems are currently defined:

Abbreviation	Description
PRO	Pro/Engineer
INV	Autodesk Inventor
CAT	CATIA
SLW	SolidWorks
SLE	Solid Edge
UGS	NX
IDE	I-deas
MED	MEDUSA
ME1	ME10
ACD	AutoCAD
ACM	AutoCAD Mechanical
ECS	ECSCAD (AutoCAD-based)
ACS	AutoCAD Speedikon-X
MDT	Mechanical Desktop
MST	Microstation

The abbreviations for object types are controlled centrally by SAP. This is where you find the combinations of letters that are currently defined:

Abbreviation	Description
A	Assembly
P	Part
D	Drawing
PG	Part Generic (gen. part; contains "family table" with "variant table")
AG	Assembly Generic (gen. part; contains "family table" with "variant table")
PI	Part Instance (part fam. member, generated from the part family)
AI	Assembly Instance (part fam. member, generated from the part family)
L	Layout (has a controlling function; is used to group several files)
N	Standard part
X	Exchange assembly
W	No material required ("Without material")
U	Do not take into account subordinate documents for Bill of Material

## 9.2 CDESK\_DRAW

Table `CDESK_DRAW` contains the interface-specific information. It is also referred to as *enhanced DRAW table* because it enhances the DRAW table with some new additional fields. The interface uses the table and creates additional information and other data redundantly. Hence, information does not have to be determined anew every time. This improves performance.

### 9.2.1 Important Fields in Table CDESK\_DRAW

Stored data is saved in various fields of SAP table `CDESK_DRAW`:

- SUBTYPE
- MASTER
- FAMILY
- DIRTY
- CHECKIN\_DATE.

The following table illustrates the differences between the two versions:

	DRAW	CDESK DRAW
DType		SUBTYPE (1-5 chars)
Application Type	RES4 (1-5 chars)	
Master flag	RES3 (M / N)	MASTER (M / N)
Family flag	RES3 (NV, V)	FAMILY (NV / V / N)
Dirty flag	RES3 (Y)	DIRTY (" " / X)

	DRAW	CDESK DRAW
Save-time	RES2 (date / time)	CHECKIN_DATE
Save-count		RES5 (20 chars)
Document unit		FLAG2 (I=inch / M=metrical)
Free for customer		FLAG1
Free for customer		FLAG3
Last modified date 2		RES6
Reserved for DSC		RES7
Free for customer		RES8

Here, an important field is the save-counter, which is stored in field RES5 of table CDESK\_DRAW. For every check-in operation of a document, the save-counter is increased by one and used for up-to-date checks of the originals.

### 9.2.2 Additional Fields in CDESK\_DRAW

These fields are filled automatically via BAdIs document\_main01-before\_save or bom\_update.

Field in CDESK_DRAW	Description	Remark
MAT_AVAILABLE	has material link	Duplicated to MAT_AVAILABLE in table /DSCSAG/DOC_RED
NM_AVAILABLE	Dependent document exists	Duplicated to NM_AVAILABLE in table /DSCSAG/DOC_RED
RES6	Last modified date 2	

### 9.2.3 Fields (Performance Fields) in /DSCSAG/DOC\_RED

This table is introduced to store additional redundant data from the SAP system to ensure the respective information can be accessed quickly.

Field name	Type	Description	Value	Remark
DOKAR	CHAR 3	Document type		Key field
DOKNR	CHAR 25	Document number		Key field
DOKVR	CHAR 2	Document version		Key field

