



Configuration Guide | PUBLIC

Document Version: 1.0 – 2025-02-24

Configuration Guide for PLM System Integration 3.0 for SAP S/4HANA FP02

Content

- 1 About This Guide. 6**
- 2 Initial Configuration. 7**
 - 2.1 Mandatory Configuration Activities. 9
- 3 PLMSI Configuration. 11**
 - 3.1 Domain Model Code Lists. 11
 - Define Domain Model Change Type. 12
 - Define Domain Model Consumption Type. 12
 - Define Domain Model Criticality Code. 12
 - Define Domain Model Customer Order Type. 13
 - Define Domain Model Customer Order Item Product Structure Type. 13
 - Define Domain Model Document Type. 13
 - Define Domain Model Embedded Inspection Characteristics Type. 13
 - Define Domain Model Factory Type. 13
 - Define Domain Model Inspection Characteristics Type. 14
 - Define Domain Model Maturity Code. 14
 - Define Domain Model MIME Type. 14
 - Define Domain Model Product Type. 14
 - Define Domain Model Production Operation List Type. 14
 - Define Domain Model Relation Type. 15
 - Define Domain Model Semantic File Type. 15
 - Define Domain Model Status Code. 15
 - Define Domain Model Structure Type. 15
 - Define Domain Model View Structure Version Type. 15
 - Define Domain Model View Structure Version Edge Type. 16
 - Define Domain Model View Structure Version Node Type. 16
 - Define Domain Vendor Type. 16
 - Define Domain Model Work Area Type. 16
 - 3.2 Scenario and Version. 16
 - Define Scenario and Version. 17
 - Define Parallel Processing Settings for Inbound. 23
 - Define Scenario Determination Rules. 23
 - Define External System. 24
 - Define Rule Group. 25
 - Define Runtime Settings. 26
 - System Setup for Integrating Several External PLM Systems (N:1). 27

3.3	DM Change Object to SAP Change Master.	28
	Map DM Change Object to SAP Change Master.	28
	BAdI: Determine External System for ECN as CO Outbound Scenario.	31
	BAdI: Filter Relevant ECN for ECN as CO Outbound Scenario.	31
	BAdI: CO as ECN Post-Processing.	32
	BAdI: CO as ECN Post-Processing – Read Extension Fields.	33
3.4	DM Change Object to SAP Change Record.	33
	Define Current Settings for DM CO to SAP CR.	34
	Map DM Change Object to SAP Change Record.	34
	BAdI: Change Object as CR Responsible.	39
	BAdI: Determine External System for CR as CO Outbound Scenario.	39
	BAdI: Filter Relevant CR for CR as CO Outbound Scenario.	40
	BAdI: CO as Change Record Post-Processing.	41
	BAdI: CO as Change Record Post-Processing - Read Extension Fields.	41
3.5	DM Document Version to SAP Document Info Record.	42
	Define File Redirect Settings.	42
	Define Mapping for MIME Types.	42
	Map DM Document Version to SAP Document Info Record.	43
	BAdI: Control DV File Definition.	46
	BAdI: Determine External System for DV Outbound Scenario.	47
	BAdI: Document Version as DIR Setting for SAP Key.	48
	BAdI: DV as DIR Post-Processing.	49
	BAdI: DV as DIR Post-Processing – Read Extension Fields.	49
	BAdI: FC as File Upload Confirmation Post-Processing.	50
3.6	DM Factory to SAP Plant.	51
	Map DM Factory to SAP Plant.	51
3.7	DM IC to SAP Master Inspection Characteristic.	52
	Map DM IC to SAP Master Inspection Characteristic.	52
	Maintain Code Group Number Range.	53
	Maintain Selection Set Number Range.	54
	BAdI: IC as Master Inspection Characteristic Post-Processing.	54
3.8	DM POL to SAP Classic Routing.	55
	Map DM POL to SAP Classic Routing.	55
	Define POL Model Unit Template ECN.	57
	Maintain Code Group Number Range.	58
	Maintain Selection Set Number Range.	58
	BAdI: VC Selection Condition for POL as Classic Routing.	59
	BAdI: POL as Classic Routing Post-Processing.	59
3.9	DM POL to SAP PEO Routing.	60
	Map DM POL to SAP PEO Routing.	60
	BAdI: Determine External System for PEO Routing Outbound Scenario.	62

3.10	DM Product Version to SAP Material Master.	63
	Map DM Product Version to SAP Material Master.	63
	Define BRF+ Application for BAdI: PV as Material Post-Processing.	66
	Define Factory Template Material - DM PV to SAP Material Master.	67
	Define Product Templates - DM Product Version to SAP Material Master.	67
	BAdI: Control PV to Material Master Behavior.	68
	BAdI: Determine External System for PV Outbound Scenario.	69
	BAdI: PV as Material Key Reservation Post-Processing.	70
	BAdI: PV as Material Post-Processing.	70
	BAdI: PV as Material Post-Processing – Read Extension Fields.	71
3.11	DM Structure Version to SAP Classic BOM.	72
	Define SV Model Unit Template ECN.	72
	Map DM Structure Version to SAP Classic BOM.	72
	BAdI: VC Selection Condition for SV as Classic BOM.	75
	BAdI: SV as Classic BOM Post-Processing.	76
	BAdI: SV as Classic BOM Post-Processing – Read Extension Fields.	76
3.12	DM Structure Version to SAP PEO Version BOM.	77
	Define Version BOM Model Unit Template ECN.	77
	Map DM Structure Version to SAP PEO Version BOM.	78
	BAdI: Control Scale Factor of Transformations.	81
	BAdI: Filter Viewable Structure Items Based on Document Type.	82
	BAdI: SV as PEO Version BOM Grouping Custom Logic.	82
	BAdI: SV as PEO Version BOM Post-Processing.	83
	BAdI: SV as PEO Version BOM Post-Processing – Read Extension Fields.	84
3.13	DM Customer Order Item to SAP Sales Order Item.	84
	Map DM Customer Order Item to SAP Sales Order Item.	84
	BAdI: Determine External System for COI Outbound Scenario.	86
	BAdI: COI as Order Item Post-Processing – Read Extension Fields.	86
3.14	DM Customer Order Item Product Structure to SAP Sales Order BOM.	87
	Map DM Customer Order Item Product Structure to SAP Sales Order BOM.	87
	BAdI: Determine External System for COIPS Outbound Scenario.	89
	BAdI: COIPS as Order BOM Post-Processing.	90
3.15	DM Variant Definition to SAP Class.	91
	Map DM Variant Definition to SAP Class.	91
	Maintain Class ID Number Range.	92
	BAdI: VD as Class Post-Processing.	93
3.16	DM Variant Option Rule to SAP Object Dependency.	94
	Define Variant Configuration Engine.	94
	Map DM Variant Option Rule to SAP Object Dependency.	94
	Maintain Constraint Net ID Number Range.	95
	Maintain Dependency Number Range.	96

	Maintain Variant Table ID Number Range.	97
	BAdI: Set Reference Value for VOR Conditions and Expressions.	97
	BAdI: VOR as Characteristic Post-Processing.	98
3.17	DM Variant Option to SAP Characteristic.	99
	Define Variant Option Boolean Values.	99
	Map DM Variant Option to SAP Characteristic.	99
	Maintain Characteristic ID Number Range.	101
	Maintain Characteristic Value Number Range.	102
	BAdI: VO as Characteristic Post-Processing.	102
3.18	DM Vendor to SAP Business Partner.	103
	Map DM Vendor to SAP Business Partner.	103
	BAdI: Determine External System for Vendor Outbound Scenario.	105
	BAdI: Vendor as BP Post-Processing.	106
3.19	DM View Structure Version to SAP PEO MRS.	107
	Maintain MRS Object ID Number Range.	107
	Map DM View Structure Version to SAP PEO MRS.	107
	BAdI: VSV as PEO MRS Post-Processing.	108
3.20	Business Add-Ins (BAdIs) for PLM System Integration.	109
	BAdI: Define Custom Field and Value Mapping Conversions.	109
	BAdI: Define PEO Snapshot & Snapshot Revision Scope.	110
	BAdIs for Changing Message and Conversion.	111
	BAdIs for Defining Custom Logic.	112

1 About This Guide

This guide describes the configuration of PLM system integration for SAP S/4HANA for the integration with an external PLM system.

The following documentation uses the term "SAP S/4HANA" to refer to both SAP S/4HANA Cloud Private Edition and SAP S/4HANA.

PLM system integration for SAP S/4HANA supports the connection of multiple external PLM systems with deviating configurations, based on the logical instance and the application. For more information about the installation as well as security and operations, refer to the Administration Guide in the [SAP Help Portal](#).

Product Information

Product	PLM system integration for SAP S/4HANA
Release	3.0 FP02
Released	Q1, 2025

This guide addresses the following target audience:

- System administrators
- Technical consultants
- Key users

2 Initial Configuration

This chapter outlines the initial configuration steps you need to do.

Configuration Content

We provide standard content in the form of Business Configuration Sets (BC Sets) for the integration of an external PLM system.

You can use this content for the initial configuration of your SAP system. You can change and extend the configuration wherever necessary, to fit your needs.

Prerequisites

In case of data migration, you must migrate the necessary data before you activate the required BC Sets.

Before you can activate standard content, you need to do the following:

In Customizing, choose ► [Cross-Application Components](#) ► [Document Management](#) ► [General Data](#) ► [Define Data Carrier](#) ► Choose [Define data carrier type "server, front end"](#) and verify if an entry with description **Default** is maintained for type *PC*. If no such entry is maintained, create it and save your entry.

Content Activation

1. Use transaction `SCPR3` to view the available BC Set content.
2. Use transaction `SCPR20` to activate the provided BC Sets as required.

Note

Repeated activation of the same BC Sets may overwrite existing number range settings. If you need to activate the same BC Set twice, exclude the respective data and verify if number range data is ok after activation.

You can find information on all new and changed BC Sets for this release in the [What's New Viewer](#) for PLM system integration for SAP S/4HANA.

The technical names of all BC Sets for release 3.0 FPO2 of PLM system integration for SAP S/4HANA end in [_R3FP02](#).

Note

The number [R3FP02](#) in the BC Set name refers to the release number of PLM system integration for SAP S/4HANA, not to a release of SAP S/4HANA or SAP S/4HANA Cloud Private Edition! For information on the compatibility of your PLM system integration for SAP S/4HANA release with available releases of SAP S/4HANA or SAP S/4HANA Cloud Private Edition, refer to the [Compatibility Information for PLM System Integration](#).

The following BC Set sets were updated:

- /PLMS/BCS_OVERALL_R3FP02 - BC Set for all default configuration, contains all other BC Sets and subsets for [/PLMS](#)
- /PLMS/BCS_NUMBER_RANGES_R3FP02: BC Set for number ranges that are required for the integration. It is *mandatory* to activate this BC Set.
- /PLMS/BCS_AIF_R3FP02 – BC Set for AIF settings
- /PLMS/BCS_NON_PLMSI_R3FP02 - BC Set for non-PLMSI configuration and object configuration
- /PLMS/BCS_PLMSI_R3FP02 - BC Set for all business objects, global settings
- /PLMS/BCS_COI_R3FP02 - BC Set for Customer Order Item settings
- /PLMS/BCS_COIPS_R3FP02 - BC Set for Customer Order Item Product Structure settings
- /PLMS/BCS_DV_R3FP02 - BC Set for Document Version settings
- /PLMS/BCS_GS_R3FP02 - BC for Global Settings. This BC Set contains the following:
 - /PLMS/BCS_POL_R3FP02 - BC Set for Production Operation List settings
 - /PLMS/BCS_PV_R3FP02 - BC Set for Product Version settings
 - /PLMS/BCS_SV_R3FP02 - BC Set for Structure Version settings
 - /PLMS/BCS_VD_R3FP02 - BC Set for Variant Definition settings
 - /PLMS/BCS_VO_R3FP02 - BC Set for Variant Option settings
 - /PLMS/BCS_VOR_R3FP02 - BC Set for Variant Option Rule settings
 - /PLMS/BCS_IC_R3FP02 – BC Set for Inspection Characteristic settings
 - /PLMS/BCS_COCM_R3FP02 - BC Set for Change Object as Change Master settings
 - /PLMS/BCS_COCR_R3FP02 - BC Set for Change Object as Change Record settings
 - /PLMS/BCS_POL_PEO_R3FP02 – BC Set for POL-PEO settings
 - /PLMS/BCS_SVV_R3FP02 – BC Set for Version BOM settings
 - /PLMS/BCS_VDR_R3FP02 – BC Set for Vendor settings

We recommend that you activate the complete default configuration for [PLMS: /PLMS/BCS_OVERALL_R3FP02](#), which contains all other BC Sets and subsets for [/PLMS](#). Also, it is mandatory that you activate [/PLMS/BCS_NUMBER_RANGES_R3FP02](#) for number ranges.

Optionally, you can activate the following:

- [/PLMS/BCS_NON_PLMSI_R3FP02](#) (non-PLMSI configuration and object configuration)
- [/PLMS/BCS_COCR_R3FP02](#) (which contains SAP S/4HANA-specific settings for change object as change record)
- [/PLMS/BCS_COCM_R3FP02](#) (which contains SAP S/4HANA-specific settings for change object as change master)
- [/PLMS/BCS_AIF_R3FP02](#) (AIF logging).

You can verify the details of all BC Sets and subsets for [/PLMS](#) with transaction S`CP`R20.

Note

If there are activation issues, please choose the *Expert Mode* and start the activation again.

Note

Entering a business partner is a mandatory activity during the activation of [/PLMS/BCS_COCR_R3FP02](#). If no business partner information exists in your system, you need to manually create business partner data before activating [/PLMS/BCS_COCR_R3FP02](#). For more information, see [Mandatory Configuration Activities \[page 9\]](#).

Note

If you are integrating with SAP S/4HANA 2023 or higher, you need to additionally activate the BC Sets provided using SAP Note [3422031](#). Make sure you follow the exact instructions and sequence as outlined in this SAP Note! The activation of additional BC Sets is required due to additional settings added in SAP S/4HANA 2023. Note that errors with text `Table key not supported by activation links` should be ignored during activation of the BC set `ZPLMS_BCS_PIP_POL_R3FP02`, as they do not prevent the BC set from being activated. It is not necessary to reactivate `/PLMS/` BC Sets after an activation of `z` BC Sets.

3. Use transaction `SCPR20` to verify that the activation has been successful.

Content Management Settings

Testing your Content Server Settings

You can use the built-in content repository `DMS_C1_ST` to test your content server setup. For more information, see the chapter [Content Server Options and Installation](#) in the Administration Guide for PLM system integration for SAP S/4HANA.

Note

You need to make the following settings to ensure correct data exchange when using the built-in content repository `DMS_C1_ST`: Call up the [Display Content Repositories: Detail](#) screen and ensure the following settings are active:

- *HTTPS on frontend: HTTPS required*
- *HTTPS on backend: HTTPS required*

2.1 Mandatory Configuration Activities

This chapter outlines manual activities that you need to carry out for a successful configuration of PLM system integration for SAP S/4HANA.

Business Partner Data for Change Record

It is mandatory to provide *Business Partner* information during the activation of BC Set `/PLMS/BCS_COCR_R3FP02`. You can choose from existing business partners or create and use *Business Partner 101* in your system. To create this business partner, carry out the following steps:

1. Open transaction `BP` for change record integration.
2. Enter all required business partner data, including name, address, country and e-mail.
3. Save your entries. If needed, you can replace this business partner with other existing business partners later.

BC Set for Number Ranges

It is mandatory to activate BC Set /PLMS/BCS_NUMBER_RANGES_R3FP02: this BC Set contains the number ranges that are required for the integration. For more information, see [Initial Configuration \[page 7\]](#).

Additional BC Sets for an Integration With SAP S/4HANA 2023 or Higher

If you are integrating with SAP S/4HANA 2023 or higher, you need to additionally activate the BC Sets provided in [3422031](#). The activation of additional BC Sets is required due to additional settings added in SAP S/4HANA 2023.

3 PLMSI Configuration

This chapter outlines the configuration activities you must do to make full use of system integration with this add-on.

Complete all the necessary configuration steps in Customizing under [Cross-Application Components](#) > [PLM System Integration](#). There, you'll find all necessary customizing activities to configure this add-on.

For general information on Customizing, search for the chapter [The IMG Structure](#) on the [SAP Help Portal](#).

Version Configuration

You can configure which scenario is selected for a particular business process (based on a list of pre-defined scenarios), set which key and value mappings are to be used on which fields, and set which fixed values (for example, material industry sector, default document type, default class for documents, and so on) are used.

Value Mapping

This version supports 1:1 value mapping. Use Customizing for [Domain Model Code Lists \[page 11\]](#) to view and enhance the list of fields for which value mapping is possible.

Note

In Customizing, you can set up integrations that result in a mapping different external objects to the same SAP object. In this case, external data is merged. For consistency reasons, we recommend that you don't merge different external entities into one SAP entity (see also [3227954](#)).

3.1 Domain Model Code Lists

Domain Model code lists provide field values for the data exchange between the SAP system and the external PLM system.

In real-time business scenarios, fields can have different sets of valid values based on the context. For example, while creating a document in an SAP system, you may notice that the list of possible values for [DocumentType](#) may be different from the valid list of [DocumentTypes](#) in the external PLM system.

Also, it's possible for the source of the valid document types to be different in each of these scenarios. For example, the list of valid document types for an SAP document can come from the SAP system, while the list of valid document types for a document from an external PLM system can come from a domain. This means that

the source of valid values is different depending on the software version. Code Lists have been developed for these scenarios.

The Meta Domain Model uses code lists to define the allowed or expected values for certain fields. These values can then be used, for example, to map certain values from an external PLM system to the respective values in the SAP system (for example, material type).

Depending on your deployment, you can find sets of preconfigured values in this Customizing activity. You can modify or delete these values as required, or add additional values to enhance your customizing options.

You can define values for all available code lists or add new values, as needed. As a result, you can use the values defined here in subsequent configuration activities, for example, for field and value mapping.

The SAP system will check all values received by the external PLM system against the defined values for the respective fields.

Note

Default code lists are provided through BC Sets (see [Initial Configuration \[page 7\]](#)). You can enhance and modify the default code lists.

3.1.1 Define Domain Model Change Type

Define Domain Model code list values for change types.

Here, you can define Domain Model code list values for change types, for example, [Change Notice](#) or [Problem Report](#). These values can be used in the related configurations for change management.

3.1.2 Define Domain Model Consumption Type

Define Domain Model code list values for consumption types.

Here, you can define Domain Model code list values for consumption types, for example, [Consumption](#) or [Inspection](#). These values can be used in the related configurations for production operation list integration.