Field name	Type	Description	Value	Remark
DOKTL	CHAR 3	Document part		Key field
CHECKOUTFILESIZE	NUMC 12	Check-out file size (bytes)		
MASTERFILENAME	CHAR 255	File name master original		
NAMEVALUE1	CHAR 255	Name/value pairs		
MASTERPHIO_ID	CHAR 32	PHIO of the master original		
ASSEMBLY	CHAR 1	Flag: Assembly or Part	A, P	
GENERIC	CHAR 1	Flag: Generic Part or Instance Part	G, I	
VALIDFROM_RED	DATS 8	"Valid from" date		
REVLEVEL_RED	CHAR 2	Revision Level		
MASTER	CHAR 1	Flag: Master/dependent document	M, N	
FAMILY	CHAR 4	Family member		
MAT_AVAILABLE	CHAR 1	Flag: Material assigned	' ', X, O	Duplicate of MAT_AVAILABLE in table CDESK_DRAW. O = assignment to non-existent material
NM_AVAILABLE	CHAR 1	Check box: dependent document assigned	' ', X	Duplicate of NM_AVAILABLE in table CDESK_DRAW.
STORAGE	CHAR 31	Storage Category		
REFERENCE	CHAR 1	Flag: Has external references (wavelink)	' ', R	
MOE	CHAR 1	Flag: Original Exists	0, 1	
MOC	CHAR 1	Flag: Original Checked out	0, 1	
AOE	CHAR 1	Flag: Additional Original Exists	0, 1	
AOC	CHAR 1	Flag: Additional Original Checked out	0, 1	
A2OE	CHAR 1	Flag: Has 2D Preview	0, 1	

Field name	Type	Description	Value	Remark
A3OE	CHAR 1	Flag: Has 3D Preview	0, 1	
NM_OLD	CHAR 1	Check box: dependent document is older than master	' ', X	

#### 9.2.4 Relationship Types/Sort Field in Document BOM Item

The relation type is stored in the sort field of the document structure (i.e. `STPO-SORTSTRING` or `BAPI_DOC_STRUCTURE-SORTSTRING`) and has control functions. The first three characters define the CAD system; these are followed by a ":".

The abbreviations for relation types are controlled centrally by the SAP system. Here are the letters that are currently defined:

Relation type	Description
Y	Not required external references/inter-part links (part – Wave link source, part- Inter-part reference, part- User-defined feature)
R	Reference (for example in an exchange assembly)
C	Component relation (part - assembly or sub-assembly - assembly)
I	Member relation (member - part family)
D	Dependent document relation (master model – drawing, master model– alt rep) The document property <code>dependent document</code> is exclusively determined by the existence of relation type "D" in the sort field of the reference to the master document. Master documents are documents that do not have a "D" reference to another document. The master/dependent document flag in field <code>MASTER</code> of <code>CDESK_DRAW</code> is therefore not primary information but information derived from relation type "D" which is designed to make processing simpler for example when determining icons. References with relation type "D" are always resolved <code>asSaved</code> , irrespective of the set explosion rule. This ensures that the right versions of the master documents and dependent documents are made available.
S	Suppressed (component is suppressed in assembly)
M	Master ("leading" object in a drawing)
L	Layout relation (layout at a controlled object)
W	No material required ("Without material")
U	Do not take into account subordinate documents for Bill of Material

The report consists of three sections:

- Document data

You can change certain documents.

- SORTF settings

Both characters that are replaced are entered here. SORTF alt specifies which character is currently set. SORTF new states which character is set. The fields SORTF new is fetched from setting PLM.DOCSTRUCTURE.SORTFFIELD.REFERENCECHAR.

Determination of SORTF old (alt):

If new=R then old=Y, if new=Y then old=R.

- Options

Change STPO: Table STPO is edited.

Change Extended DRAW: The entries in FLAG1 of CDESK\_DRAW are changed.

Test mode: If you set this checkmark, the changes are not written to the database (no COMMIT).

Package size: This setting states how large the processing packages are. To avoid memory problems, processing is split in packages.

### 9.2.5 Folder Contents in the SAP ECTR for SAP S/4HANA Desk in /DSCSAG/FOL\_ALC

The folders used in SAP ECTR for SAP S/4HANA are documents. In SAP ECTR for SAP S/4HANA, folders can also contain other objects. The keys of these objects are stored in SAP table /DSCSAG/FOL\_ALC. The table is structured as follows:

Filed name	Type	Description	Remark
TOPFL	CHAR 50	Object key of the folder in which the object is located.	Key field
DOKOB	CHAR 50	SAP object type	Key field
OBJKY	CHAR 50	Object key	Key field

Supported SAP object types and structure of the object key:

DOKOB	OBJKEY	Object Description	Remark
Mara	[Internal material number]	Material	
AENR	[Change number]	Change number	
CRHD	Combination of: [Plant][Work center number]	SAP work center	
IFLOT	[Number techn. location]	Technical location	
PLAF	[Plan order number]	Plan order	
PLKO	Combination of: [Plan type][Plan group][Plan group counter]	Task list	
PORDER	[Production order number]	Production order	
KNA1	[Customer number]	Customer	
LFA1	[Supplier number]	Supplier	
CRVS_B	[Production resource/tool number]	Production resource/tool material	

DOKOB	OBJKEY	Object Description	Remark
EQUI	[Equipment number]	Equipment	
VBAK	[Sales order number]	Sales order	
VBAP	Combination of: [Sales order number][position number]	Sales order item	
PRPS	[Internal number of the WBS element]	WBS element	
PROJ	[Internal project number]	Project definition	
WORKITEM	[Work item ID]	Work item	
SOFM	Combination of: [FOLTYP][FOLYR][FOLNO][Object type] [Object number]	SAP mail	

### 9.3 /DSCSAG/PREF

The table contains settings that are required for the interface to work on the ABAP side. The table consists of the two rows `Name` and `Value`. The `Name` column contains the settings name and the `Value` column contains the value of the setting.

### 9.4 /DSCSAG/CHECK

Table `/DSCSAG/CHECK` contains information about which module is active and which is inactive. Settings in this table control the response of the interface and activate or deactivate entire functions. To activate a module for all users, it must be activated for the `DEFAULT` user. Each module can be activated individually for a certain user. If a module has been activated for a certain user, all previous default values are invalid for this user and must be set explicitly for the user.

### 9.5 /DSCSAG/DOC\_RED\_UPDATE

The framework is enhanced with SAP table `/DSCSAG/DOC_RED`. This table is an enhancement of table `CDESK_DRAW`. Since some data from this table is used in the framework, this table must be filled once after importing the transport.

Report `/DSCSAG/DOC_RED_UPDATE` is made available for filling the table.

The report has two modes:

- [Automatic mode](#)
- [Manual mode](#)

In automatic mode, the corresponding data records for all documents included in `CDESK_DRAW` are created in table `/DSCSAG/DOC_RED`.

In manual mode, the entries in table `/DSCSAG/DOC_RED` can be created for a selected set of documents.

### Application Parameters

- `Test (w/o db changes)`  
No changes in the database. This option can be used for analysis purposes.
- `Clear all redundant data`  
If this checkmark is set, the data records in the table are emptied. The entries in the table exist; these entries contain no data. If this parameter is not set, an update is performed in a second run.

- `Package size`  
The number of the documents that can be sent to the update program at once. We recommend a number between 100 and 1000. The number depends on the efficiency of the update program.
- `Determine 'Valid from' date`  
If this checkmark is set, the data for the "Valid from" date is determined and copied to the table.

## Possible Message Output

- `No data found. Table: <Table name>`  
The table contains no data.
- `The master WSAPPL to DTYPE <DTYPE> of DIR <document key> cannot be determined`  
The workstation application of the master original for the specified DType could not be determined for the document. Possible causes:
  - No workstation application has been defined for this DType.
  - The settings have not been uploaded to the SAP.
  - The DType does not have a fixed master original.
- `Error during saving in DB`  
An error occurred during saving in the database. The error is a database problem.
- `Data successfully saved`  
Data has been successfully saved in the database.
- `The DTYPE for DIR <document key> cannot be determined`  
The DType cannot be determined for the document. Possible cause: The subtype is not set in table CDESK\_DRAW.

If no DType for the respective document is found in CDESK\_DRAW the preference variable is taken into account:

```
plm.control.empty.dtype.<document type> = <DType>
```

By default, these preference variables are uploaded using the Administrator function [Upload Settings to SAP](#).