For each Domain Model consumption type, you can use the [Domain Model Backflush Relevant](#) checkbox. If you set the consumption type to backflush relevant, the system sets the reference value of the logical component created in the PEO XML document for this consumption type to false. If the consumption type is not backflush relevant, the system sets the reference value of the logical component to true.

3.1.3 Define Domain Model Criticality Code

Define Domain Model code list values for criticality codes.

Here, you can define Domain Model code list values for criticality codes, for example, [low](#). These values can be used in the related configurations for inspection characteristic integration.

3.1.4 Define Domain Model Customer Order Type

Define Domain Model code list values for customer order types.

Here, you can define Domain Model code list values for customer order types, for example, [Sales Order](#). These values can be used in the related configurations for customer order integration.

3.1.5 Define Domain Model Customer Order Item Product Structure Type

Define Domain Model code list values for customer order item product structure (COIPS) types.

Here, you can define Domain Model code list values for customer order item product structure (COIPS) types, for example, [Sales Order BOM](#). These values can be used in the related configurations for COIPS integration.

3.1.6 Define Domain Model Document Type

Define Domain Model code list values for document types.

Here, you can define Domain Model code list values for document types, for example, [Drawing](#). These values can be used in the related configurations for document version integration.

3.1.7 Define Domain Model Embedded Inspection Characteristics Type

Define Domain Model code list values for embedded inspection characteristics types.

Here, you can define Domain Model code list values for embedded inspection characteristics types, for example, [Dimension](#) or [Duration](#). These values can be used in the related configurations for inspection characteristics integration.

3.1.8 Define Domain Model Factory Type

Define Domain Model code list values for factory types.

Here, you can define Domain Model code list values for factory types, for example, [Production](#). These values can be used in the related configurations, for example, for vendor integration.

3.1.9 Define Domain Model Inspection Characteristics Type

Define Domain Model code list values for inspection characteristic types.

Here, you can define Domain Model code list values for inspection characteristic types, for example, [Quantitative](#) or [Qualitative](#). These values can be used in the related configurations for inspection characteristics integration.

3.1.10 Define Domain Model Maturity Code

Define Domain Model code list values for maturity codes.

Here, you can define Domain Model code list values for maturity codes, for example, [Prototype](#) or [End of Life](#). These values can be used in the configurations for any business object, for example, product version or structure version integration.

3.1.11 Define Domain Model MIME Type

Define Domain Model code list values for MIME types

Here, you can define Domain Model code list values for MIME types, for example, [application/vnd.ms-powerpoint](#). These values can be used in the related configurations in document version integration.

3.1.12 Define Domain Model Product Type

Define Domain Model code list values for product types.

Here, you can define Domain Model code list values for product types, for example, [Raw Material](#) or [Finished Product](#). These values can be used in the related configurations for product version integration.

3.1.13 Define Domain Model Production Operation List Type

Define Domain Model code list values for production operation list types.

Here, you can define Domain Model code list values for production operation list types, for example, [Manufacturing](#). These values can be used in the related configurations for production operation list integration.

3.1.14 Define Domain Model Relation Type

Define Domain Model code list values for relation types.

Here, you can define Domain Model code list values for relation types, for example, [Reference](#) or [Solution](#). These values can be used in the related configurations for change object integration.

3.1.15 Define Domain Model Semantic File Type

Define Domain Model code list values for semantic file types.

Here, you can define Domain Model code list values for semantic file types, for example, [Work Instruction](#) or [Component Specification](#). These values can be used in the related configurations in document version integration.

3.1.16 Define Domain Model Status Code

Define Domain Model code list values for status codes.

Here, you can define Domain Model code list values for status codes, for example, [Approved](#) or [Blocked](#). These values can be used in the configurations for any business object, for example, product version or structure version integration.

3.1.17 Define Domain Model Structure Type

Define Domain Model code list values for structure types.

Here, you can define Domain Model code list values for structure types, for example, [Engineering BOM](#) or [Manufacturing BOM](#). These values can be used in the related configurations for structure version integration.

3.1.18 Define Domain Model View Structure Version Type

Define Domain Model code list values for view structure version (VSV) types.

Here, you can define Domain Model code list values for view structure version (VSV) types, for example, [Physical](#). These values can be used in the related configurations for VSV integration.

3.1.19 Define Domain Model View Structure Version Edge Type

Define Domain Model code list values for view structure version (VSV) edge types.

Here, you can define Domain Model code list values for view structure version (VSV) edge types, for example, [Parent to Child Relation](#). These values can be used in the related configurations for VSV integration.

3.1.20 Define Domain Model View Structure Version Node Type

Define Domain Model code list values for view structure version (VSV) node types.

Here, you can define Domain Model code list values for view structure version (VSV) node types, for example, [Root](#). These values can be used in the related configurations for VSV integration.

3.1.21 Define Domain Vendor Type

Define a Domain Model code list for vendor types.

Here, you can define Domain Model code list values for vendor types, for example, [Supplier](#) or [Manufacturer](#). These values can be used in the related configurations for vendor integration.

3.1.22 Define Domain Model Work Area Type

Define Domain Model code list values for work area types.

Here, you can define Domain Model code list values for work area types, for example, [Line](#) or [Unit](#). These values can be used in the related configurations for factory integration.

3.2 Scenario and Version

In this section, you can define data for different scenarios and scenario versions.

These settings are used to differentiate between integration configuration setups.

3.2.1 Define Scenario and Version

In this Customizing activity, you create definitions for the scenario and version. These settings are used to differentiate between integration configuration setups.

A scenario includes settings for inbound and outbound messages. The external application selects the relevant scenario either as defined by an incoming message or based on the following attributes:

- External application
- API version
- Component version
- Logical instance

Note

A scenario can only exist with a version. For more information on the handling of Domain Model versions during inbound data transfer, see [3227948](#).

Background

For a specific integration configuration, you first create a scenario and an initial version. Subsequently, you can create additional scenario versions for each scenario. A scenario version defines a certain set of configurations. Scenario versions can be maintained independently.

You define specific scenario determination criteria that are used to determine which scenario and scenario version is used for a specific integration run.

Activities

1. Create a new scenario ID or select an existing scenario ID.
2. Choose *Scenario Version* and define detailed data for the required scenario: Enter a *Scenario Description* and *Key Mapping Mode*: The Key Mapping mode controls key mapping for inbound objects.
 - *Acceptance Mode*: All tempPLMIDs that are received for a business ID are stored.
 - *Overwrite Mode* (default): Latest tempPLMID that is received for a business ID is stored as active tempPLMID.
 - *Exclusive Mode*: One tempPLMID is stored for a business ID. In case the business ID is received with a deviating tempPLMID, an error is raised during message processing.
3. Select if AIF logging is active (on) or not (off).
4. Define which object-related messages are returned in the message result: Select the relevant SAP message type using the *Send API System Message Type (Send Type)* dropdown list.
 - For more information, see [Error Handling](#).
5. *Define Value Mapping*: In this activity, you can map values coming from an external PLM system to values in the SAP system. You can use or create a value mapping ID and assign specific values to this value mapping ID.

Example: Mapping

You want to map an external value for a *Basic Material* to a basic material in the SAP system.

1. Select a scenario and scenario version.
2. Choose *Define Value Mapping* and choose a value mapping ID.
The *Mandatory* checkbox controls whether the system issues a warning or an error in case a value is missing. Select the checkbox according to the desired result:
 - *Mandatory* checkbox is selected: If a value is missing, an error is issued, and processing is interrupted at this step.
 - *Mandatory* checkbox isn't selected: If a value is missing, a warning is issued, and the default value from the mapping is used.
3. Choose *Define Mapping Values* and assign the source value, in this case a basic material:
 - Choose the direction for value mapping, for example, inbound (in).
 - In the *Source Value* field, enter the material, for example, *Metal*.
 - In the *Target Value* field, enter the internal material you would like to map to, for example, *SAP Metal*.

You can now use this value when creating a product version, document version, structure version (inbound values only) or change object (inbound values only).

6. *Assign Integration Pattern*: Here, you assign an integration pattern to a combination of object type and Domain Model type. Integration patterns enable you to select different target applications for a defined integration object.

Example: The DM type *ChangeObject* can be mapped to an *SAP Change Master* or an *SAP Change Record* in the SAP system.

SAP delivers default integration patterns for typical implementations (see [Integration Patterns \[page 19\]](#)). When a request is processed, the system groups objects according to the assignments of integration patterns to the Domain Model type of these objects. Thus, you can differentiate implementations by Domain Model type. If no Domain Model type is specified, the system uses the integration pattern that is assigned to a certain object type for all Domain Model types that are available for this object type.

To assign an integration pattern, select the required *Object Type*, for example, *CO*, and *Integration Pattern*, for example, *SAP_COasChangeMaster*. Use the *F4* help to select the *Domain Model Type* for which you want to use this integration pattern, for example, *Problem Report*. You can assign one integration pattern per combination of Domain Model type and object type.

7. *Define Compatibility Mode*: Here, you define a compatibility mode for a certain integration pattern. The *Compatibility Mode* defines whether a key mapping is created between inbound object IDs and PLM entities in the SAP system, if existing objects have not been created by the integration, and if certain preconditions are met. Select an integration pattern with the *F4* help and set the required compatibility mode:

- If *SAP_COIPsAsOrderBillOfMaterial* is selected, a key mapping is created between inbound customer order item product structure IDs (header and components) and PLM entities in the SAP system.
Precondition Example: When an order bill of material with the provided validity date is received, the change object must have a key mapping (key mapping has been created by the integration). There is no automatic mapping of change objects.

COIPS compatibility mode is skipped in the following case:

- A received DM component for the provided valid-from date is not found in the system.

If a product version received as part of a COIPS does not exist in the SAP system, the COIPS compatibility mode is rejected.

- If *SAP_PVasMaterialMaster* is selected, a key mapping is created between inbound product version IDs and PLM entities in the SAP system. Precondition Example: When a material with an attached revision

level is received, the change object must have a key mapping (key mapping has been created by the integration). There is no automatic mapping of change objects or linked documents.

- If *SAP_SVasClassicBillOfMaterial* is selected, a key mapping is created between inbound structure version IDs (header and components) and PLM entities in the SAP system. Precondition Example: When a bill of material with the provided validity date is received, the change object must have a key mapping (key mapping has been created by the integration). There is no automatic mapping of change objects.

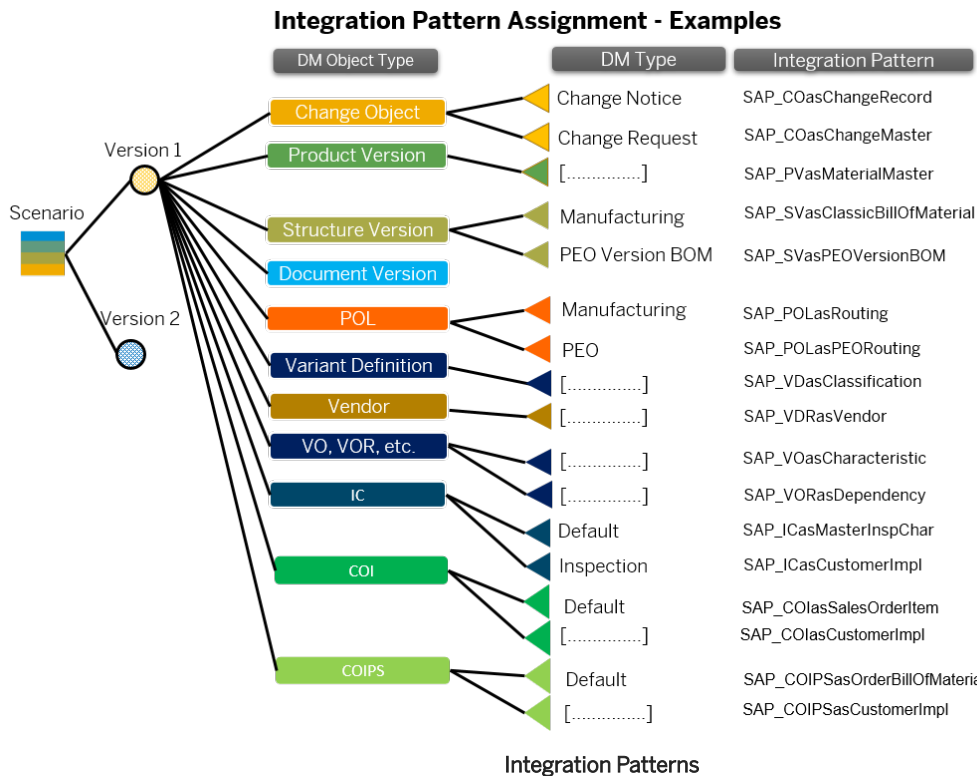
SV compatibility mode is skipped in the following case:

- At least one component with parameter effectivity is sent.
- A received DM component for the provided valid-from date is not found in the system.
If a product version received as part of an SV does not exist in the SAP system, the SV compatibility mode is rejected.
- If *SAP_VDRasVendor* is selected, a key mapping is created between inbound vendor IDs and PLM entities in the SAP system. Precondition Example: No internal numbering is activated for vendor configuration.

3.2.1.1 Integration Patterns

Integration patterns enable you to select different target applications for a defined integration object.

Each integration pattern represents one target SAP application.



The table below gives an overview of the available integration patterns. You can choose to use your own coding and integration logic to replace the standard implementation for integrating a certain object. To

do so, use the [F4](#) help to select one of the customer implementation integration patterns, for example, [SAP_DVasCustomerImplementation](#).

Note

Relations to other objects have to be handled by the customer/partner implementation.

Note

The following customer-specific integration patterns are not supported for the Change Record outbound integration process in PLM system integration (PLMSI) 3.0 FP02:

- `SAP_PVasCustomerImplementation` for Product Version
- `SAP_DVasCustomerImplementation` for Document Version
- `SAP_COIasCustomerImpl` for Customer Order Item
- `SAP_COIPsasCustomerImpl` for Customer Order Item Product Structure

Integration Patterns for PLM system integration for SAP S/4HANA

Object Type	Integration Pattern	Use	Recommendation
Change Object (CO)	<code>SAP_COasChangeMaster</code>	The DM change object data is mapped to an SAP change master. You can define mapping details in Map DM Change Object to SAP Change Master [page 28]	The default DM type change notice is typically mapped to an SAP change master.
Change Object (CO)	<code>SAP_COasChangeRecord</code>	The DM change object data is mapped to an SAP change record. You can define mapping details in Map DM Change Object to SAP Change Master [page 28]	The default DM types change request and problem report are typically mapped to an SAP change record.
Document Version (DV)	<code>SAP_DVasDocumentInfoRecord</code>	The DM document version object data is mapped to an SAP document info record. You can define mapping details in Map DM Document Version to SAP Document Info Record [page 43] .	Please note: <code>SAP_DVasCustomerImplementation</code> is used for document version custom logic (see BA1: Document Version (DV) Custom Logic [page 116]) and <code>SAP_DVasCustomerImpl</code> is used for document version file custom logic (see BA1: Document Version File (DVF) Custom Logic [page 117]).

Object Type	Integration Pattern	Use	Recommendation
Document Version File (DVF)	SAP_DVasDocumentInfoRecord	The DM file data is mapped to a file read request. You can define mapping details in Map DM Document Version to SAP Document Info Record [page 43] .	Note: SAP_DVasCustomerImplementation is used for document version custom logic (see BAdI: Document Version (DV) Custom Logic [page 116]) and SAP_DVFasCustomerImpl is used for document version file custom logic (see BAdI: Document Version File (DVF) Custom Logic [page 117]).
File Confirmation (FC)	SAP_FCasFileUploadConfirmation	The DM file data is mapped to an original of a file upload confirmation. You can define mapping details in Map DM Document Version to SAP Document Info Record [page 43] .	
Inspection Characteristic (IC)	SAP_ICasMasterInspChar	The DM inspection characteristic object data is mapped to SAP master inspection characteristics. For more information, see Map DM IC to SAP Master Inspection Characteristic [page 52] .	
Customer Order Item	SAP_COlasSalesOrderItem	The DM customer order item is mapped to SAP sales order item types. For more information, see Map DM Customer Order Item to SAP Sales Order Item [page 84] .	
Customer Order Item Product Structure	SAP_COIPsasOrderBillOfMaterial	The DM customer order item product structure is mapped to SAP sales order BOM types. For more information, see Map DM Customer Order Item Product Structure to SAP Sales Order BOM [page 87] .	
Production Operation List (POL)	SAP_POLasRouting SAP_POLasPEORouting	The DM production operation list data is mapped to an SAP routing or SAP PEO routing. You can define mapping details in Map DM POL to SAP Classic Routing [page 55] , Map DM POL to SAP PEO Routing [page 60] .	

Object Type	Integration Pattern	Use	Recommendation
Product Version Key Reservation (PR)	SAP_PRasMaterialMaster	The DM product version key reservation data is mapped to an SAP material master. You can define mapping details in Map DM Product Version to SAP Material Master [page 63] .	
Production Version (PV)	SAP_PVasMaterialMaster	The DM product version data is mapped to an SAP material master. You can define mapping details in Map DM Product Version to SAP Material Master [page 63] .	
Structure Version (SV)	SAP_SVasClassicBillOfMaterial	The DM structure version data is mapped to an SAP classic BOM. You can define mapping details in Map DM Structure Version to SAP Classic BOM [page 72] .	
Structure Version (SV)	SAP_SVasPEOVersionBOM	The DM structure version data is mapped to an SAP PEO version BOM. You can define mapping details in Map DM Structure Version to SAP PEO Version BOM [page 78] .	
Variant Definition (VD)	SAP_VDasClassification	The DM variant definition data is mapped to an SAP classification. You can define mapping details in Map DM Variant Definition to SAP Class [page 91] .	
Variant Option (VO)	SAP_VOasCharacteristic	The DM variant option is mapped to an SAP characteristic. You can define mapping details in Map DM Variant Option to SAP Characteristic [page 99] .	
Variant Option Rule (VOR)	SAP_VORasDependency	The DM variant option rule is mapped to an SAP object dependency. You can define mapping details in Map DM Variant Option Rule to SAP Object Dependency [page 94] .	

Object Type	Integration Pattern	Use	Recommendation
Vendor (VDR)	SAP_VDRasVendor	The DM vendor data is mapped to an SAP business partner. You can define mapping details in Map DM Vendor to SAP Business Partner [page 103] .	
View Structure Version (VSV)	SAP_VSVasPEOMRS	The DM view structure version (VSV) is mapped to an SAP manufacturing reference structure (MRS). You can define mapping details in Map DM View Structure Version to SAP PEO MRS [page 107] .	

3.2.2 Define Parallel Processing Settings for Inbound

Define how to parallelize and package the processing of the objects included in one payload, in order to improve performance.

You can make these settings for several object types, for example, ProductVersion or StructureVersion, and the request type *Maintain*.