### 9.5.1 /DSCSAG/PLM\_DOC\_VER\_BUFFER\_FIL

#### Logon Parameters

- `Write log`  
Specifies where the log is written. The log is written to the standard output/SPOOL.
- `Maximum number of entries`  
The first `n` entries found are processed; `n` is the value entered.
- `Package size`  
Number of updates that can be sent to the update program at once.
- `Separator used`  
This field is left empty.

**10.1 Bases for Optimization in WAN Installations**

**10.1.1 Measures for Optimization**

The following measures help optimize performance:

- Minimization of API calls, use of batch APIs
- Avoidance of SAP modules that execute client queries
- Use of transactions that run in the background whenever possible.
- Use of storage for JCO structures

**10.1.2 SAP API Calls**

For SAP communication in wide area networks, there are two basic influencing factors:

**Latency (RTT)**

The value specified in the `Latency` or `round trip time` (RTT) specifies how long a package takes to travel from the client to the server and back. Typical values for `Latency` are:

<b>LAN</b>	<b>&lt; 1ms</b>
WAN (Germany)	~ 20ms - 30ms
WAN (USA)	~ 200ms - 250ms
WAN (China)	~ 300ms - 350ms

The easiest way to determine the latency values is by means of a ping from the client to the servers.



**Example**

```
ping -l 50 saglin3
```

You can use transaction OS01 to execute pings from the application server to the client. This can also be used to test the latency to the database server.

**Transfer Rate**

The transfer rate refers to the quantity-based data throughput within a WAN connection. This is measured by transferring a (known) larger quantity of data and measuring the execution time required to do so. Data that cannot be compressed is used to determine the throughput. Compression can falsify results. The TCP protocol is used to transfer data in packages of a specified size (Windows size). Since basically every package has to be confirmed, the window size and latency play an important role here.

The time for the execution of an SAP API program call in a WAN environment is composed as follows:

$$T_{\text{aus}} = T_{\text{sap}} + T_{\text{rfc}} + T_{\text{data}}$$

The individual values are explained in more detail:

## Taus

Execution time of an API call in a WAN environment

## Tsap

Processing time in SAP ERP. Since database operations are performed, the execution time does not depend on the latency. If you notice this, a call-back to a remote RFC destination is performed somewhere in SAP ERP. We have detected calls to the client (program calls, query of the frontend type) and to content servers (status queries for 4.6c).

## Trfc

The call of an API in a WAN situation is always longer by exactly this latency. So if an API in the LAN takes 10ms to execute, it will take 310ms in the WAN (China). At SAP, there is dependency on the type of logon. For transactions in the foreground (GUI), the RFC layer obviously queries information from the client.

For logons in the foreground, the following applies:

- $Trfc = 3 \times Latency$

For logons in the background, the following applies:

- $Trfc = Latency$ .

## Tdata

The transfer time for the data is calculated as the quotient of the amount of data and the transfer time. The amount of data is calculated from the width of a record and the number of records. So if you transfer 1000 records of 1000 bytes, this amounts to 1 MByte of data.

Compress this data prior to the transfer from the RFC subsystem. Since these structures tend to be empty, a compression factor of 1:10 - 1:20 is realistic. That is, 1 megabyte turns into approximately 100 kilobytes. At a throughput of 50 KByte/sec this takes 2 seconds. If compression does not work, it takes 20 seconds. You can verify this in a customer installation in an RFC trace. If larger data structures contain readable character strings there, something is wrong.

### 10.1.3 Optimization of RFC Data Transfer

When transferring large amounts of data, it is very important to compress the data to be transferred.

#### Fast RFC Serialization

JCo 3.1 offers the option to transfer data more efficiently with "Fast RFC serialization," which is technically based on a column-based serialization approach. See [https://support.sap.com/content/dam/support/en\\_us/library/ssp/products/connectors/jco/jco\\_31\\_documentation.pdf](https://support.sap.com/content/dam/support/en_us/library/ssp/products/connectors/jco/jco_31_documentation.pdf) for more information.

The setting can be made in `SAPlogon.txt`.

## Define WAN Compression Threshold as Profile Parameter

See [SAP Note 373452](#) for more information.

### 10.1.4 Automatic Performance Analysis for ABAP Function Modules or Classes

For certain ABAP function modules or classes, performance data is collected specifically here. This data can then be analyzed and evaluated using SAP transaction SAT.

You use the following preference variable to specify for which APIs performance data is collected:

```
sap.bapi.runtimeAnalyzeApis = <apiname_to_analyze>
```



#### Example

```
sap.bapi.runtimeAnalyzeApis = /DSCSAG/DOC_CHECKIN_GET_URL;/DSCSAG/DOC_CREATE2
```

Here, the use of wildcards is also allowed at the end of the API names, for example:

```
sap.bapi.runtimeAnalyzeApis = /DSCSAG/FOL*/;/DSCSAG/DOC*/;DSCSAG/MM_NEWVER_MULTII2
```

## 10.2 Concrete Measures with SAP ECTR for SAP S/4HANA

### 10.2.1 Logging on to SAP in the Background

Each RFC transaction is assumed to take one latency for logons in the background and three times the latency for logons in the foreground. Try to use logons in the background when possible. To do this, preference variable `sap.logon.SapGui* = 0` is set. If called SAP modules call up dialogs, this can cause problems.

### 10.2.2 Preference Variable for Transactions in the Foreground

In the following preference variable, you can specify a list of modules that are called in a transaction in the foreground:

```
sap.bapi.useGuiTransaction=<MODULE_1>;<MODULE_2>;...
```

### 10.2.3 Replacing Function Modules

You can use this preference variable to replace a function module with another, for example a customer-specific function module:

```
sap.bapi.replace.<BAUSTEIN_NAME> = <Z_MODULE>
```

### 10.2.4 Using the URL Checkout

If you are using the URL checkout, the communication between the application server and the local SAPHTTP which takes place otherwise is deactivated. This also significantly speeds up required checkout transactions from local cache servers. To do this, the following variable must be set:

```
plm.doc.checkouturls = true
```

When the URL checkout is used, the known fallback strategies are not required (open via GUI channel). If the client cannot find the content server, it cannot be opened. This is normally due to network addresses that cannot be resolved by DNS. At the same time, `DMS_C1_ST` is not supported because web access is not possible here.

Using the following preference variables, you can specify how many downthreads can be used simultaneously:

```
plm.doc.checkout.UrlWorkerCount
```

The URL checkout can be prevented with the following preference variables. If this preference variable is integrated and set to `true`, it forces the checkout via BAPI:

```
plm.doc.checkout.original.forceUseBapi
```

The following preference variable can be used to overwrite the previously set preference variable:

```
plm.doc.checkouturls = true
```

### **10.2.5 Using the JCO Repository**

Before a new transaction is opened, JCO reads the related structure definition from SAP when an API is called for the first time. This takes place once and in the background, but it is time consuming in remote locations. In SAP ECTR for SAP S/4HANA, you can generate a JCO repository. In this process, the API structure definitions are read once and stored in a repository file. When the definitions from the repository are used, the system start time is greatly reduced.

The JCO repository is used via the following preference variable in the `system-default.txt` file:

```
system.sap.readRepositoryFromFile = true
```

The repository saves the structure definitions redundantly. When support packages, SAP Notes, or new DSC transports are imported, the structures in the SAP system can change. If the structures in the SAP system have changed, regenerate the repository.

The default storage path for the JCO repository is `%PLM_INSTDIR_CONF%/customize/aux-files`.

### **10.2.6 Folder Operations**

#### **Making the Text Field for Folder Names Editable**

The following preference variable controls whether the text field for the folder name can be edited or not:

```
plm.om.folder.editableName = true/false
```

### **10.2.7 Limiting the Automatic Detail Reading for Work Items in the SAP Office Inbox**

If the SAP Office window is active in SAP ECTR for SAP S/4HANA, the details on the work items are read when documents are updated. Depending on the content of SAP Office, this can take quite a long time (10 - 15 seconds).

You can use the following preference variable to limit this:

```
plm.sapoffice.autoupdate.maxMessageCount = xxxx
```

If the number of work items in SAP Office exceeds the value in the preference variable, the details are no longer read automatically. This only happens when the user opens SAP Office manually.