1. Choose the *Object Type*, *Request Type*, *Integration Pattern* (select using the *F4* help), *Logon/Server Group*, and *Default Package Size*.

Note

The default package size defines the ideal size for packages. Depending on the available data, the package size may differ.

You need to choose an integration pattern using the **F4** help. You can choose from all integration patterns that offer at least one task that allows for parallelization.

2. Select the *Active* checkbox to activate parallel processing.

You can disable parallelization for your user by setting the `/PLMF/DISABLE_PAR` parameter in transaction **ST03**. This disables parallelization even if it is activated in Customizing for the respective scenario version combination. If parallelization is active but disabled in the user settings, the system issues a warning.

3.2.3 Define Scenario Determination Rules

Define the necessary scenario determination criteria. You can specify attributes to be valid for any received value by flagging them as generic, as required.

Choose an external application and select attributes depending on your company's needs. Choose *F4* for more details on each attribute.

You can select the *Highest Available Version* option or enter a specific version to use in the *Version* field. The highest version is determined by sorting the versions in alphabetical order.

You can configure that the *Scenario ID* and *Scenario Version* to be used are defined by the sender. To do so, select the option *Scenario Determination by Message*.

Note

If *Scenario Determination by Message* is selected, the following conditions apply:

- The relevant lines are defined by *External Application*, *API Version*, *Logical Instance*, and *Component Version*.
- The scenario ID and scenario version are selected according to the values defined in the payload message header data for *configuration* and *configurationVersion*.
- It is not possible to enter a scenario and scenario version in the configuration.
- The *Highest Available Version* option can't be set.
- If no relevant scenario and scenario version can be found, an error message is generated, and data processing stops.

Define Scenario Determination - Configuration Example for Siemens Teamcenter Integration

External Application	External Application Generic	API Version Generic	API Version	Component Version Generic	Component Version	DM Logical Instance	Logical Instance Generic	Scenario Determination by Message	Highest Available Version	Scenario ID (scenario to be used)	Scenario Version
	X	X		X				X	X	SAP_Default	
TEAM-CENTER			1		20					SAP_Default	002
TEAM-CENTER			2		20					SAP_Specific1	001
TEAM-CENTER	X				21.1					SAP_Teamcenter_21.1	004
TEAM-CENTER			2	X						SAP_other_example	001
									X		

3.2.4 Define External System

Define the external system for outbound data transfer.

You define the receiving (target) systems for sending data from the SAP system to an external PLM system.

PLM system integration for SAP S/4HANA is designed to support integration between several external PLM systems and one SAP system (N:1).

For an *External Logical System* and *External Application*, specify the *RFC Destination* (mandatory), *APIVersion* (default is 0), and *Component Version*. Note that as of release 2.0 FP02, the API version must be 2. For more information, see [Installing PLM system integration for SAP S/4HANA](#).

For N:1 integrations, rule group configuration and business object-specific outbound condition configurations are used to determine the External Logical System and External Application which specify the RFC Destination (mandatory), APIVersion (default is 0), and Component Version that are relevant for integration. The scenario determination logic uses these values from the external system configuration to retrieve the scenario/version that will be used to build up the request for external system.

Note

Change object integration can handle one SAP system to one external system integration (1:1) and several external PLM systems to one SAP system (N:1) integration. The *Change Object Default Destination* checkbox is used to switch the behavior. If this checkbox is marked, the *External Logical System*, *External Application*, *RFC Destination*, *API Version*, and *Component Version* are determined directly from the external system configuration to handle change object integration. If the checkbox is not marked, the system uses the [BAdI: Determine External System for CR as CO Outbound Scenario \[page 39\]](#) for filtering. The change object 1:1 system integration has not changed compared to previous releases and supports your business processes as usual.

The values for *Time Zone* and *Time Rounding* define the time zone of the external system and how the SAP system handles and converts the validity of incoming and outgoing object data (*ValidFrom*), for example, for change objects:

The SAP system converts incoming timestamp data (*ValidFrom*) in accordance with the customizing settings for the external system: You can choose to round up or down. Internally, the SAP system only uses date format for validity. For empty (*ValidFrom*) data, the system uses the default date *01.01.1900*. The default valid-from date may be overridden by object-specific customizing. It is not considered when processing objects with a change master. When returning time data to the external system, the SAP system interprets the stored valid-from value with the customized time zone of the target system.

3.2.5 Define Rule Group

Rule groups are used to assign certain object types to a specific external PLM system for outbound integration.

You can define rule groups through *Rule Group IDs* and *Rule Group Descriptions* and map them to an external system.

1. Choose *Rule Group Definition* to define a new rule group ID and description or choose from the existing values.
2. Choose *Rule Group System Assignment* and assign an *External Logical System* and *External Application* to the *Rule Group ID*. Select an *Operation* to define how the SAP object interaction behaves based on the existence of key mapping in the target system. Choose from the following options:
 - *SAP Update*: The SAP object is updated but the key mapping does not exist for the target system.
 - *DM Update*: The SAP object is updated, and the key mapping exists in the target system.
 - *DM New*: The SAP object is created/updated but the key mapping does not exist for any target systems.

Select the *CR Relevance* checkbox for outbound scenarios where the change object is represented as a change record. These values are used in the view *Define Relevance Type* to provide the relevant *Change Record Item Relevance* types.

3. Choose *Define Relevance Type*, verify that the *CR Relevance* checkbox is selected, and enter the required data in *Change Record Item Relevance*.

Note

Relevance types can be defined for outbound-relevant scenarios where the change object is represented as a change record to determine sent solution items and reference object items. If no values are selected, the rule applies to all relevance types. If relevance types are selected, the rule is only relevant for the objects that are assigned to a CR with one of the listed relevance types in context of the assigned outbound system.

4. Save your entries.

3.2.6 Define Runtime Settings

Here, you can define the runtime settings for the integration.

You can select if you want to log trace information. If tracing is activated, additional technical info messages will be logged during processing of inbound and outbound messages. These messages can be viewed in the Business Application Log (BAL) / SLG1 for the corresponding objects:

- /PLMF/SERVICE (Inbound)
- /PLMF/EVENTS + /PLMF/PLMSI_OUT (Outbound)

You can choose from the following options:

- Off - Default, tracing disabled.
- On - Tracing enabled.
- On (Errors Only) - Tracing enabled for errors only.

You can also enable trace logging using the user parameter */PLMF/ENABLE_TRC*:

- If the user parameter is set, it overwrites the configuration setting.
- If it is not set or initial, the configuration setting applies.
- If tracing is set to *On* in the configuration and the user parameter is set to *Errors Only*, only errors are traced.

You can set user parameters using the transaction *SU3*.

Note

When you activate tracing, the system starts to collect a vast number of additional messages - this may affect system performance. We recommend that you use tracing for analyzing purposes only. You should not use tracing in a productive system!

You can define the *RFC Destination of the MDG Hub*. This is necessary to define the RFC destination of the MDG hub system that is used for all MDG relevant processes. Once you have defined a destination here, it is used instead of the local MDG.

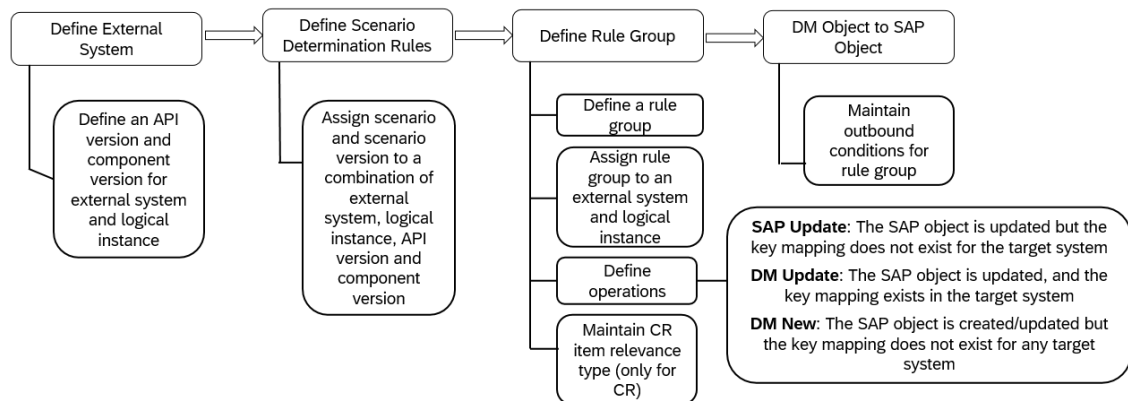
3.2.7 System Setup for Integrating Several External PLM Systems (N:1)

PLM system integration for SAP S/4HANA is designed to support the integration between several external PLM systems and one SAP system (N:1).

To set up the integration between several external PLM systems and one SAP system, you need to follow the steps as outlined here:

N:1 System Setup for Outbound Maintain Requests and Data Federation (Inbound/Outbound)

Configuration Steps under
Cross-Application Components -> PLM System Integration -> Scenario and Version



Based on the maintained conditions, the object-specific BADIs for determining external systems will determine the target systems per object type. You can adjust all default BAdI implementations as needed.

Related Information

[Define External System \[page 24\]](#)

[Define Scenario Determination Rules \[page 23\]](#)

[Define Rule Group \[page 25\]](#)

3.3 DM Change Object to SAP Change Master

This node groups activities for mapping change objects to SAP change masters.

3.3.1 Map DM Change Object to SAP Change Master

In this Customizing activity, you maintain all necessary definitions to map the change object data from the Meta Domain Model to the respective SAP change object.

The SAP change object represents a change number from the change master and its revision.

Note

Change object integration can handle 1:1 integration of one SAP system to one external system integration and N:1 integration of several external PLM systems to one SAP system integration. For more information, see [Define External System \[page 24\]](#).

We recommend that you follow the configuration steps in the given order.

Prerequisites

- You have made all relevant settings in Customizing for [Domain Model Code Lists \[page 11\]](#). You can add more values as needed before you start mapping, or during the mapping process.
- Scenarios and versions have been configured in the system, in Customizing for [Define Scenario and Version \[page 17\]](#).
- You have maintained a change master (CM) profile in Customizing for [Maintain Profile](#) under **► Cross-Application Components** **► Engineering Change Management** **▾**.
- The integration pattern for the relevant change object type is set to `SAP_COasChangeMaster`. For more information, see [Integration Patterns \[page 19\]](#).

Activities

1. Select a *Scenario ID* and *Scenario Version* to specify the scenario ID and version for which you want to define and map change object data.
2. **Define Mapping of Change Object Type:** Here, you map a Domain Model change object type to the SAP change object type.
Choose the Domain Model change type, default authorization group, and Change Master (CM) profile. For outbound mapping, you can set a *Default Outbound Type* (to define that the respective Domain Model change type is used in change object outbound integration), *Default Outbound Item Type* (it defines the Domain Model relation type used for change object items in outbound integration of change master data), and whether the *Outbound Communication ID Mode* will be defined by the SAP system or the external

PLM system. Use the (ECN outbound) *Solution Item Type* checkbox to distinguish whether items or change objects are part of the solution items or related items section. You can also set a *Valid-From* date. It's used if the external system does not provide a valid-from date. If you leave this field empty, the system uses a fallback date.

You can maintain a change master profile in Customizing for *Maintain Profile* under **► Cross-Application Components ► Engineering Change Management ►**.

When you integrate the following objects, ensure that the profile enables ECN usage as outlined in the following table:

Object Type Profile

ECN Object Type	DM Object Type
Material BOM	SV
Task List	POL
Document	DV
Material	PV with revision
Characteristics of Class	Variant Model
Object Dependencies	SV/POL with selection conditions
Variant Table	Variant Model

3. **Define Change Object Status:** Here, you define a combination of Domain Model status, maturity code, and SAP ECN status. If there is a change, the change object status will be modified as set here. Select the conditions under which the update happens. Use the *Outbound-Relevant* checkbox to define the relevant SAP status to DM status and DM maturity mapping for outbound requests.
4. **Define Inbound Field Mapping and Define Outbound Field Mapping:**
Use field mapping to decide how fields from the source structure are transferred to fields in the target structure. You can use *F4* help to find the respective values. The help provides all fields including fields from the structure and included structures. To define field mapping, you do the following:
 - Choose *Create*, *Update*, or *Modify* in the column *MapMode*, to define what is valid for this line. (*Modify* includes *Create* and *Update*).
 - Define an SAP target structure and SAP target attribute to define which field in the SAP system is to be set. Choose *F4* to see the available values. If the *Overwrite* checkbox is selected, the system uses the stored value instead of a value from the payload. This checkbox is for information purposes only and the setting can't be changed.
 - Define a Domain Model source structure (you can also use <EXTENSION_FIELDS>) and Domain Model source attribute to define which data will be mapped to the target system.
 - Define a conversion in case you want to transform the inbound data before setting it to the target field. You can use preconfigured conversions, and conversions that are specific to your customer or target data. For more information, see [BAdI: Define Custom Field and Value Mapping Conversions \[page 109\]](#).
 - Define a default value if you want to set the standard value in case no data is received in the source field or if no source field is mapped. Note that you must define a value in internal format.

ⓘ Note

If you expect partial payloads, make sure you maintain the settings for *FVM Default Value Mode*.

Field and value mapping supports customer field extensibility by extension fields.

You can use <EXTENSION_FIELDS> to map extension fields that are sent from the external PLM system to a standard field or classification in the SAP system. The name of the extension field must be passed to the source attribute field.

The assignment is done using standard technical conversion rules. You can define a domain value mapping to do data conversions using tables. You can also define a conversion function. The system offers a set of standard conversion functions. You can extend these functions using a BAdI. For more information, see [BAdI: Define Custom Field and Value Mapping Conversions \[page 109\]](#). If a data type other than *DATS* is used, use the *Timestamp* conversion function to map date and time information from a text field into a timestamp in the SAP system. This enables the conversion between the long timestamp used in the Meta Domain Model and the SAP format.

5. **Define Inbound Class Assignment:** Here, you can assign classes and class types that are added to a change object when it's created and updated, when the change object is sent from the external PLM system to the SAP system. Choose class type 025 to assign a classification to an SAP change master. You can choose a conversion routine (if needed), set default values, or set a value mapping for the value transfer. You can also use these class assignments as a mapping target. You can map to an SAP characteristic (target) and define which Domain Model source field from the Domain Model input structure will be used, for example, to map a status to a characteristic.
 - **Define Inbound Characteristics Mapping:** Here, you define the value mapping for the characteristics of the class. You map values of the Domain Model characteristics to the SAP target fields and extension fields of the change object.
6. **Define Outbound Class Assignment:** Here, you can assign classes and class types that are added to a change object when it's created and updated, when the change object is sent from the SAP system to the external PLM system. You can choose a conversion routine (if needed), set default values, or set a value mapping for the value transfer. You can also use these class assignments as a mapping source. You can map a characteristic and define which DM target field in the DM target structure will be used, for example, to map a characteristic to a status.
 - **Define Outbound Characteristics Mapping:** Here, you define the value mapping for the characteristics of the class. You map values of the characteristics to the Domain Model target fields and extension fields of the change object.
7. **Define Conditions for Outbound:** You can use the fields *Change Table Name* and *Attribute Name* to define which fields in the change master tables trigger the transfer to the external PLM system when they are updated in the SAP system.
 - **Rule Group ID:** Defines a group of rules. Based on the rule group ID, the system determines the correct target systems for outbound data transfer. Target systems can be defined in the configuration activity [Define Rule Group \[page 25\]](#).
 - Enter a number in the *Counter for Outbound Attribute* field. You can maintain multiple entries for the same combination of *Change Table Name* and *Attribute* to specify different values.
 - **Value From:** If the configuration contains a value in the *Value From* field, the original material value is ignored, and only a change in the attribute is considered relevant.
 - **Value From Any:** If *Value From Any* is selected, the original material value is ignored, and only a change in the attribute is considered relevant.
 - **Value To:** If the configuration contains a value in the *Value To* field, the change of the material must set this value for the defined field to be valid.
 - **Value To Any:** If *Value To Any* is selected, the actual target material value is ignored, and only a change in the attribute is considered relevant.

3.3.2 BAdI: Determine External System for ECN as CO Outbound Scenario

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to determine target systems that are relevant for ECN outbound integration.

/PLMF/BD_EC_N_TGT_SYST_DET_N

Note

This BAdI is used for N:1 system mapping only.

This BAdI verifies if a system is relevant for outbound integration and returns a list of relevant target systems:

- Target systems for which the sent DM change object type is marked as outbound-relevant in Customizing for [Map DM Change Object to SAP Change Master](#) > [Define Mapping of Change Object Type](#) , see [Map DM Change Object to SAP Change Master \[page 28\]](#).
- Target systems for which changes have been made to a sent change master.

This BAdI offers a default implementation that is called at runtime but can be overridden. Systems that you define as target systems will receive changes that affect ECNs in the SAP system.

For more information about the standard settings (filters, single or multiple uses), see the [Enhancement Spot Element Definitions](#) tab in the BAdI Builder (transaction [SE18](#)).

Activities

1. Create the implementation of the enhancement spot.
2. Adapt the data from the template as required.
3. Save your entries.

3.3.3 BAdI: Filter Relevant ECN for ECN as CO Outbound Scenario

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to specify the engineering change numbers (ECN) that are sent as change objects (CO) to an external PLM system.

/PLMF/BD_OUTBOUND_CO_EC_N

Note

This BAdI is used for 1:1 system integration only.

Standard Settings

This BAdI returns ECNs:

- For which the *Change Object* DM object type is uniquely determined from the L1 mapping configuration in *Map DM Change Object to SAP Change Master*
- For which the status is in the list of relevant statuses specified in the *Value To* attribute of the *Define Conditions for Outbound* view
- For which object management records refer to objects integrated by PLMSI, for example, document version or product version

For more information about the standard settings (filters, single or multiple uses), see the Enhancement Spot Element Definitions tab in the BAdI Builder (transaction [SE18](#)).

Activities

1. Choose *Create*. Create the implementation of the enhancement spot.
2. Adapt the data from the template as required.
3. Save your entries.

3.3.4 BAdI: CO as ECN Post-Processing

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to enhance the SAP integration with customer-specific steps after object processing for change objects has been performed successfully for inbound maintain request types.

```
/PLMF/BD_CO_CUST_POSTPROC_STEP
```

This BAdI can raise messages that influence the overall status. Additional messages are added to the existing messages and are written to the logs after the SAP process flow. The status is set accordingly based on the error category.

Note

You cannot use this BAdI to change the payload.

This BAdI has no default implementation. For more information about the standard settings (filters, single or multiple uses), see the *Enhancement Spot Element Definitions* tab in the BAdI Builder (transaction [SE18](#)).