### **10.2.8 Preventing a Memory Overflow when Checking in Several Documents**

Checking in many documents at the same time can cause a memory overflow. That is why the following preference variable has been used to limit the maximum permissible number of documents to be saved at the same time to 30 by default:

```
plm.document.checkin.splitCount = 30
```

When this threshold is exceeded, the module responsible for the check-in transaction is called several times.

### **10.2.9 Testing the HTTP Connection**

1. Start SAP ECTR for SAP S/4HANA.
2. Open an SAP GUI from SAP ECTR for SAP S/4HANA.
3. Start transaction `SE38`.
4. Start program `RSHTTP05`.
5. The system displays OK four times.

If that is not the case, check the SAP GUI Workdir directory. This directory contains a log file you can use for analysis. If the program writes: `...please delete`, you delete the destination in transaction `SM59` and run the program again. SAP ECTR for SAP S/4HANA then creates the destination anew.

#### 10.2.10 Testing the FTP Connection

1. Start SAP ECTR for SAP S/4HANA.
2. Open an SAP GUI from SAP ECTR for SAP S/4HANA.
3. Start transaction `SE38`.
4. Start program `RSFTP005`.
5. The system displays `OK` four times.

If that is not the case, check the SAP GUI Workdir directory. It contains a log file that you can use for analysis purposes. If the program writes: `...please delete`, you delete the destination in transaction `SM59` and run the program again. SAP ECTR for SAP S/4HANA then creates the destination anew.

#### 10.2.11 Testing the Content Server Performance

1. Start SAP ECTR for SAP S/4HANA.
2. Open an SAP GUI from SAP ECTR for SAP S/4HANA.
3. Start transaction `SE38`.
4. Start program `RSHTTP80`
5. Enter the following values:  
Size 1 MB  
Test runs 10
6. The security and opening times are 400 max. If the times are higher, then improve content server performance

#### 10.2.12 SAP Content Server Cache Statistics

If the cache of the content server is full, this costs time and resources and can lead to timeouts. You can use transaction `CSADMIN` to check whether rolling takes place:

#### Transaction CSADMIN

1. Under `Server Selection`, choose the tab page `Server`.
2. In the input field `HTTP Script`, use `F4` to select the desired HTTP server.  
The `Statistics` tab page contains the required information.
3. Check the parameter `CacheDelete`, which contains the number of delete operations. An increase in the value of this parameter when updating shows that displacement takes place. The parameters `"CacheCurrentSize"` and `"CacheMaxSize"` are also located there. If the `"CacheCurrentSize"` is identical to the `"CacheMaxSize"`, the cache is almost full and will soon start displacement.

On the one hand, the size of the DB or the available file system determines the size of the cache. On the other hand, there is setting parameter `"CacheSizeMB,"` which determines the maximum size of the cache. Due to the blocking factor, however, the DB or available hard disk space should be approx. 30-50 % larger than the value set in `CacheSizeMB`. When you change the settings: Always restart the content server cache!  
Operating system command (DOS box, CMD shell): `iisreset`

### 10.3 Improving the Database Accesses

Update the statistics for these tables with `db20`.

### 10.3.1 Table DMS\_PHRE\_CD1

Set up the following two indexes on the table:

#### Z1 - index with these fields

- mandt
- re-class
- repl\_class
- rep2\_class
- rep2\_id

#### Z2 - index with these fields

- mandt
- reio\_id
- re\_class

### 10.3.2 Table DMS\_LODESC\_CD1

Set up index on the z1 index with this field:

- LOIO\_ID
- due to access from program SAPLSDCL.

### 10.3.3 Regular Updates of the Optimizer Statistics

Manually

- Regularly check the DB statistics of the following tables using transaction DB20:

```
BALDAT; BALHDR; DMS_DOC_FILES; DMS_DOC2LOIO; DMS_LO_CD1; DMS_PH_CD1; DMS_PH_PROP_CD1;
DMS_PHF_CD1; DMS_PHIO2FILE; DMS_LODESC_CD1; DRAW; DRAP; DRAD; DRAT; DOST; DRAZ;
DRAW_LAST_CHANGE; STXH; STPO; STKO; STXH; STXL; STZU; STAS; INOB; KLAH; KLAT; KSML;
KSSK; AUSEP; CDESK_DRAW; /DSCSAG/DOC_VERS; /DSCSAG/BBOX; /DSCSAG/DOC_RED;
/DSCSAG/EXPORT; /DSCSAG/FOL_ALC; /DSCSAG/IMPORT; /DSCSAG/RIP; /DSCSAG/STRUCT1;
/DSCSAG/UDF_CNT; /DSCSAG/UDF_LAY
```

- Use transaction DB14 to check whether an update run has been performed.

Automatically

- If you want to schedule regular updates, use transaction DB13.
- You can use transaction DB21 to check whether the above tables are taken into account in the scheduled updates. Field PROBE should contain no less than P30. P3 is too little.

### 10.3.4 Updating the Class Statistics

To update the class statistics or generate them for the first time, use transaction CLST, which can improve performance enormously when there are many classified objects.

- Note 74693 - Class System Object Search General Performance.
- Note 917987 - General Performance Variant Configuration

### 10.3.5 Max DB

To avoid performance problems, install patch 7.6.05 immediately.

Ensure that DB Analyzer is running. DB Analyzer automatically collects important performance data

Check the DB parameters using Note #0814704 - MaxDB Version 7.6 Parameter Settings for OLTP/BW. This can also be done semi-automatically using the DB Analyzer configuration file from Note: #1111426 - Parameter Check for liveCache/MaxDB Instances.

## 10.4 Deleting BOMs with Transformation Matrices

### Symptom

Long runtime when deleting document BOMs if transformation matrices exist for items.

### Cause:

Missing index on table DMUPOB.

### Solution

Use transaction SE11 to create the new index DMUPOB~T for table DMUPOB:

- Change database table DMUPOB:  
In the menu, choose 'GoTo--> Indexes ...' or directly click the button 'Indexes ...'.
- In the dialog 'Indexes for table DMUPOB' choose the button 'Create - Create Index'
  - Index identification: T
  - Short description: Use of transformation matrix; Non-Unique Index on all database systems
  - Index fields: MANDT; TMXGUID
- Save and activate the index
- Update the database statistics in table DMUPOB as described in note 429146.

## 10.5 Long Wait Time when a Document BOM Cannot be Written

### Symptom

Document structure cannot be written in first attempt.

### Cause

Example from API log:

```
#185 /DSCSAG/DOC_BOM_DELETE
DOCUMENTNUMBER      | 0000000000000000010781338
DOCUMENTPART        | 001
DOCUMENTTYPE         | UGD
DOCUMENTVERSION     | 01

#186 /DSCSAG/DOC_BOM_MAINTAIN2
| E      | 29 | 160      | BOM 0000000000000000010781338/UGD/001/01
is locked by SCHMIDT
```

If function module /DSCSAG/DOC\_BOM\_MAINTAIN2 returns this message, Java waits a certain time and then calls the function module again. This transaction is repeated several times if the error occurs again.

## Solution

The following preference variables can be set in the `default.txt` file.

Abbreviate the wait time and reduce the number of repetitions:

- `plm.docstructure.maintan.retryAtLock = <number>`  
Number of attempts if the BOM is locked after the deletion. Default value = 0
- `plm.docstructure.maintan.retryAtLockWaitTime = <time in seconds>`  
Wait time until the next attempt in seconds. Default value = 5 seconds

## 10.6 PLM Observer Sleep Mode

Unnecessary loads on SAP systems and the network by the PLM Observer can be avoided by putting the PLM Observer into sleep mode after a period of user inactivity.

To customize whether and when the PLM Observer goes into sleep mode, you can make the following adjustments in `default.txt`.

To enable sleep mode, set this preference variable to `true`.

- `plm.observer.sleep.mode.timeout.enabled = true`

Set this preference variable to the number of seconds of user inactivity allowed before sleep mode is activated. This value is set to 900 by default.