Activities

1. Choose *Create*.

2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.3.5 BAdI: CO as ECN Post-Processing – Read Extension Fields

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to enhance the SAP integration with customer-specific steps after object processing for change object as ECN has been performed successfully for inbound read request types.

```
/PLMS/BD_CO_ECN_READ_EXTN_FLD
```

This BAdI can read extension fields to add into the outbound data. Additional extension fields are added to the existing ones.

Note

You cannot use this BAdI to change the payload.

For more information about the standard settings (filters, single or multiple uses), see the [Enhancement Spot Element Definitions](#) tab in the BAdI Builder (transaction SE18).

Activities

1. Choose [Create](#).
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.4 DM Change Object to SAP Change Record

This node groups activities for mapping change objects to SAP change records.

3.4.1 Define Current Settings for DM CO to SAP CR

In this Customizing activity, you maintain current settings for mapping the change object types from the Meta Domain Model to the respective SAP change record types.

Note

Flex workflows are not supported for change records that are created through the integration. If you need to use flex workflows, you need to create the change record manually.

We recommend that you follow the configuration steps below in the given order.

Activities

1. Select a *Scenario ID* and *Scenario Version* to specify the scenario ID and version for which you would like to define and map change object. Choose *Continue*.
2. Select a DM change type, for example, *Problem Report*.
3. Choose a process route template (*Start Workflow at Snapshot Creation*).

3.4.2 Map DM Change Object to SAP Change Record

In this Customizing activity, you maintain all necessary definitions to map the change object types from the Meta Domain Model to the respective SAP change record types.

Note

Change object integration can handle one SAP system to one external system integration (1:1) and several external PLM systems to one SAP system (N:1) integration. For more information, see [Define External System \[page 24\]](#).

Note

Flex workflows are not supported for change records that are created through the integration. If you need to use flex workflows, you need to create the change record manually.

We recommend that you follow the configuration steps in the given order.

Prerequisites

- You have made all relevant settings in Customizing for [Domain Model Code Lists \[page 11\]](#). You can add additional values as needed before you start mapping or during the mapping process.
- Scenarios and versions have been configured in the system, in Customizing for [Define Scenario and Version \[page 17\]](#).

- You have maintained a change master (CM) profile in Customizing for [Maintain Profile](#) under ► [Cross-Application Components](#) ► [Engineering Change Management](#) ▾.
- The integration pattern for the relevant change object type is set to `SAP_COasChangeRecord`. For more information, see [Integration Patterns \[page 19\]](#).

Note

The following customer-specific integration patterns are not supported for the Change Record outbound integration process in PLM system integration (PLMSI) 3.0 FPO2:

- `SAP_PVasCustomerImplementation` for Product Version
- `SAP_DVasCustomerImplementation` for Document Version
- `SAP_COIasCustomerImpl` for Customer Order Item
- `SAP_COIPsasCustomerImpl` for Customer Order Item Product Structure

Activities

1. Select a [Scenario ID](#) and [Scenario Version](#) to specify the scenario ID and version for which you would like to define and map change object data. Choose [Continue](#).
2. [Define Mapping of Change Object Type](#): Here, you map a Domain Model change object type to the SAP change record data.
Choose the following:

- Change record type, the name of the person responsible for the change record, mark the Default Outbound Type checkbox if this CO type should be used as the default CO type for change object outbound integration, and choose a CM (Change Master) profile and if the outbound communication ID mode will be defined by the SAP system or the external PLM system.

You can also set a valid-from date, the solution item relevance, and choose a process route template ([Start Workflow at Snapshot Creation](#)).

Note

The [Start Workflow at Snapshot Creation](#) feature is only available as of SAP S/4HANA version 2020 FPS02.

You can maintain a change master profile in the Customizing activity [Maintain Profile](#) under ► [Cross-Application Components](#) ► [Engineering Change Management](#) ▾.

When you integrate the following objects, ensure that the profile enables ECN usage as outlined in the following table:

Object Type Profile

ECN Object Type	DM Object Type
Material BOM	SV
Task List	POL
Document	DV

ECN Object Type	DM Object Type
Material	PV with revision
Characteristics of Class	Variant Model
Object Dependencies	SV/POL with selection conditions
Variant Table	Variant Model

For mapping during inbound processing (sending from an external PLM system to the SAP system), choose the following:

- *Change Record Type* specifies the type for change record. You can maintain change record types in Customizing for *Logistics – General* under ► *Product Lifecycle Management (PLM)* ► *Change Records* ► *Settings for Engineering Changes* ► *Define Engineering Change Record Types* .
- *Change Record Responsible* defines the Business Partner ID of the person responsible for the change record if no Business Partner ID can be determined based on information from the external PLM system.

For mapping during outbound processing (sending from the SAP system to an external PLM system), choose the following:

- *Outbound Communication ID Mode* defines how the change object ID is determined.
- *Solution Item Relevance* indicates why the item is included in the change record. You can maintain relevance types in Customizing for *Logistics – General* under ► *Product Lifecycle Management (PLM)* ► *Change Records* ► *Settings for Engineering Changes* ► *Define Item Relevance* .

Set a default *Valid-From* date. This date is used if the external system does not provide a valid-from date. If you leave this field empty, the system uses a fallback date. For more information on the valid-from date, see [Define Scenario and Version \[page 17\]](#).

3. *Define Change Object Status*: Here, you define a combination of Domain Model status, maturity code, and SAP change record status. In case of a change, the change object status is modified as set here. Select the conditions under which the update happens.
 - Use the *Outbound-Relevant* checkbox to define the relevant SAP status to DM status and DM maturity mapping in case of outbound maintain and inbound read requests
 - Use the *Snapshot-Relevant* checkbox to define the CO statuses for which snapshot creation is triggered. The change record status is taken from the Customizing activity ► *Logistics - General* ► *Product Lifecycle Management (PLM)* ► *Change Records* ► *User Statuses* ► *Define User Statuses* . The *User Status* field `USER_STATUS_01` Field needs to be active. Entering data into this field sets the SAP status of a change record based on the DM status code and DM maturity code sent from the external system.

ⓘ Note

These settings are only applicable for the change record-related snapshot creation, which occurs when a message of the CO as CR type is received. A manufacturing snapshot can also be created as a result of processing a POL as PEO MPD message, where these settings are not checked.

4. *Define Inbound Field Mapping* and *Define Outbound Field Mapping*:

Use field mapping to decide how fields from the source structure are transferred to fields in the target structure. You can use [F4](#) help to find the respective values. The help provides all fields, including fields from the structure and included structures. To define field mapping, you do the following:

- Choose *Create*, *Update*, or *Modify* in the column *MapMode*, to define what is valid for this line. (*Modify* includes *Create* and *Update*).
- Define an SAP target structure and SAP target attribute to define which field in the SAP system is to be set. If the *Overwrite* checkbox is selected, the system uses the stored value instead of a value from the payload. This checkbox is for information purposes only and the setting can't be changed.
- Define a Domain Model source structure (you can also use <EXTENSION_FIELDS>) and Domain Model source attribute to define which data will be mapped to the target system.
- Define a conversion in case you want to transform the inbound data before setting it to the target field. You can use preconfigured conversions, and conversions that are specific to your customer or target data. For more information, see [BAdI: Define Custom Field and Value Mapping Conversions \[page 109\]](#).
- Define a default value if you want to set the standard value in case no data is received in the source field or if no source field is mapped. Note that you must define a value in internal format.

Note

If you expect partial payloads, make sure you maintain the settings for *FVM Default Value Mode*.

Field and value mapping supports customer field extensibility by extension fields.

You can use <EXTENSION_FIELDS> to map extension fields that are sent from the external PLM system to a standard field or classification in the SAP system. The name of the extension field must be passed to the source attribute field.

The assignment is done using standard technical conversion rules. You can define a domain value mapping to do data conversions with tables. You can also define a conversion function. The system offers a set of standard conversion functions. You can extend these functions using a BAdI. For more information, see [BAdI: Define Custom Field and Value Mapping Conversions \[page 109\]](#). If a data type other than *DATS* is used, use the *Timestamp* conversion function to map date information from a text field into a timestamp in the SAP system. This enables the conversion between the long timestamp used in the Meta Domain Model and the SAP format.

5. **Define Inbound Class Assignment:** Here, you can assign classes and class types that are added to a change object when it's created and updated, when the change object is sent from the external PLM system to the SAP system. Choose class type 106 to assign a classification to an SAP change record. You can choose a conversion routine (if needed), set default values, or set a value mapping for the value transfer. You can also use these class assignments as a mapping target. You can map to an SAP characteristic (target) and define which Domain Model source field from the Domain Model input structure will be used, for example, to map a status to a characteristic.
 - **Define Inbound Characteristics Mapping:** Here, you define the value mapping for the characteristics of the class. You map values of the Domain Model characteristics to the SAP target fields and extension fields of the change object.
6. **Define Outbound Class Assignment:** Here, you can assign classes and class types that are added to a change object when it's created and updated, when the change object is sent from the SAP system to the external PLM system. You can choose a conversion routine (if needed), set default values, or set a value mapping for the value transfer. You can also use these class assignments as a mapping source. You can map a characteristic and define which DM target field in the DM target structure will be used, for example, to map a characteristic to a status.
 - **Define Outbound Characteristics Mapping:** Here, you define the value mapping for the characteristics of the class. You map values of the characteristics to the Domain Model target fields and extension fields of the change object.

7. **Define Relation Type:** Here, you define which reference type will be used in the SAP change record for the Domain Model relation type of the change order.

Note

General object relevance for outbound integration is defined in [Define Rule Group \[page 25\]](#).

The fields *Domain Model Relation Type*, *Object Type*, and *Relevance* are mandatory. Only the following object types are supported: BOM, DOC, EBM, MAT, MBM, and OBM. Choose the relevance *MPS* for manufacturing snapshots and *SNP* for engineering snapshots.

The *(Change Record Relation Type) Priority* defines the priority for adding the same reference item or solution item to a single change record multiple times, each with different relevance values. It also defines if the message type of the change record type is set to *Error* or if it is empty: As of SAP S/4HANA 2023, it is possible to add the same reference item or solution item to one change record several times with different relevance values.

- If multiple items with the same key and different relevance values are expected, you don't need to enter a priority to distinguish different relevance values when adding the same object to a change record several times. If you want to store all incoming items, the message type of the change record type needs to be *Information* or *Warning*.
- If the message type of the CR type is set to *Error* or if it is empty, you need to enter a *Priority* to define the saving behavior: The record with the highest priority is saved.

The message type of change record types can be set in [Logistics - General > Product Lifecycle Management > Change Records > Define Engineering Change Record Types](#). Here, you can set the type of message that the system shows if a reference object is already used by another change record. In case of such a parallel change, the message types *Information*, *Warning*, and *Error* are available and can be assigned to a change record type. For more information, search for *Change Record Types* in the [SAP Product Help for SAP S/4HANA](#).

Send Referenced Object during Outbound: Select the checkbox to determine if another referenced object of the marked type will be sent as a separate message to the external PLM system during CO outbound maintain requests with referenced COIPS (only valid for 1:1 integrations of one external system with one SAP system (*CO Default Destination* checkbox is selected in [Define External System \[page 24\]](#))).

8. **Define Conditions for Outbound:** You can use the fields *Change Object as CR Table Name* and *Attribute Name* to define which fields in the change record tables trigger a data transfer to the external PLM system when they are updated in the SAP system.
 - **Rule Group ID:** Defines a group of rules. Based on the rule group ID, the system determines the correct target systems for outbound data transfer. Target systems can be defined in the configuration activity [Define Rule Group \[page 25\]](#).
 - **Counter for Outbound Attribute:** Enter a number in the field. You can maintain multiple entries to the same combination of Change Object Table Name and Attribute to specify different values.
 - **Value To:** If the configuration contains a value in this field, the change of the change record must set this value for the defined field to be valid.
 - **Value To Any:** If this checkbox is selected, the actual original value is ignored, and only a change in the attribute is considered relevant.

3.4.3 BAdI: Change Object as CR Responsible

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to change the mapping of a change object (CO) stakeholder to an SAP business partner.

```
/PLMS/BD_CR_RESPONSIBLE
```

Standard Settings

This BAdI offers a default implementation which is called at runtime but can be overridden.

For more information about the standard settings (filters, single or multiple uses), see the Enhancement Spot Element Definitions tab in the BAdI Builder (transaction [SE18](#)).

Activities

You need a custom field mapping, for example, the field *Labor* for a product version. You want to create an implementation that derives the value using a custom table access.

1. Create the implementation of the enhancement spot.
2. Activate your implementation.

3.4.4 BAdI: Determine External System for CR as CO Outbound Scenario

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to determine target systems that are relevant for CR outbound integration.

```
/PLMS/BD_CR_TGT_SYST_DET
```

This BAdI offers a default implementation that is called at runtime but can be overridden. Systems that you define as target systems will receive changes that affect change records in the SAP system. Changes related to the main ECN are also considered.

The BAdI returns target systems for which the following conditions are met:

- the change object as change record DM type is uniquely determined from the key mapping or the L1 mapping configuration in [Map DM Change Object to SAP Change Record](#)
- the scenario configuration contains the [SAP_COasChangeRecord](#) integration pattern
- the changes are applied to the change record or a date shift is applied to the main ECN of the change record, the [Change Record Status](#), [Authorization Group](#) or [Change Number](#) values are used to check the outbound relevance of the change record according to the values specified for the [Value To](#) attribute in [Define Conditions for Outbound](#) with the corresponding rule group ID.

For more information about the standard settings (filters, single or multiple uses), see the [Enhancement Spot Element Definitions](#) tab in the BAdI Builder (transaction [SE18](#)).

Activities

1. Choose [Create](#). Create the implementation of the enhancement spot.
2. Adapt the data from the template as required.
3. Save your entries.

3.4.5 BAdI: Filter Relevant CR for CR as CO Outbound Scenario

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to specify the change records (CR) that are sent as change objects (CO) to an external PLM system.

/PLMS/BD_OUTBOUND_CO_CR

Note

This BAdI is used for 1:1 system integration only.

Standard Settings

This BAdI returns CRs:

- For which the Change Object DM object type is uniquely determined from the L1 mapping configuration in [Map DM Change Object to SAP Change Record](#).
- For which the status is in the list of relevant statuses specified in the [Value To](#) attribute of the [Define Conditions for Outbound](#) view.
- For which change items refer to objects integrated by PLMSI, for example, document version or product version.

For more information about the standard settings (filters, single or multiple uses), see the Enhancement Spot Element Definitions tab in the BAdI Builder (transaction [SE18](#)).

Activities

1. Create the implementation of the enhancement spot.
2. Adapt the data from the template as required.

3.4.6 BAdI: CO as Change Record Post-Processing

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to enhance the SAP integration with customer-specific steps after change record objects processing has been performed successfully for inbound maintain request types.

```
/PLMS/BD_COCR_CUST_POSTPRC_STP
```

This BAdI can raise messages that influence the overall status. Additional messages are added to the existing messages and are written to the logs after the SAP process flow. The status is set accordingly based on the error category.

Note

You cannot use this BAdI to change the payload.

This BAdI has no default implementation. For more information about the standard settings (filters, single or multiple uses), see the *Enhancement Spot Element Definitions* tab in the BAdI Builder (transaction SE18).

Activities

1. Choose *Create*.
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.4.7 BAdI: CO as Change Record Post-Processing - Read Extension Fields

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to enhance the SAP integration with customer-specific steps after all change object processing has been performed for inbound read request types.

```
/PLMS/BD_CO_CR_READ_EXTN_FLD
```

This BAdI can read extension fields to add into the outbound data. Additional extension fields are added to the existing ones.

Note

You cannot use this BAdI to change the payload or process structure.

For more information about the standard settings (filters, single or multiple uses), see the *Enhancement Spot Element Definitions* tab in the BAdI Builder (transaction SE18).

Activities

1. Choose [Create](#).
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.5 DM Document Version to SAP Document Info Record

This node groups several activities for mapping document versions to SAP document info records.

3.5.1 Define File Redirect Settings

Define the file redirect service URL for each client of the system.

In this Customizing activity, you configure the file redirect service URL for each client of the system. Make sure that the service URL ends with the redirect service name.

ⓘ Note

To avoid accessing potentially unsafe or unforeseen resources, it's essential to authenticate URLs through an allowlist to permit or block different functional areas (for example, redirects). Therefore, additional SAP standard configuration is required to manage allowlists. URL validation using an allowlist is used in the following instances:

- Determining a link to a redirect service in the redirect service customizing
- In a redirect service logic after receiving a link from an external system.

If no allowlist is configured in the system, all links are accepted as valid. For information about how to set up an allowlist, see [Manage HTTP Allowlists](#).

You can run the report [Update File Redirect Service Path URL – /PLMS/DV_FILE_URL_UPDATE](#) to update all existing dynamic URL files with the current File Redirect Service path URL. For more information, see [Housekeeping Activities](#).

3.5.2 Define Mapping for MIME Types

Map the Domain Model MIME types to the file extensions for PLM.

You can enter a workstation application and an SAP storage category. Use the [Outbound-Relevant](#) checkbox to define the mapping in case of outbound requests.

3.5.3 Map DM Document Version to SAP Document Info Record

In this Customizing activity, you make the required settings for the document version integration.

You define the mapping of Domain Model document version types to the SAP document types. You can also define the required details such as semantic file types, status, and maturity codes. The bidirectional [Document Version File Redirect Service](#) lets you handle inbound and outbound file transfer through dynamic URLs instead of file downloads, meaning you can display or download document version files (DVs) through static or dynamic URLs. The DV files themselves remain in the source system.

Note

To avoid accessing potentially unsafe or unforeseen resources, it's essential to authenticate URLs by using an allowlist to permit or block different functional areas (for example, redirects). Therefore, additional SAP standard configuration is needed to manage allowlists. URL validation using an allowlist is used in the following instances:

- Determining a link to a redirect service in the redirect service customizing
- In a redirect service logic after receiving a link from an external system

If no allowlist is configured in the system, all links are accepted as valid. For information about how to set up an allowlist, see [Manage HTTP Allowlists](#).

You can use the SAP Document Management Service on BTP for secure, efficient file management in the cloud. It provides the logic to send x access tokens, allowing file access during outbound scenarios such as sending document maintenance requests from SAP to external systems.

We recommend that you follow the configuration steps in the given order.

Prerequisites

- You have made all relevant settings in Customizing for [Domain Model Code Lists \[page 11\]](#). You can add additional values as required before you start mapping, or during the mapping process.
- Scenarios and versions have been configured in the system, in Customizing for [Define Scenario and Version \[page 17\]](#).
- You have maintained a change master (CM) profile in Customizing for [Maintain Profile](#) under **Cross-Application Components** > [Engineering Change Management](#).
- You have implemented one of the following content management strategies:
 - [SAP Content Server](#). For information about how to access the SAP Content Server over HTTPS (we recommend using HTTPS, not HTTP), see SAP Note [712330](#). For the subsequent introduction of the Security Level for Documents (URL Signature), see SAP Note [792366](#). For the general setup of the SAP Content Server, go to the [SAP Help Portal](#) and search for [SAP Content Server](#). Make sure to use the latest version of the documentation.
 - [SAP Document Management Service on BTP](#). For information on how to configure the SAP Document Management Service on BTP, see [How to configure SAP Document Management Service on BTP](#) (no official SAP Help Portal documentation). You must also implement SAP Note [3246481](#) (The information in the note also applies to the PLMSI use case).

- The integration pattern for the relevant document version type is set to `SAP_DVasDocumentInfoRecord`. For more information, see [Integration Patterns \[page 19\]](#).

Note

The customer-specific integration pattern `SAP_DVasCustomerImplementation` is not supported for the Change Record outbound integration process in PLM system integration (PLMSI) 3.0 FP02.

- You have defined MIME type mappings in the configuration activity [Define Mapping for MIME Types \[page 42\]](#).

Activities

1. Select a *Scenario ID* and *Scenario Version* to specify the scenario ID and version for which you'd like to define structure version data.
2. *Define Mapping of Document Type*: Map a Domain Model document type to the SAP document type.
 - *CAD Indicator*: Sets the CAD indicator in the document info records for documents of this type to indicate that a document is managed by CAD integration.
 - *Version Schema for Document* indicates how version numbers are assigned for the document. Select *External numbering* to use the version number from the payload, or *Internal numbering* for the version number to be generated internally.
If you want to use internal numbering, you need to configure the system to use internal numbering and internal versioning. To do so, go to [Cross-Application Components > Document Management System > Define Document Types](#) (transaction `DC10`). Choose a document type, or create a new type if needed. Set the *Number Assignment* to *Only internal number assignment* and select *Version Assigned Automatically*.
 - *Document Version Revision Level* indicates that a revision level is automatically assigned to a document revision. You can set this only if *Version Schema for Document* is set to *Internal numbering*.
 - *Document Version Attachment*: Determines if the document of the corresponding Domain Model document type can be used as an attachment to a change object:
 - **Activated**: The document version is created as an attachment to the change object.
 - **Deactivated**: The document version is created but not attached to the change object.

Note

Documents that are used as attachments cannot be used as related/solution items by the integration. Example: A change object has a related item and an attachment, both are represented as the same DIR. In this case, only the attachment is integrated, and the related item is skipped during processing.

- *Workstation Application Dynamic URL*: Defines the workstation application that's used for receiving a dynamic URL during inbound file processing. The SAP system will use the created URL file to display the respective file instead of downloading it from an external system. If this field is empty, handling of dynamic URLs is not possible.

Note

The defined workstation application must contain settings that are relevant to create URL files, such as file format, MIME type, and so on.

Workstation Application Static URL:

Defines the workstation application that's used for receiving or sending a static URL during inbound and outbound file processing. If this field is empty, handling of static URLs is not possible.

Note

URLs are stored in a DMS table that can only be updated from Fiori. In the DVF Display GUI, you can only display the data. The static URL use case is only suitable for stable URLs. It is not adequate for storing references to documents that are stored on external shares or PDM systems. This is mainly due to possible server changes in these systems, especially in cloud setups. Any server changes can lead to broken URLs that can't be fixed.

You can run the report [Update File Redirect Service Path URL – /PLMS/DV_FILE_URL_UPDATE](#) to update all workstation applications used for dynamic URLs with the current File Redirect Service path URL. For more information, see [Housekeeping Activities](#).

- For data federation, select how you want to display document version files through [DVF Display \(GUI or Fiori\)](#).
3. **Define Document Status:** Here, you define a combination of status and maturity code. If there is a change, the document status is modified as set here. Select under which conditions the update happens. Use the [Outbound-Relevant](#) checkbox to define the relevant SAP status to DM status & DM maturity mapping in case of outbound requests.
 4. **Define Inbound Field Mapping and Define Outbound Field Mapping:**
Use field mapping to decide how fields from the source structure are transferred to fields in the target structure. You can use [F4](#) help to find the respective values. The help provides all fields including fields from the structure and the included structures. To define field mapping, do the following:
 - To define what is valid for this line, choose [Create](#), [Update](#), or [Modify](#) in the column [MapMode \(Modify includes Create and Update\)](#).
 - Define an SAP target structure and SAP target attribute to define which field in the SAP system will be set. Choose [F4](#) to see the available values. If the [Overwrite](#) checkbox is selected, the system uses the stored value instead of a value from the payload. This checkbox is for information purposes only and the setting can't be changed.
 - Define a Domain Model source structure (you can also use `<EXTENSION_FIELDS>`) and Domain Model source attribute to define which data will be mapped to the target system.
 - Define a conversion in case you want to transform the inbound data before setting it to the target field. You can use preconfigured conversions, and conversions that are specific to your customer or target data. For more information, see [BAId: Define Custom Field and Value Mapping Conversions \[page 109\]](#).
 - Define a default value if you want to set the standard value in case no data is received in the source field or if no source field is mapped. Note that you must define a value in internal format.

Note

If you expect partial payloads, make sure to maintain the settings for [FVM Default Value Mode](#).

- The [Value Mapping ID](#) identifies a set of mapping values.

Field and value mapping supports customer field extensibility by extension fields.

You can use `<EXTENSION_FIELDS>` to map extension fields that are sent from the external PLM system to a standard field or classification in the SAP system. The name of the extension field must be passed to the source attribute field.

The assignment is done using standard technical conversion rules. You can define a domain value mapping in order to do data conversions using tables. You can also define a conversion function. The system offers

a set of standard conversion functions. You can extend these functions using a BAdI. For more information, see [BAdI: Define Custom Field and Value Mapping Conversions \[page 109\]](#). If a data type other than *DATS* is used, use the *Timestamp* conversion function to map date and time information from a text field into a timestamp in the SAP system. This enables the conversion between the long timestamp used in the Meta Domain Model and the SAP format.

5. **Define Class Assignment:** Class Assignments let you map Domain Model attributes or values to SAP classifications and store this mapping. Here, you can assign class numbers and class types that will be added to a document info record when it's created and updated. You can select a class to allocate to the SAP object during its creation or update. In a second step, you define the characteristic that will be the target of the mapping.
 - **Define Characteristics Mapping:** Here you can choose a conversion routine (if needed), set default values, or set a value mapping for the value transfer. You can also use these class assignments as a mapping target. You can map to a characteristic (target) and define which source field from the input structure will be used, for example, to map a status to a characteristic.
6. **Define Conditions for Outbound:** You can use the fields *Document Table Name* and *Attribute Name* to define which fields in the document master tables trigger a data transfer to the external PLM system when they are updated in the SAP system.
 - **Rule Group ID:** Defines a group of rules. Based on the rule group ID, the system determines the correct target systems for outbound data transfer. Target systems can be defined in the configuration activity [Define Rule Group \[page 25\]](#).
 - **Value From:** If the configuration contains a value in the *Value From* field, the original value is ignored, and only a change in the attribute is considered relevant.
 - **Value From Any:** If *Value From Any* is selected, the actual original value is not relevant, only the fact that the attribute was changed.
 - **Value To:** If the configuration contains a value in the *Value To* field, the change must set in order for this value in order for the defined field to be valid.
 - **Any Value To:** If *Value To Any* is selected, the actual target value is ignored, and only a change in the attribute is considered relevant.
 - Enter a number in the *Counter for Outbound Attribute* field. You can maintain multiple entries for the same combination of *Document Table Name* and *Attribute* to specify different values.
7. **Define Workstation Application for File as URL Outbound:** Here, you define to which workstation application the files are to be sent as URLs during outbound processing. If the *URL Active* checkbox is marked, the external system does not upload the respective document and instead uses the URL for document display. See also [BAdI: Control DV File Definition \[page 46\]](#).

3.5.4 BAdI: Control DV File Definition

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI during DV outbound transfer. It controls whether files will be physically sent to the external PLM system or if a dynamic URL will be used to access the files from the external PLM system without file transfer.

```
/PLMS/BD_DV_FILE_BY_URL_DETN
```

Note

To avoid accessing potentially unsafe or unforeseen resources, it's essential to authenticate URLs using an allowlist to permit or block different functional areas (for example, redirects). Therefore, additional SAP

standard configuration is required to manage allowlists. URL validation using an allowlist is used in the following instances:

- Determining a link to a redirect service in the redirect service customizing.
- In a redirect service logic after receiving a link from an external system.

If no allowlist is configured in the system, all links are accepted as valid. For information about how to set up an allowlist, see [Manage HTTP Allowlists](#).

During version document transfer, you can either send the respective files to the external PLM system, or you can provide a dynamic URL for displaying the files in the external PLM system without file transfer.

SAP provides a default implementation that reads the Customizing data and determines whether a file will be sent as a URL or downloaded to the external PLM system. For more information about Customizing, see [Map DM Document Version to SAP Document Info Record \[page 43\]](#).

For more information about the standard settings (filters, single or multiple uses), see the Enhancement Spot Element Definitions tab in the BAdI Builder (transaction [SE18](#)).

BAdI Settings

This is a single-use BAdI.

Activities

1. Choose [Create](#).
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.5.5 BAdI: Determine External System for DV Outbound Scenario

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to determine target systems that are relevant for document version outbound integration.

```
/PLMF/BD_DOC_TGT_SYST_DETN
```

This BAdI verifies if a system is relevant for outbound integration and returns a list of relevant target systems:

- Target systems for which the sent document version type is marked as outbound-relevant in Customizing for [Map DM Document Version to SAP Document Info Record](#) [Define Mapping of Document Type](#).
- Target systems for which changes have been made to a sent document.

This BAdI offers a default implementation that is called at runtime but can be overridden. Systems that you define as target systems will receive changes that affect document versions in the SAP system.

For more information about the standard settings (filters, single or multiple uses), see the [Enhancement Spot Element Definitions](#) tab in the BAdI Builder (transaction [SE18](#)).

Activities

1. Choose [Create](#). Create the implementation of the enhancement spot.
2. Adapt the data from the template as required.
3. Save your entries.

3.5.6 BAdI: Document Version as DIR Setting for SAP Key

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to modify the SAP document version key in order to have control over documents to be created.

```
/PLMF/BD_DV_SET_SAP_KEY
```

The SAP document version key consists of document type, document number, document version, and document part. It is derived from the business ID.

The BAdI implementations can be filtered by the document version type received in the messages from external systems.

Prerequisites

The SAP document ID has been derived from the business ID as part of the incoming data flow of document version creation.

Activities

1. Choose [Create](#).
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.5.7 BAdI: DV as DIR Post-Processing

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to enhance the SAP integration with customer-specific steps after document version objects processing has been performed successfully for inbound maintain request types.

```
/PLMF/BD_DV_CUST_POSTPROC_STEP
```

This BAdI can raise messages that influence the overall status. Additional messages are added to the existing messages and are written to the logs after the SAP process flow. The status is set accordingly based on the error category.

Note

You cannot use this BAdI to change the payload.

This BAdI has no default implementation. For more information about the standard settings (filters, single or multiple uses), see the *Enhancement Spot Element Definitions* tab in the BAdI Builder (transaction SE18).

Activities

1. Choose *Create*.
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.5.8 BAdI: DV as DIR Post-Processing – Read Extension Fields

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to enhance the SAP integration with customer-specific steps after document version objects processing has been performed successfully for inbound read request types.

```
/PLMF/BD_DV_READ_EXTN_FLD
```

This BAdI can read extension fields to add into the outbound data. Additional extension fields are added to the existing ones.

Note

You cannot use this BAdI to change the payload.

For more information about the standard settings (filters, single or multiple uses), see the *Enhancement Spot Element Definitions* tab in the BAdI Builder (transaction SE18).

Activities

1. Choose *Create*.
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.5.9 BAdI: FC as File Upload Confirmation Post-Processing

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to enhance the SAP integration with customer-specific steps after file upload confirmation objects processing has been performed successfully for inbound maintain request types.

```
/PLMF/BD_FC_CUST_POSTPROC_STEP
```

This BAdI can raise messages that influence the overall status. Additional messages are added to the existing messages and are written to the logs after the SAP process flow. The status is set accordingly based on the error category.

Note

You cannot use this BAdI to change the payload.

This BAdI has no default implementation. For more information about the standard settings (filters, single or multiple uses), see the *Enhancement Spot Element Definitions* tab in the BAdI Builder (transaction SE18).

Activities

1. Choose *Create*.
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.6 DM Factory to SAP Plant

This node groups activities for mapping factories to SAP plants.

3.6.1 Map DM Factory to SAP Plant

In this Customizing activity, you assign a factory to an SAP plant.

You can use this assignment during product configuration as part of plant-specific material configuration, and also in structure version and production operation list mapping, where factory data is mapped to the corresponding SAP plant for the creation of a BOM or routing.

You can assign a combination of Domain Model factory ID and type to an SAP plant. The assignment enables you to set a plant-specific status for this assignment in the activity [Map DM Product Version to SAP Material Master \[page 63\]](#). When you create a product in a specific plant, the status will be mapped according to this assignment. You can select the maintained plant-specific data in field mapping dialogues, for example in [Define Inbound Field Mapping](#) for a product version.

We recommend that you follow the configuration steps in the given order.

Prerequisites

- You have made all relevant settings in the Customizing for [Domain Model Code Lists \[page 11\]](#), including a sourcing strategy code. You can add additional values as required before you start mapping, or during the mapping process.
- Scenarios and versions have been configured in Customizing for [Define Scenario and Version \[page 17\]](#).

Activities

1. Select a [Scenario ID](#) and [Scenario Version](#) to specify the scenario and version for which you would like to define a factory.
2. [Define Factory Assignment](#): Here, you assign the factory to an SAP plant. Choose the Domain Model factory ID and the Domain Model factory type. Then, choose the plant you want to assign. Select the [Outbound-Relevant](#) checkbox if you want to send the data for the factory assignment to an external PLM system during outbound processing. Subsequently, you can define a plant-specific status and template material, and define inbound field mapping for plant data in [Map DM Product Version to SAP Material Master \[page 63\]](#).

Note

You can enter and save only one record for each combination of [Plant](#) and [Outbound-Relevant](#).

3. [Define Work Area and Work Center](#): Here, you can map incoming work area data to a work center in the SAP system. You can use the Domain Model work area ID and type to assign the work center.

3.7 DM IC to SAP Master Inspection Characteristic

This node groups several activities for mapping inspection characteristics to SAP master inspection characteristics.

3.7.1 Map DM IC to SAP Master Inspection Characteristic

In this configuration activity, you can define essential control parameters for inspection characteristic management. We recommend that you follow the configuration activities below in the given order.

Prerequisites

- Domain Model code lists have been defined in the system.
- Scenarios and versions have been configured in configuration activity [Define Scenario and Version \[page 17\]](#).
- The integration pattern for the relevant inspection characteristic is set to *SAP_ICasMasterInspChar*. For more information, see the chapter **Integration Patterns** in the Configuration Guide for PLM system integration for SAP S/4HANA on the SAP Help Portal.
- You have familiarized yourself with the contents of SAP Note [3257849](#): *Creating initial characteristic value without value description*.

Activities

1. Select a *Scenario ID* and *Scenario Version* to specify the scenario and version for which you would like to define the IC data.
2. **Define Mapping of Insp Char Type:** Here, you map a Domain Model inspection characteristic type to the master inspection characteristic type. Confirm and save your selections.
 - Choose an Inspection Characteristic Type to classify the inspection characteristic according to its functionality.
 - In *Default Plant*, you can specify the plant to be set by default for an inspection characteristic.
 - In *Default Authorization Group*, enter a default authorization group to enable authorization protection for extended table maintenance.
3. **Define Inspection Characteristic Status:** Here, you define a combination of status and maturity code. In case of a change, the IC status will be modified as set here. Select the conditions under which the update happens.
4. **Define Inbound Field Mapping:** Here, you use field mapping to decide how fields from the source structure are transferred to fields in the target structure. You can use F4 help to find the respective values. The help provides all fields including fields from the structure and included structures. To define field mapping, you do the following:

- Choose *Create*, *Update*, or *Modify* in the *MapMode* column to define what is valid for this line. (*Modify* includes *Create* and *Update*).

Define an SAP target structure and SAP target attribute to define which field in the SAP system you want to set.

SAP Target Attribute: Choose *F4* to see the available values. If the Overwrite checkbox is selected, the system uses the stored value instead of a value from the payload. This checkbox is for information purposes only and the setting can't be changed.

- Define a Domain Model source structure and Domain Model source attribute to define which data you want to map to the target system.
 - Define a conversion in case you want to transform the inbound data before setting it to the target field. For more information, see [BADl: Define Custom Field and Value Mapping Conversions \[page 109\]](#).
 - Define a default value if you want to set the standard value, in case no data is received in the source field or if no source field is mapped. Note that you must define a value in an internal format. If you expect partial payloads, make sure you maintain the settings for *FVM Default Value Mode*.
 - The *Value Mapping ID* defines a set of mapped values.
5. **Define Crit. Code and Weighting of Characteristic**: Here, you can assign an SAP weighting value (for example 01) to a received DM criticality code value (for example, low). You can use this weighting value during inspection characteristic maintenance. Confirm and save.

Note: You can create additional criticality codes in [Define Domain Model Criticality Code \[page 12\]](#).

3.7.2 Maintain Code Group Number Range

Make the required settings for the number ranges used for generating unique values for code groups in order to restrict available values.

Prerequisites

At least one interval should be created. Without an interval for number ranges, codes and code groups cannot be created.

Activities

1. *Display Intervals*: Here, you can see the existing intervals.
2. *Change Intervals*: Here, you can add a new interval, and change or delete existing intervals for a number range.
3. *Change NR Status*: Here, you specify the status for a number range.

3.7.3 Maintain Selection Set Number Range

Make the required settings for the number ranges used for generating unique values for selection sets in order to restrict available values.

Prerequisites

At least one interval should be created. Without an interval for number ranges, selection sets cannot be created.

Activities

1. *Display Intervals*: Here, you can see the existing intervals.
2. *Change Intervals*: Here, you can add a new interval, and change or delete existing intervals for a number range.
3. *Change NR Status*: Here, you specify the status for a number range.

3.7.4 BAdI: IC as Master Inspection Characteristic Post-Processing

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to enhance the SAP integration with customer-specific steps after inspection characteristics objects processing has been performed successfully for inbound maintain request types.

```
/PLMF/BD_IC_CUST_POSTPROC_STEP
```

This BAdI can raise messages that influence the overall status. Additional messages are added to the existing messages and are written to the logs after the SAP process flow. The status is set accordingly based on the error category.