- `plm.observer.sleep.mode.timeout.in.seconds = 900`

## 11 ERROR ANALYSIS

### 11.1 Problem Analysis

To locate any issues that have arisen in the program flow more quickly, use an OSS message (PLM-ECC) to contact Support. Under [Options](#), ensure that the trace level is set to [debug](#) and the checkmark is set at [Write API Log File](#). You can call up the log file via the icon tray menu.

### 11.2 Information on the Error

To ensure you get help solving your problem as soon as possible, certain information is required. First, all relevant log files are required:

Log file	Description
Log file	(Logging the program run)
API logs	(Logging the SAP API calls)
NX-syslog	(By default, these files are located in directory %TEMP%\<ECTR>\logs9)

Set the trace level to [Debug](#) and activate API logging. You can set this via the options.

If you have received an error message, you can also summarize the error information directly via the displayed dialog. If you click [<Copy>](#), detailed error information is copied to the clipboard. This makes it easy to copy the error message into an e-mail. If you click [<Generate Report>](#), a file dialog opens. You can choose in which directory all required information ([API Logs](#) and [Log file](#)) are stored as a zip archive. You can forward this zip archive to the Service Center.

### 11.3 Additional Help with Troubleshooting

To locate and resolve errors, the SAP ECTR for SAP S/4HANA menu contains references to directories and functions for troubleshooting under "[Debugging](#)." The following sections describe the directories in more detail.

#### Directories

- **Log file directory**  
This is where the API log files are located. These log files log the access to the SAP system.
- **Communication directory**  
This directory contains files that are required for the flow of the file communication between SAP ECTR for SAP S/4HANA and external applications. These include command files, signal files and confirmation files. Furthermore, temporary Excel files that are generated in the [Object Browser](#) appear here.
- **Installation directory**  
This directory contains all files that are relevant for the local installation and configuration, including the executable program file (`dsc_StartPlm.exe`).
- **User working directory**  
This directory contains all files for documents that have been opened for viewing or editing in SAP ECTR for SAP S/4HANA. For each file, there is also an additional data directory, each with a di file and an XML file that contains the metadata of the document.
- **Additional data directory of the last active part**

This is where the additional data directory of the last active part is opened.

## Functions

- **Activate debug window script**  
Opens a document for outputting debug information in the text editor in parallel to the operation of SAP ECTR for SAP S/4HANA.
- **Activate command shell application start**  
When SAP ECTR for SAP S/4HANA is restarted, this opens a cmd.exe window.  
You can analyze the system environment in this window.
- **Activate log file script**  
Generates log files in the log file directory during the operation of SAP ECTR for SAP S/4HANA.  
Activate application start command editor
- **Opens a dialog in which the command for the application to be started for additional originals of a document can be changed.**
- **Trace log file**  
Displays the active log files in a console window.

**12.1 Additional Information**

This table provides links to additional information on the Operations Guide.

<b>Content</b>	<b>Quick Link to SAP Service Marketplace</b>
Master Guide, Installation Guide and Upgrade Guide	<a href="http://service.sap.com/instguides">http://service.sap.com/instguides</a> <a href="http://service.sap.com/ibc">http://service.sap.com/ibc</a>
Corresponding SAP Notes	<a href="http://service.sap.com/notes">http://service.sap.com/notes</a>
Released platforms	<a href="http://service.sap.com/platforms">http://service.sap.com/platforms</a>
Network security	<a href="http://service.sap.com/securityguide">http://service.sap.com/securityguide</a> <a href="http://service.sap.com/network">http://service.sap.com/network</a>
Technical infrastructure	<a href="http://service.sap.com/ti">http://service.sap.com/ti</a>
SAP Solution Manager	<a href="http://service.sap.com/solutionmanager">http://service.sap.com/solutionmanager</a>

[www.sap.com/contactsap](http://www.sap.com/contactsap)

© 2022 SAP SE or an SAP affiliate company. 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 SE or an SAP affiliate company.

The information contained herein may be changed without prior notice. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, and they should not be relied upon in making purchasing decisions.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies. See [www.sap.com/copyright](http://www.sap.com/copyright) for additional trademark information and notices.

**THE BEST RUN**