Note

You cannot use this BAdI to change the payload.

This BAdI has no default implementation. For more information about the standard settings (filters, single or multiple uses), see the *Enhancement Spot Element Definitions* tab in the BAdI Builder (transaction SE18).

Activities

1. Choose [Create](#).
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.8 DM POL to SAP Classic Routing

This node groups several activities for mapping production operation lists to SAP routings.

3.8.1 Map DM POL to SAP Classic Routing

In this Customizing activity, you make the required settings for routing integration. You define the mapping of Domain Model production operation list types to the SAP product types and routing types. In addition, you can define essential control parameters for routing management. We recommend that you follow the configuration steps in the given order.

Prerequisites

- You have made all relevant settings in the Customizing for [Domain Model Code Lists \[page 11\]](#). Domain Model code lists have been defined. You can add additional values as needed before you start mapping, or during the mapping process.
- Scenarios and versions have been configured in Customizing activity [Define Scenario and Version \[page 17\]](#).
- You have maintained a change master (CM) profile in Customizing for [Maintain Profile](#) under ► [Cross-Application Components](#) ► [Engineering Change Management](#) ▾. Most objects that are received from the external PLM system include a valid-from date. The SAP system uses this date in combination with the data from the change master profile to create the change master for the object. For more information on the valid-from date, see [Define Scenario and Version \[page 17\]](#).
- The integration pattern for the relevant production operation list type is set to `SAP_POLasRouting`. For more information, see [Integration Patterns \[page 19\]](#).

Activities

1. Select a [Scenario ID](#) and [Scenario Version](#) to specify the scenario ID and version for which you would like to define structure version data.
2. [Define Mapping of Production Operation List](#): Here you map a Domain Model production operation list type to the SAP product type and routing type. If components of a classic BOM are enhanced with substitutes,

and the BOM is assigned to a DM production operation list through a DM structure version, at operation level the allocation for this POL also incorporates these substitutes when the POL is maintained again from the external PLM system. Confirm and save your selections.

- Set default *Valid-From* date. This date is used if external system does not provide a valid-from date. The system uses a fallback date if you leave this field empty. For more information on the *Valid-From* date, see [Define Scenario and Version \[page 17\]](#).

Note

For change objects, the valid-from date is used from the change object payload. For production operation list, the valid-from date is used from the production operation payload.

- Choose a *Task List Type* to classify the task list according to its functionality, and a *List Usage*.
 - *Enforce Internal Numbering*: When this checkbox is selected, the internal numbering range will be consumed to generate the routing group ID for the provided external business ID.
 - *Suboperation Mode*: Determines how suboperations received from an external PLM system are handled in the SAP system: For *DM suboperation as suboperation*, the suboperation needs to belong to the same sequence as the parent operation. For *DM suboperation as operation*, the suboperation can belong to a different sequence than its parent operation, just like an independent operation.
 - *Initial Operation/Activity Number*: Sets the first number of the operation.
 - *Operation/Activity Increment*: Sets the increment for automatically increasing the number for the operation.
 - *Effectivity Type*: Here, you can assign an effectivity type to specify that the validity of the changes is determined by the parameters of the effectivity type, instead of by a valid-from date.
 - *Inspection Characteristic Decimal Places*: Displays the number of decimal places for inspection characteristics.
 - *Position Number Determination*: Defines whether the position number is determined by a predecessor (default) or by DM position number.
 - *Parallel Sequence Determination*: To enable parallelization for classic routing operations, you can enable multiple modes:
 - By default, only a standard sequence is created, and all operations use the same sequence (*Default*).
 - Alternatively, you can choose to create a sequence based on the new collection sequence from the request payload (*Sequence-Based*). This supports standard, parallel, and alternative sequences.
 - You can also create or update a sequence based on the predecessor of the provided operations using a heuristic approach (*Heuristic-Based*). This supports standard and parallel sequences.
3. *Define Production Operation List Status*: Here, you define a combination of status and maturity code. In case of a change, the routing status will be modified as set here. Select the conditions under which the update happens.
 4. *Define Inbound Field Mapping*: You use field mapping to decide how fields from the source structure are transferred to fields in the target structure. You can use *F4* help to find the respective values. The help provides all fields including fields from the structure and included structures. To define field mapping, you do the following:
 - Choose *Create*, *Update* or *Modify* in the *MapMode* column to define what is valid for this line. (*Modify* includes *Create* and *Update*).
 - Define an SAP target structure and SAP target attribute to define which field in the SAP system you want to set. You can set the value of your sequences and other attributes through the extension field,

provided that you have first set the type as a sequence (*Sequence-Based*) in the *Define Mapping of Production Operation List*, under *Parallel Sequence Determination*.

- *SAP Target Attribute*: Choose *F4* to see the available values. If the *Override* checkbox is selected, the system uses the stored value instead of a value from the payload. This checkbox is for information purposes only and the setting can't be changed.
- Define a Domain Model source structure and Domain Model source attribute to define which data you want to map to the target system.
- Define a conversion in case you want to transform the inbound data before setting it to the target field. For more information, see [BAdI: Define Custom Field and Value Mapping Conversions \[page 109\]](#).
- Define a default value if you want to set the standard value, in case no data is received in the source field or if no source field is mapped. Note that you must define a value in an internal format.

Note

If you expect partial payloads, make sure you maintain the settings for *FVM Default Value Mode*.

5. *Define DM Parameters Mapping*: Here, you define the mapping of DM unit effectivity to effectivity parameters in the SAP system. You define effectivity parameters in Customizing for *Engineering Change Management* under **► Parameter Effectivity ► Define Parameters ►**.
6. *Define DM Range Parameter Mapping*: Here, you define the mapping of DM unit effectivity to an open interval effectivity parameter in the SAP system. This setting is used for mapping the value of the *unitTo* and *unitFrom* fields from the payload for a production operation list.
7. *Define Crit. Code and Weighting of Characteristic*: Here, you can assign an SAP weighting value (for example 01) to a received DM criticality code value (for example, low). You can use this weighting value during inspection characteristics maintenance. Confirm and save.

Note

You can create additional criticality codes in [Define Domain Model Criticality Code \[page 12\]](#).

3.8.2 Define POL Model Unit Template ECN

In this configuration activity, you can use a model unit ECN template to define default effectivity values for a POL operation.

A model unit template ECN is a template change number that is used to define default effectivity values for an SV component or POL operation. It is used when operations in a product operation list or components in a bill of material (BOM) are received without dedicated model unit effectivity (MUE) but the object is under model unit control.

Prerequisites

- Scenarios and versions have been defined in configuration activity [Define Scenario and Version \[page 17\]](#).
- A mapping of production operation list type to a Domain Model product type has been defined in the configuration activity *Define Mapping of Production Operation List* in [Map DM POL to SAP Classic Routing \[page 55\]](#).

Activities

1. Select a *Scenario ID* and *Scenario Version* to specify the scenario and version for which you would like to define data.
2. Select *Mapping of Production Operation List* to specify the mapping of a Domain Model production operation list type to a Domain Model product type for which you would like to define a MUE ECN template.
3. Select *Define MUE ECN Template* to specify a model unit template ECN for the selected DM production operation list type and product type.

3.8.3 Maintain Code Group Number Range

Make the required settings for the number ranges used for generating unique values for code groups in order to restrict available values.

Prerequisites

At least one interval should be created. Without an interval for number ranges, codes and code groups cannot be created.

Activities

1. *Display Intervals*: Here, you can see the existing intervals.
2. *Change Intervals*: Here, you can add a new interval, and change or delete existing intervals for a number range.
3. *Change NR Status*: Here, you specify the status for a number range.

3.8.4 Maintain Selection Set Number Range

Make the required settings for the number ranges used for generating unique values for selection sets in order to restrict available values.

Prerequisites

At least one interval should be created. Without an interval for number ranges, selection sets cannot be created.

Activities

1. *Display Intervals*: Here, you can see the existing intervals.
2. *Change Intervals*: Here, you can add a new interval, and change or delete existing intervals for a number range.
3. *Change NR Status*: Here, you specify the status for a number range.

3.8.5 BAdI: VC Selection Condition for POL as Classic Routing

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to modify the selection condition for *production operation list*.

```
/PLMF/BD_POL_SELECTION_COND
```

The BAdI implementation enhances the converted internal format.

Prerequisites

The incoming production operation list data must contain a selection condition in the external format.

Activities

1. Choose *Create*.
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.8.6 BAdI: POL as Classic Routing Post-Processing

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to enhance the SAP integration with customer-specific steps after production operation list objects processing has been performed successfully for inbound maintain request types.

```
/PLMF/BD_POL_CUST_POSTPRC_STEP
```

This BAdI can raise messages that influence the overall status. Additional messages are added to the existing messages and are written to the logs after the SAP process flow. The status is set accordingly based on the error category.

Note

You cannot use this BAdI to change the payload.

This BAdI has no default implementation. For more information about the standard settings (filters, single or multiple uses), see the *Enhancement Spot Element Definitions* tab in the BAdI Builder (transaction `SE18`).

Activities

1. Choose *Create*.
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.9 DM POL to SAP PEO Routing

This node groups several activities for mapping production operation lists to SAP PEO routings.

3.9.1 Map DM POL to SAP PEO Routing

In this Customizing activity, you make the required settings for SAP S/4HANA manufacturing for production engineering and operations (PEO) routing integration.

You define the mapping of production operation list types to the Domain Model type, SAP document type, and SAP document part. You can receive unit effectivity parameters on operation and suboperation level. In the SAP system, you can create snapshots or snapshot revisions to prepare the subsequent steps in the handover process.

Note

Engineering snapshots are supported as of SAP S/4HANA version 2021, manufacturing snapshots are supported only as of SAP S/4HANA version 2022 FPS1.

Based on the POL payload, the integration provides the necessary information for the MPD PIP scenario. For more information, search for *PiP Workspace* in the [SAP Help Portal](#). We recommend that you follow the configuration steps below in the given order.

Prerequisites

- You have made all relevant settings in the configuration for [Domain Model Code Lists \[page 11\]](#). You can add additional values as required before you start mapping or during the mapping process.

- Scenarios and versions have been defined in the system, in the configuration activity [Define Scenario and Version \[page 17\]](#).
- The integration pattern for the relevant POL type is set to `SAP_POIasPEORouting`. For more information, see [Integration Patterns \[page 19\]](#).
- You have maintained values in [Maintain Code Group Number Range \[page 53\]](#).
- You have maintained values in [Maintain Selection Set Number Range \[page 54\]](#).

Activities

1. Select a *Scenario ID* and *Scenario Version* to specify the scenario and version for which you would like to define and map routing data.
2. *Define Mapping of Production Engineering Operation*: Here, you map a *Production Operation List Type* to the SAP *Document Type* and SAP *Document Part* (identifier). You can select the number of decimal places for inspection characteristics. Set the *Processing Mode* to define how the Domain Model object is processed, for example, *Direct processing* or *Staged processing* for asynchronous processing in PEO.

- Direct processing - Creates a DIR that contains inbound information that can be manually verified in the system.
- Staged processing - Creates a DIR that contains inbound information. Once the shop floor routing is created according to the change record process, the external system is notified automatically.

Select an option for the *Automatic Creation of Operation Activity* to enable or disable the automatic creation of operation activities.

- *On* (default setting) – The automatic creation of operation activities is enabled. Activities are created if conditions are met.
- *Off* - The automatic creation of operation activities is disabled. Activities are not created even if conditions are met.

Determine the item type for product and structure version components in work instructions using the fields *PIP Source Item Type for PVs* and *PIP Source Item Type for SV Components*. As a result of the PEO-POL processing, the system generates an XML file. In this file, an `<sap-reference>` tag represents each version, with the *ItemType* attribute reflecting the value specified in the customizing field as a source item identifier.

The received data is passed in XML format and can be used in the *Production Integration Portal (PiP)* in PEO. Confirm and save your selections.

3. *Define Crit. Code and Weighting of Characteristic*: Here, you can assign an SAP weighting value (for example 01) to a received DM criticality code value (for example, low). You can use this weighting value during inspection characteristics maintenance. Confirm and save.

Note

You can create additional criticality codes in [Define Domain Model Criticality Code \[page 12\]](#).

4. *Define Inbound Field Mapping*: Domain Model code lists have been defined in the system, in the Customizing. Use field mapping to decide how fields from the source structure are transferred to fields in the target structure. You can use *F4* help to find the respective values. The help provides all fields including fields from the structure and included structures. To define field mapping, you do the following:
 - Choose *Create*, *Update*, or *Modify* in the column *MapMode*, to define what is valid for this line. (*Modify* includes *Create* and *Update*).

- Define an SAP target structure and SAP target attribute to define which field in the SAP system is set. Choose [F4](#) to see the available values. If the [Override](#) checkbox is selected, the system uses the stored value instead of a value from the payload. This checkbox is for information purposes only and the setting can't be changed.
- Define a Domain Model source structure (you can also use <EXTENSION_FIELDS>, see below) and Domain Model source attribute to define which data will be mapped to the target system.
- Define a conversion in case you want to transform the inbound data before setting it to the target field. The assignment is done using standard technical conversion rules. You can use preconfigured conversions, and conversions that are specific to your customer or target data. Define a domain value mapping to do data conversions using tables. You can also define a conversion function. The system offers a set of standard conversion functions.

🔗 Example

If a data type other than [DATS](#) is used, use the [Timestamp](#) conversion function to map date information from a text field into a timestamp in the SAP system. This enables the conversion between the long timestamp used in the Meta Domain Model and the SAP format.

You can extend these functions through the enhancement spot [/PLMF/ES_FVM_CONVERSION](#). [BAdI: Define Custom Field and Value Mapping Conversions \[page 109\]](#).

- Define a default value if you want to set the standard value, in case no data is received in the source field or if no source field is mapped. Note that you must define a value in internal format.

📌 Note

If you expect partial payloads, make sure you maintain the settings for [FVM Default Value Mode](#).

Field and value mapping supports customer field extensibility by extension fields, for example, to map plant data from product and factory data. You can use <EXTENSION_FIELDS> to map extension fields that are sent from the external PLM system to a standard field or classification in the SAP system. The name of the extension field should be passed to the source attribute field.

3.9.2 BAdI: Determine External System for PEO Routing Outbound Scenario

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to determine target systems that are relevant for PEO routing outbound integration.

```
/PLMS/BD_PEO_SFR_TGT_SYST_DETN
```

The BAdI returns a list of relevant target systems from which the POL was created or updated.

This BAdI offers a default implementation that is called at runtime but can be overridden. Systems that you define as target systems will receive changes that affect product versions in the SAP system.

For more information about the standard settings (filters, single or multiple uses), see the [Enhancement Spot Element Definitions](#) tab in the BAdI Builder (transaction [SE18](#)).

Activities

1. Choose [Create](#). Create the implementation of the enhancement spot.
2. Adapt the data from the template as required.
3. Save your entries.

3.10 DM Product Version to SAP Material Master

This node groups several activities for mapping product versions to SAP material masters.

3.10.1 Map DM Product Version to SAP Material Master

In this Customizing activity, you maintain all necessary definitions to map the product version data from the Meta Domain Model to the respective SAP product version.

The SAP product version represents a material from the material master and its revision. We recommend that you follow the configuration steps in the given order.

Prerequisites

- You have made all relevant settings in Customizing for [Domain Model Code Lists \[page 11\]](#). You can add additional values as needed before you start mapping, or during the mapping process.
- Scenarios and versions have been configured in the system, as shown in Customizing for [Define Scenario and Version \[page 17\]](#).
- You have maintained a change master (CM) profile in Customizing for [Maintain Profile](#) under **► Cross-Application Components** **► Engineering Change Management**. Most objects that are received from the external PLM system include a valid-from date. The SAP system uses this date in combination with the data from the change master profile to create the change master for the object.
- The integration pattern for the relevant product version type is set to `SAP_PVasMaterialMaster`. For more information, see [Integration Patterns \[page 19\]](#).

Note

The customer-specific integration pattern `SAP_PVasCustomerImplementation` is not supported for the Change Record outbound integration process in PLM system integration (PLMSI) 3.0 FPO2.

- The compatibility mode is set as required, see [Define Scenario and Version \[page 17\]](#).

Activities

1. Select a *Scenario ID* and *Scenario Version* to specify the scenario ID and version for which you want to define product version data.
2. *Define Mapping of Product Type*: Here, you map a Domain Model product type to the SAP product type. The *Product Version Derivation Type* determines if the material will be created with or without revision level.

Note

If a material is created with revision level, and a change object is received, the system checks if the revision level can be produced using the change number from the change object. If this is not possible, PLMSI produces an ECN on demand, using the respective Customizing profile.

In *Workflow for Auto UoM*, enter the workflow to be triggered after the creation of an alternative unit of measure (UoM) (instead of the base UoM of the BOM), as specified by *Create Alternative Product Version UoM* under *Define Mapping of Structure Type* in Customizing activity *Map DM Structure Version to SAP Classic BOM* [page 72]. If you use configurable materials, enter data for *Configuration Profile Name* and *Configuration Profile Status*.

The checkbox *MDG Process* determines whether Master Data Governance (MDG) is active.

Master Data Governance (MDG) helps you govern change-related master data in your SAP system. You can set up MDG for PV as material master and as material revision scenarios to ensure the following:

- Material numbers adhere to MDG rules (reservation and creation)
- Used materials follow your company-wide governance process

You need to activate MDG individually per PV type.

Note

MDG can be set up for SAP S/4HANA 2020 or higher. For more information on Master Data Governance, see [How-to Information for SAP Master Data Governance](#) (no official product documentation).

The implementation of [3271481](#) is required to ensure the accurate operation of business validation checks in the Master Data Governance (MDG) for the product version local scenario process.

If you select the *MDG Process* checkbox, you must select a *Type of MDG Request* to define the type of MDG request that is created.

If you select *Only Copy Plants Matching Factory in Payload* in the *Copy Plant from Template Material* field, the system ignores a template material's plant data if this data does not match the PV factory assignment of a DM message. This setting uses a default implementation of [BAdI: Control PV to Material Master Behavior](#) [page 68].

3. *Define Product Status*: Here, you define which combination of status and maturity code will result in which cross-plant product status. Select under which conditions the status is set. Any cross-plant status setting work that you do in the field and value mapping will overwrite the L1 mapping for this value. Use the *Outbound-Relevant* checkbox to define the relevant SAP status to DM status and DM maturity mapping that's to be sent to an external PLM system during outbound processing.

Note

You can enter and save only one record for each combination of *Cross-Plant Product Status* and *Outbound-Relevant*.

4. **Define Plant-Specific Status:** Here, you define which combination of status and maturity code results in which plant-specific status in the SAP system. If you've assigned a factory to an SAP plant in the Customizing activity [Map DM Factory to SAP Plant \[page 51\]](#), you can set a plant-specific status for this assignment here. The system will use this status when processing a status record that includes a Domain Model status for a specific factory. The available update modes and other settings correspond to those described above for [Define Product Status](#).
5. **Define Additional Basic Material Views and Define Plant Views:** The product version integration pattern `SAP_PVasMaterialMaster` maintains material masters in a multi-step approach. First, the API is called to maintain organization-independent data (for example, data stored in table `MARA`), and in further calls, organization-related data is maintained (for example, data stored in table `MARC`). For both, you can define which views will be created:
6. In [Define Additional Basic Material Views](#), you define which views you want to create in addition to the standard (K view/basic data view) while processing organization-independent data for a given product for which data is provided by a message or field and value mapping.
7. In [Define Plant Views](#), you define which views you want to create for a given plant if the corresponding plant data is provided by a message or field and value mapping.
8. **Define Inbound Field Mapping and Define Outbound Field Mapping:**
Use field mapping to decide how fields from the source structure are transferred to fields in the target structure. You can use [F4](#) help to find the respective values. The help provides all fields, including fields from the structure and included structures. To define field mapping, you do the following:
 - Choose [Create](#), [Update](#), or [Modify](#) in the column `MapMode`, to define what is valid for this line ([Modify](#) includes [Create](#) and [Update](#)).
 - Define an SAP target structure and SAP target attribute to define which field in the SAP system will be set. Choose [F4](#) to see the available values. If the [Override](#) checkbox is selected, the system uses the stored value instead of a value from the payload. This checkbox is for information purposes only and the setting can't be changed.
 - Define a Domain Model source structure (you can also use `<EXTENSION_FIELDS>`, see below) and Domain Model source attribute to define which data will be mapped to the target system.
 - Define a conversion if you want to transform the inbound data before setting it to the target field. The assignment is done using standard technical conversion rules. You can use preconfigured conversions, as well as customer-specific or target-data-specific ones. Define a domain value mapping to do data conversions with tables. You can also define a conversion function. The system offers a set of standard conversion functions.

🔗 Example

Use the [Timestamp](#) conversion function to map date information from a text field into a timestamp in the SAP system, if a data type other than `DATS` is used. This enables the conversion between the long timestamp used in the Meta Domain Model and the SAP format.

You can extend these functions with the enhancement spot `/PLMF/ES_FVM_CONVERSION`. [BA1: Define Custom Field and Value Mapping Conversions \[page 109\]](#).

- Define a default value if you want to set the standard value in case no data is received in the source field or if no source field is mapped. Note that you must define a value in internal format.

📌 Note

If you expect partial payloads, make sure you maintain the settings for [FVM Default Value Mode](#).

Field and value mapping supports customer field extensibility by extension fields, for example, to map plant data from product and factory data. You can use `<EXTENSION_FIELDS>` to map extension fields that

are sent from the external PLM system to a standard field or classification in the SAP system. The name of the extension field should be passed to the source attribute field. You can use extension fields for the DM target structure.

9. **Define Class Assignment:** Here, you can assign classes that will be added to products when creating and updating them. You can choose a conversion routine, if needed, set default values, or set a value mapping for the value transfer. You can also use these class assignments as a mapping target. You can map to a characteristic (target) and define which source field from the input structure will be used, for example, to map a status to a characteristic.
 - **Define Characteristics Mapping:** Here, you define the value mapping for the characters of the class. The functionality corresponds to the one for field and value mapping.
10. **Define Conditions for Outbound:** You can use the fields *Product Table Name* and *Attribute Name* to define which fields in the material tables trigger the transfer to the external PLM system when they are updated in the SAP system.
 - **Rule Group ID:** Defines a group of rules. Based on the rule group ID, the system determines the correct target systems for outbound data transfer. Target systems can be defined in the configuration activity [Define Rule Group \[page 25\]](#).
 - Enter a number in the *Counter for Outbound Attribute* field. You can maintain multiple entries for the same combination of *Product Table Name* and *Attribute* to specify different values.
 - **Value From:** If the configuration contains a value in the *Value From* field, the original value of the material must correspond to this value for it to be valid.
 - **Value From Any:** If *Value From Any* is selected, the original material value is ignored, and only a change in the attribute is considered relevant.
 - **Value To:** If the configuration contains a value in the *Value To* field, this value must be set by the change of the material for the defined field to be valid.
 - **Any Value To:** If *Value To Any* is selected, the target material value is ignored, and only a change in the attribute is considered relevant.

Related Information

[Domain Model Code Lists \[page 11\]](#)

[Define Scenario and Version \[page 17\]](#)

3.10.2 Define BRF+ Application for BAdI: PV as Material Post-Processing

Define the BRFplus application with the corresponding function for the default BAdI implementation `/PLMF/BI_PV_CUST_POSTPRSTP_BRF` of the [BAdI: PV as Material Post-Processing \[page 70\]](#).

Prerequisites

You have activated the default BAdI implementation `/PLMF/BI_PV_CUST_POSTPRSTP_BRF` of the [BAdI: PV as Material Post-Processing](#) [page 70].

Use

BRFplus (Business Rule Framework Plus) is a framework for managing business logic within SAP that is incorporated in the ABAP stack. It allows efficient and simple definition and processing of complex business rules.

After you have activated the default implementation of BAdI: PV as Material Post-Processing (`/PLMF/BI_PV_CUST_POSTPRSTP_BRF`), you can configure BRFplus application and function. The BRFplus application and function interfaces must apply to the input and output data structures used in the default implementation `/PLMF/BI_PV_CUST_POSTPRSTP_BRF`.

These BRFplus applications and functions are called in the example implementation for:

- [BRF+ Application](#)
Defines the BRFplus application that is called for material post-processing.
- [BRF+ Function](#)
Defines the BRFplus function from the BRFplus application that is called for material post-processing.

3.10.3 Define Factory Template Material - DM PV to SAP Material Master

Define a product template for a specific factory.

You can define one template per factory. The system uses this template during the creation of factory data in combination with the incoming payload data. If no based-on material is provided as part of incoming data, the data is created only based on the template. For more information, see [Define Product Templates - DM Product Version to SAP Material Master](#) [page 67].

3.10.4 Define Product Templates - DM Product Version to SAP Material Master

Define a reference product as a template for a certain product type (material master).

This template will be used during product creation in the SAP system. When a product version is received for creation in the SAP system, and the data below is missing in the payload, the system will create this data based on the existing template for the respective product type. Use the [F4](#) help for the [Product Template](#) field to filter

the available data according to various criteria, for example, *material by BOM* or *Material transferred from CAD system*. The following material views are used:

- Basic data 1 and 2
- MRP1
- MRP2
- MRP3
- MRP4
- Plant Stock Data

However, if this data is already included in the incoming payload, the received data will be used during creation. If the data is sent later, during update, it will overwrite the data previously created based on the template.

Note

The SAP product (material) type maintained in the template must match the product type assigned to the Domain Model product type (in the step *Define Mapping of Product Type*).

Note

If the template comprises plant data, this data will be copied, too, without creating key mapping for the factory. Therefore, you cannot receive factory data from subsequent payloads from the external PLM system for this factory. You can customize factory-specific template data in [Map DM Factory to SAP Plant \[page 51\]](#).

3.10.5 BAdI: Control PV to Material Master Behavior

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI for the integration pattern `SAP_PVasMaterialMaster` to influence specific functions in PV to Material integration.

```
/PLMF/BD_PV_M_IN_PROD_CTRL
```

This BAdI contains the following methods:

- `Define_Basic_View`: for defining which views are to be created in addition to the standard (K view/basic data view) for a given product. SAP provides a default implementation that reads the Customizing data and determines which views will be created, based on the Customizing settings maintained in [Map DM Product Version to SAP Material Master \[page 63\]](#) > Define Additional Basic Material Views.
- `Ignore_Plant_Data_Of_Template`: for defining if the system should ignore plant data provided by a template material. SAP provides a default implementation that ignores a template material's plant data if this data does not match the PV factory assignment of a DM message. To ignore this plant data, the *Only Copy Plants Matching Factory in Payload* mode needs to be selected in the *Copy Plant from Template Material* field in Customizing for [Map DM Product Version to SAP Material Master \[page 63\]](#) > Define Mapping of Product Type.

For more information about the standard settings (filters, single or multiple uses), see the Enhancement Spot Element Definitions tab in the BAdI Builder (transaction [SE18](#)).

BAdI Settings

This is a single-use BAdI.

Activities

1. Choose *Create*.
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.10.6 BAdI: Determine External System for PV Outbound Scenario

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to determine target systems that are relevant for product version outbound integration.

```
/PLMF/BD_PROD_TGT_SYST_DET
```

This BAdI verifies if a system is relevant for outbound integration and returns a list of relevant target systems:

- Target systems for which the sent product version type is marked as outbound-relevant in Customizing for [Map DM Product Version to SAP Material Master](#) > [Define Mapping of Product Type](#).
- Target systems for which changes have been made to a sent product.

This BAdI offers a default implementation that is called at runtime but can be overridden. Systems that you define as target systems will receive changes that affect product versions in the SAP system.

For more information about the standard settings (filters, single or multiple uses), see the [Enhancement Spot Element Definitions](#) tab in the BAdI Builder (transaction [SE18](#)).

Activities

1. Choose *Create*. Create the implementation of the enhancement spot.
2. Adapt the data from the template as required.
3. Save your entries.

3.10.7 BAdI: PV as Material Key Reservation Post-Processing

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to enhance the SAP integration with customer-specific steps after product version key reservation version objects processing has been performed successfully for inbound maintain request types.

```
/PLMF/BD_PV_KR_CUST_PSTPR_STEP
```

This BAdI can raise messages that influence the overall status. Additional messages are added to the existing messages and are written to the logs after the SAP process flow. The status is set accordingly based on the error category.

Note

You cannot use this BAdI to change the payload.

This BAdI has no default implementation. For more information about the standard settings (filters, single or multiple uses), see the *Enhancement Spot Element Definitions* tab in the BAdI Builder (transaction SE18).

Activities

1. Choose *Create*.
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.10.8 BAdI: PV as Material Post-Processing

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to enhance the SAP integration with customer-specific steps after product version objects processing has been performed successfully for inbound maintain request types.

```
/PLMF_BD_PV_CUST_POSTPROC_STEP
```

This BAdI can raise messages that influence the overall status. Additional messages are added to the existing messages and are written to the logs after the SAP process flow. The status is set accordingly based on the error category.

Note

You cannot use this BAdI to change the payload.

This BAdI has no default implementation. For more information about the standard settings (filters, single or multiple uses), see the *Enhancement Spot Element Definitions* tab in the BAdI Builder (transaction SE18).

Activities

1. Choose [Create](#).
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.10.9 BAdI: PV as Material Post-Processing – Read Extension Fields

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to enhance the SAP integration with customer-specific steps after product version objects processing has been performed successfully for inbound read request types.

```
/PLMF/BD_PV_READ_EXTN_FLD
```

This BAdI can read extension fields to add into the outbound data. Additional extension fields are added to the existing ones.

Note

You cannot use this BAdI to change the payload.

This BAdI offers the example BAdI implementation [/PLMV/PV_READ_EXTN_FLD](#) *Product Read Request: Aggregated Stock Prices*. For more information about the standard settings (filters, single or multiple uses), see the [Enhancement Spot Element Definitions](#) tab in the BAdI builder (transaction SE18).

Activities

1. Choose [Create](#).
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.11 DM Structure Version to SAP Classic BOM

This node groups several activities for mapping structure versions to SAP classic BOMs.

3.11.1 Define SV Model Unit Template ECN

In this configuration activity, you can use a model unit ECN template to define default effectivity values for a structure version component.

A model unit template ECN is a template change number that is used to define default effectivity values for an SV component or POL operation. It is used when operations in a product operation list or components in a bill of material (BOM) are received without dedicated model unit effectivity (MUE) but the object is under model unit control.

Prerequisites

- Scenarios and versions have been defined in configuration activity [Define Scenario and Version \[page 17\]](#).
- A mapping of structure version type and product type has been defined in the configuration activity [Define Mapping of Structure Version Type](#) in [Map DM Structure Version to SAP Classic BOM \[page 72\]](#).

Activities

1. Select a [Scenario ID](#) and [Scenario Version](#) to specify the scenario and version for which you would like to define data.
2. Select [Define Mapping of Structure Version](#) to specify the mapping of a Domain Model structure version type to a Domain Model product type for which you would like to define a MUE ECN template.
3. Select [Define MUE ECN Template](#) to specify a model unit template ECN for the selected DM structure version and product type.

3.11.2 Map DM Structure Version to SAP Classic BOM

In this Customizing activity, you maintain all necessary settings for structure version integration.

You define the mapping of Domain Model structure types to the SAP product types and BOM types. In addition, you can define essential control parameters for BOM management. You can create plant-specific structure versions. We recommend that you follow the configuration steps in the given order. You can receive structure version data, including visualizations and geometric instances.

You can receive substitutes for materials in a Domain Model structure version component from an external PLM system. These substitutes are added as additional components to the BOM in the SAP system. All

attributes of the SV components in the payload, such as quality, selection condition, transformation matrix, and so on are generated for the substitute material as well. Plant data and alternative UoMs can also be updated for substitutes. Each component can have multiple substitutes. In the SAP BOM, substitutes are assigned to an alternative group that is specific to the respective component and its assigned data.

Note

When a partial payload is received and a previously received component is deleted through this payload, the substitutes of this component are also deleted in the SAP system.

Prerequisites

- You have made all relevant settings in Customizing for [Domain Model Code Lists \[page 11\]](#). You can add additional values as required before you start mapping, or during the mapping process, for example a [Default Plant](#).
- Scenarios and versions have been configured in the system, in the Customizing activity [Define Scenario and Version \[page 17\]](#).
- You have maintained a change master (CM) profile in Customizing for [Maintain Profile](#) under [Cross-Application Components > Engineering Change Management](#).
- The integration pattern for the relevant structure version type is set to `SAP_SVasClassicBillofMaterial`. For more information, see [Integration Patterns \[page 19\]](#).

Activities

1. Select a [Scenario ID](#) and [Scenario Version](#) to specify the scenario ID and version for which you would like to define structure version data.
2. [Define Mapping of Structure Type](#): Here, you map a Domain Model structure type to the SAP product type and BOM types.
In [Default Plant](#), you can specify the plant to be set by default for a BOM if the Domain Model structure type is not plant-specific.
Select [Create Alternative UoM](#) to activate the automatic creation of an alternative unit of measure and unit of issue for a material or substitute material (only for classic BOM) with the quantity UoM of the component in the payload.
If [Implicit Plant Creation](#) is enabled, the system calls the PV maintain request for all given PVs in the SV maintain request. It uses the PV-key information and the factory data provided in the SV request. If the PV implementation is set up accordingly, the relevant plant data is created. This may involve copying data from given templates.
Enter a default [Authorization Group](#) to enable authorization protection for extended table maintenance.
Select [CAD Indicator](#) to set the CAD indicator in the document info records for documents of this type to indicate that a document is managed by CAD integration.
In [Effectivity Type](#), you can assign an effectivity type to be used for the creation of engineering change numbers (ECN) in case an SV or POL is transferred with model unit effectivity. If you assign an effectivity type, the validity of changes is determined by the parameters of the effectivity type, instead of by a valid-from date.

Confirm your selection and save.

3. **Define Structure Status:** Here, you define a combination of status and maturity code. In case of a change, the BOM status will be modified as set here. Select the conditions under which the update happens.
4. **Define Inbound Field Mapping** and **Define Outbound Field Mapping:**
Use field mapping to decide how fields from the source structure are transferred to fields in the target structure. You can use **F4** help to find the respective values. The help provides all fields including fields from the structure and included structures. To define field mapping, you do the following:
 - Choose **Create**, **Update**, or **Modify** in the column **MapMode**, to define what is valid for this line. (**Modify** includes **Create** and **Update**).
 - Define an SAP target structure and SAP target attribute to define which field in the SAP system will be set. Choose **F4** to see the available values. If the **Overwrite** checkbox is selected, the system uses the stored value instead of a value from the payload. This checkbox is for information purposes only and the setting can't be changed.
 - Define a Domain Model source structure (you can also use <SV-EXTENSION_FIELDS>, <COMPONENTS-EXTENSION_FIELDS>, or <SUBSTITUTES-EXTENSION_FIELDS>, see below) and a Domain Model source attribute to define which data will be mapped to the target system. You can map reference designators through the DM source attribute **REFERENCE_DESIGNATOR**.

Note

- The usage of <EXTENSION_FIELDS> is obsolete, but still possible to use for compatibility reasons.
- You cannot use <EXTENSION_FIELDS> in combination with another extension field.
- Extension fields of a child structure cannot be used for the parent structure.

Note

If you want to map texts of incoming reference designators, you can maintain the respective values, for example:

- **Domain Model Source Structure:** <SUBSTITUTES-EXTENSION_FIELDS>
- **Domain Model Source Attribute:** REFERENCE_DESIGNATOR_TEXT (20 characters maximum)
- **Domain Model Target Structure:** SUI_CLASS_DATA
- **Domain Model Target Attribute:** UPTXT

In the SAP system, the reference designator is stored as a BOM sub-item and will be visible as an installation point.

- Define a conversion in case you want to transform the inbound data before setting it to the target field. The assignment is done using standard technical conversion rules. You can use preconfigured conversions, and conversions that are specific to your customer or target data. Define a domain value mapping to do data conversions using tables. You can also define a conversion function. The system offers a set of standard conversion functions.

Example

If a data type other than **DATS** is used, use the **Timestamp** conversion function to map date information from a text field into a timestamp in the SAP system. This enables the conversion between the long timestamp used in the Meta Domain Model and the SAP format.

You can extend these functions through the enhancement spot `/PLMF/ES_FVM_CONVERSION`. **BAdI:** [Define Custom Field and Value Mapping Conversions \[page 109\]](#).

- Define a default value if you want to set the standard value in case no data is received in the source field or if no source field is mapped. Note that you must define a value in internal format.

Note

If you expect partial payloads, make sure you maintain the settings for *FVM Default Value Mode*.

Field and value mapping supports customer field extensibility by extension fields, for example, to map plant data from product and factory data. You can use <EXTENSION_FIELDS> to map extension fields that are sent from the external PLM system to a standard field or classification in the SAP system. The name of an extension field should be passed to the source attribute field. You can also use extension fields for values in the DM target structure. *Example:* If you receive a substitute for a structure version component from an external PLM system, you can map the additional data of this substitute through <EXTENSION_FIELDS> to the BOM component in the SAP system as *Alternative Item Data*.

5. *Define DM Parameters Mapping:* Here, you define the mapping of DM unit effectivity to effectivity parameters in the SAP system. You define effectivity parameters in Customizing for *Engineering Change Management* under **► Parameter Effectivity ► Define Parameters ▾**.
6. *Define DM Range Parameter Mapping:* Here, you define the mapping of DM unit effectivity to an open interval effectivity parameter in the SAP system. This setting is used for mapping the value of the *unitTo* and *unitFrom* fields from the payload for a structure version.

3.11.3 BAdI: VC Selection Condition for SV as Classic BOM

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to modify the selection condition for *structure version*.

```
/PLMF/BD_VC_SELECTION_COND
```

The BAdI implementation enhances the converted internal format

Prerequisites

The incoming Structure Version data must contain a selection condition in the external format.

Activities

1. Choose *Create*.
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.11.4 BAdI: SV as Classic BOM Post-Processing

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to enhance the SAP integration with customer-specific steps after structure version objects processing has been performed successfully for inbound maintain request types.

```
/PLMF/BD_SV_CUST_POSTPROC_STEP
```

This BAdI can raise messages that influence the overall status. Additional messages are added to the existing messages and are written to the logs after the SAP process flow. The status is set accordingly based on the error category.

Note

You cannot use this BAdI to change the payload.

This BAdI has no default implementation. For more information about the standard settings (filters, single or multiple uses), see the *Enhancement Spot Element Definitions* tab in the BAdI Builder (transaction SE18).

Activities

1. Choose *Create*.
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.11.5 BAdI: SV as Classic BOM Post-Processing – Read Extension Fields

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to enhance the SAP integration with customer-specific steps after structure version objects processing has been performed successfully for inbound read request types.

```
/PLMF/BD_SV_READ_EXTN_FLD
```

This BAdI can read extension fields to add into the outbound data. Additional extension fields are added to the existing ones.

Note

You cannot use this BAdI to change the payload.

For more information about the standard settings (filters, single or multiple uses), see the *Enhancement Spot Element Definitions* tab in the BAdI Builder (transaction SE18).

Activities

1. Choose *Create*.
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.12 DM Structure Version to SAP PEO Version BOM

This node groups several activities for mapping structure versions to SAP PEO Version BOMs.

3.12.1 Define Version BOM Model Unit Template ECN

In this configuration activity, you can use a model unit ECN template to define default effectivity values for a structure version component.

A model unit template ECN is a template change number that is used to define default effectivity values for an SV component or POL operation. It is used when operations in a product operation list or components in a bill of material (BOM) are received without dedicated model unit effectivity (MUE) but the object is under model unit control.

Prerequisites

- Scenarios and versions have been defined in the system, in the configuration activity [Define Scenario and Version \[page 17\]](#).
- A mapping of a Domain Model structure type to the SAP product type and BOM type has been defined in the configuration activity [Define Mapping of Version BOM Type](#) in [Map DM Structure Version to SAP PEO Version BOM \[page 78\]](#)

Activities

1. Select a *Scenario ID* and *Scenario Version* to specify the scenario and version for which you would like to define data.
2. Select [Define Mapping of Version BOM](#) to specify the mapping of a Domain Model structure version type to a Domain Model product type for which you would like to define a MUE ECN template.
3. Select [Define MUE ECN Template](#) to specify a model unit template ECN for the selected DM structure version and product type.

3.12.2 Map DM Structure Version to SAP PEO Version BOM

In this Customizing activity, you maintain all necessary settings for version BOM integration.

You define the mapping of Domain Model structure types to the SAP product types and BOM types. In addition, you can define essential control parameters for BOM management. You can create plant-specific BOM versions. You can receive structure version data, including visualizations and geometric instances. This enables you to view a configured product in the SAP system and use the 3D visualization across the extended supply chain, including the instance location of a component in the scene. Viewables are generated in the SAP system based on the incoming data. The structure version contains the positioning information (TMATRIX). Any type of SV is supported.

We recommend that you follow the configuration steps in the given order.

Prerequisites

- You have made all relevant settings in Customizing for [Domain Model Code Lists \[page 11\]](#). You can add additional values as required before you start mapping or during the mapping process.
- Scenarios and versions have been configured in the system, in the Customizing activity [Define Scenario and Version \[page 17\]](#).
- You have maintained a change master (CM) profile in Customizing for [Maintain Profile](#) under [Cross-Application Components > Engineering Change Management](#).
- The integration pattern for the relevant structure version type is set to `SAP_SVasPEOVersionBOM`. For more information, see [Integration Patterns \[page 19\]](#).

Note

If you want to assign a version BOM as a solution item to a change record, you need to make the following settings in [Map DM Change Object to SAP Change Record \[page 34\]](#):

- In [Define Mapping of Change Object Type](#), assign the change record type *MFG* to the required Domain Model change type, for example, change request.
- In [Define Relation Type](#), assign the object types *EBM* (EBOM) and *MBOM* to the DM relation type *Solution* with relevance *SOL* (solution).
- In this activity ([Map DM Structure Version to SAP PEO Version BOM](#)), select [Define Mapping of Version BOM Type](#) and define where the SV type will be assigned to a change record: Assign the required *CR RefObjT* (CR Reference Object Type), for example *EBM*.

Activities

1. Select a *Scenario ID* and *Scenario Version* to specify the scenario ID and version for which you would like to define and map structure version data.
2. [Define Mapping of Version BOM Type](#): Here you map a Domain Model structure type to the SAP product type and BOM version types.
Select *CAD Indicator* if the SAP CAD indicator will be turned on by default during processing.

In *Default Plant*, you can specify the plant to be set by default for a BOM version if the Domain Model structure type is not plant-specific.

Select *Create Alternative UoM* to activate the creation of an alternative unit of measure (instead of the base UoM of the BOM) for an item's material if a deviating unit of measure between material and BOM item is received.

If *Implicit Plant Creation* is enabled, materials are automatically assigned to the plant for a plant-specific BOM version, and the product version is created with one or more plant assignments. All plant-specific data is transferred from the template product.

If you select *Implicit Plant Creation*, the views *Work Scheduling*, *General Plant Data / Storage 1* and *General Plant Data / Storage 2* are added to the products of the structure version header and components.

Lifecycle Control determines the dependency between the version of a version BOM and the version of a product version (PV), and how the version BOM can be updated.

CR Reference Object Type provides a list of accepted object types for a BOM version.

Create Viewable Structure: Optionally, you can select this checkbox if you want to create a viewable structure for a selected Domain Model structure type. Because a viewable structure by itself is a document, you also must define the *Viewing Header Document Type*. If the checkbox is not set, the process of viewable structure maintenance is ignored.

Viewing DM Item Document Type: You can select a Domain Model document type for the viewable structure that's used to filter the items of the viewable structure.

Viewing Header Document Type: You can configure an SAP-specific document type. This field becomes mandatory once the checkbox *Create Viewable Structure* is activated.

Viewing Header Document Status: Set the status of previous documents that may have been sent before. This status corresponds to your document type. When you create a new version of a document as a prerequisite for the creation of a viewable structure, the older version is set with the status that's provided from this field. If you leave this field empty, the previous document version will keep the same status and a new version of the same document is created.

Use the *Snapshot-Relevant* checkbox to specify that the Domain Model BOM type is relevant for snapshot creation. If this checkbox is marked, a snapshot will be created for the selected version BOM. Choose the required *Snapshot Type*.

Note

For the feedback loop outbound scenario, you need to set the *Snapshot-Relevant* flag and choose *Manufacturing Snapshot*. For detailed information on how to configure a feedback loop, including troubleshooting, see [3344577](#).

If you select *Prepare Handover*, a snapshot will be prepared for handover to the external PLM system.

In *Group by*, you can select if you want to group version BOM line items, for example, by position number and material number or by other customer-specific criteria. Choose *Group by BAdI* to implement your own grouping logic, as defined by *BAdI: SV as PEO Version BOM Grouping Custom Logic* [page 82].

Standard grouping is based on attributes comparisons: Position number, material number, model unit effectivity, and selection conditions have to be identical. Units have to be of the same dimension. *Example*: Two components have the same material number and position number but differing selection conditions. In this case, the values are not grouped. Selection conditions and model unit effectivity cannot coexist for one component if grouping is active. If version BOM components are grouped, multiple viewables (communicated through DMU transformation matrix) are collected at the SAP version BOM component level.

Note

Grouping is not supported for the bidirectional transfer of BOM line items. Payloads that contain grouped BOM data may result in data loss or errors during inbound and outbound data transfer.

Confirm your selection and save.

3. **Define Version BOM Status:** Here, you define a combination of status and maturity code. In case of a change, the BOM version status will be modified as set here. Select the conditions under which the update happens.

Confirm your selection and save.

4. **Define Inbound Field Mapping** and **Define Outbound Field Mapping:**
Use field mapping to decide how fields from the source structure are transferred to fields in the target structure. You can use **F4** help to find the respective values. You can also use `<SV-EXTENSION_FIELDS>` or `<COMPONENTS-EXTENSION_FIELDS>` for source and target mapping (see below).

Note

- The usage of `<EXTENSION_FIELDS>` is obsolete, but still possible to use for compatibility reasons.
- You cannot use `<EXTENSION_FIELDS>` in combination with another extension field.
- Extension fields of a child structure cannot be used for the parent structure.

The help provides all fields including fields from the structure and included structures. To define field mapping, you do the following:

- Choose **Create**, **Update**, or **Modify** in the column **MapMode**, to define what is valid for this line. (**Modify** includes **Create** and **Update**).
- Define an SAP target structure and SAP target attribute to define which field in the SAP system shall be set. Choose **F4** to see the available values. If the **Overwrite** checkbox is selected, the system uses the stored value instead of a value from the payload. This checkbox is for information purposes only and the setting can't be changed.
- Define a Domain Model source structure and Domain Model source attribute to define which data shall be mapped to the target system. Choose **REFERENCE_DESIGNATOR** in the **Domain Model Source Attribute** field to map an incoming reference designator (40 characters maximum). In the SAP PEO Version BOM system, reference designators are stored in the transformation matrix.

Note

You can use the **BAdI: Control Scale Factor of Transformations** to provide a custom scale factor value for a Version BOM item as part of the SAP Transformation Matrix parameters. As PEO uses instance metadata to store reference designators, there is a risk that such instances are confounded with visual instance data. A custom scale factor helps to mitigate this risk. For more information, see **BAdI: Control Scale Factor of Transformations [page 81]**.

- Define a conversion in case you want to transform the inbound data before setting it to the target field. The assignment is done using standard technical conversion rules. You can use preconfigured conversions, and conversions that are specific to your customer or target data. Define a domain value mapping to do data conversions with tables. You can also define a conversion function. The system offers a set of standard conversion functions.

❁ Example

If a data type other than *DATS* is used, use the *Timestamp* conversion function to map date information from a text field into a timestamp in the SAP system. This enables the conversion between the long timestamp used in the Meta Domain Model and the SAP format.

You can extend these functions through the enhancement spot */PLMF/ES_FVM_CONVERSION*. BAdI: [Define Custom Field and Value Mapping Conversions \[page 109\]](#).

- Define a default value if you want to set the standard value in case no data is received in the source field or if no source field is mapped. Note that you must define a value in internal format.

ⓘ Note

If you expect partial payloads, make sure you maintain the settings for *FVM Default Value Mode*.

Field and value mapping supports customer field extensibility by extension fields, for example, to map plant data from product and factory data. You can use `<SV-EXTENSION_FIELDS>` and `<COMPONENTS-EXTENSION_FIELDS>` to map extension fields that are sent from the external PLM system to a standard field or classification in the SAP system or the other way round.

5. *Define DM Parameters Mapping*: Here, you define the mapping of DM unit effectivity to effectivity parameters in the SAP system. You define effectivity parameters in Customizing for *Engineering Change Management* under **► Parameter Effectivity ► Define Parameters ▾**.
6. *Define DM Range Parameter Mapping*: Here, you define the mapping of DM unit effectivity to an open interval effectivity parameter in the SAP system. This setting is used for mapping the value of the *unitTo* and *unitFrom* fields from the payload for a structure version.

3.12.3 BAdI: Control Scale Factor of Transformations

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). It enables you to provide a custom scale factor value for a version BOM item as part of the SAP Transformation Matrix parameters.

```
/PLMS/BD_SVV_ITEM_SCALE
```

You can use this BAdI during a version BOM inbound transfer for the SAP S/4HANA Manufacturing for production engineering and operations scenario, see [PEO Scenarios](#).

If you map a DM structure version to an SAP PEO version BOM, you can control the scale factor of transformations. The standard in PLMSI is set to **1**. Because PEO uses instance metadata to store reference designators, there's a risk that such instances are confused with visual instance data. To avoid this risk, the scale factor should be set to **0**.

You can use this BAdI to determine for which instance the scale factor should be set to **0**. If an item isn't returned, the default scale factor of **1** is applied. For more information about Customizing, see [Map DM Structure Version to SAP PEO Version BOM \[page 78\]](#).

For more information about the standard settings (filters, single or multiple uses), see the Enhancement Spot Element Definitions tab in the BAdI Builder (transaction [SE18](#)).

BAdI Settings

This is a single-use BAdI.

Activities

1. Choose [Create](#).
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.12.4 BAdI: Filter Viewable Structure Items Based on Document Type

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to enhance the SAP integration with customer-specific steps for filtering viewable structure items based on document type for inbound maintain request types.

```
/PLMS/BD_VS_ITEMS_PROCESSING
```

This BAdI can filter and add some additional viewable structure items and also raise messages that influence the overall status. Additional messages are added to the existing messages and are written to the logs after the SAP process flow. The status is set accordingly based on the error category.

This BAdI has no default implementation. For more information about the standard settings (filters, single or multiple uses), see the [Enhancement Spot Element Definitions](#) tab in the BAdI Builder (transaction SE18).

Activities

1. Choose [Create](#).
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.12.5 BAdI: SV as PEO Version BOM Grouping Custom Logic

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to group incoming items based on your own logic when creating an SAP PEO version BOM.

```
/PLMS/BD_SVV_CUST_GROUPING
```

In the activity [Map DM Structure Version to SAP PEO Version BOM \[page 78\]](#), set the structure version mapping to *Group by BAdI* to use the grouping logic defined by this BAdI.

This BAdI contains the following methods:

- `Merge`: For the implementation of customer-specific logic for grouping items.

For more information about the standard settings (filters, single or multiple uses), see the [Enhancement Spot Element Definitions](#) tab in the BAdI Builder (transaction `SE18`).

BAdI settings:

- Single - Use BAdI

Activities

1. Choose [Create](#).
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.12.6 BAdI: SV as PEO Version BOM Post-Processing

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to enhance the SAP integration with customer-specific steps after structure version as PEO version BOM objects processing has been performed successfully for inbound maintain request types.

```
/PLMS/BD_SVV_CUST_POSTPROC_STP
```

This BAdI can raise messages that influence the overall status. Additional messages are added to the existing messages and are written to the logs after the SAP process flow. The status is set accordingly based on the error category.

Note

You cannot use this BAdI to change the payload.

This BAdI has no default implementation. For more information about the standard settings (filters, single or multiple uses), see the [Enhancement Spot Element Definitions](#) tab in the BAdI Builder (transaction `SE18`).

Activities

1. Choose [Create](#).
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.12.7 BAdI: SV as PEO Version BOM Post-Processing – Read Extension Fields

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to enhance the SAP integration with customer-specific steps after structure version as PEO version objects processing has been performed successfully for inbound read request types.

```
/PLMS/BD_SVV_READ_EXTN_FLD
```

This BAdI can read extension fields to add into the outbound data. Additional extension fields are added to the existing ones.

Note

You cannot use this BAdI to change the payload.

For more information about the standard settings (filters, single or multiple uses), see the [Enhancement Spot Element Definitions](#) tab in the BAdI Builder (transaction SE18).

Activities

1. Choose [Create](#).
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.13 DM Customer Order Item to SAP Sales Order Item

This node groups activities for mapping customer order items (COI) to SAP sales order items.

3.13.1 Map DM Customer Order Item to SAP Sales Order Item

Make the required settings for customer order item (COI) integration.

In this Customizing activity, you define the mapping of Domain Model customer order types to the SAP sales order types. The system uses the COI business object for mapping.

As part of data federation setup, a display URL is sent as part of the response to the external PLM system. The display URL acts as a pointer to the specific COI data in the SAP system. The URL enables access to a sales order item directly in the SAP system from the external PLM system.

If a sales order contains a configurable material, the assigned variant option values are also sent to the external PLM system.

Also, you can define essential control parameters for COI, for example, the outbound-relevant flag. We recommend that you follow the configuration activities in the given order.

Prerequisites

- Domain Model code lists have been defined in the system.
- Scenarios and versions have been configured in the system, in Customizing for [Define Scenario and Version \[page 17\]](#).
- The integration pattern for the relevant COI processing is set to `SAP_COIasSalesOrderItem`. For more information, see [Integration Patterns \[page 19\]](#).

Note

The customer-specific integration pattern `SAP_COIasCustomerImpl` is not supported for the Change Record outbound integration process in PLM system integration (PLMSI) 3.0 FP02.

Activities

1. Select a *Scenario ID* and *Scenario Version* to specify the scenario ID and version for which you would like to define structure version data.
2. *Define Mapping of Customer Order Type*: Here, you map a Domain Model customer order type to an SAP sales order type. Use the *Outbound-Relevant* checkbox to define the relevant SAP sales order type to DM customer order type mapping in case of outbound requests.
3. *Define Outbound Field Mapping*:
Use field mapping to decide how fields from the source structure are transferred to fields in the target structure. You can use the F4 help to find the respective values. The help provides all fields, including fields from the structure and included structures. To define field mapping, you do the following:
Define an SAP source structure and SAP source attribute to define which field in the SAP system will be set. Choose *F4* to see the available values.
Define a Domain Model target structure (you can also use `<EXTENSION_FIELDS>`, see below) and Domain Model target attribute to define which data will be mapped to the target system.
Define a conversion in case you want to transform the inbound data before setting it to the target field. The assignment is done using standard technical conversion rules. You can use preconfigured conversions, and conversions that are specific to your customer or target data. Define a domain value mapping to do data conversions using tables. You can also define a conversion function. The system offers a set of standard conversion functions.

Example

If a data type other than *DATS* is used, use the *Timestamp* conversion function to map date information from a text field into a timestamp in the SAP system. This enables the conversion between the long timestamp used in the Meta Domain Model and the SAP format.

You can extend these functions through the enhancement spot `/PLMF/ES_FVM_CONVERSION`. **BAdI:** [Define Custom Field and Value Mapping Conversions \[page 109\]](#).

Define a default value if you want to set the standard value in case no data is specified in the source field or if no source field is mapped. Note that you must define a value in internal format.

Field and value mapping supports customer field extensibility by extension fields, for example, to map sales document items data. You can use <EXTENSION_FIELDS> to map extension fields that are sent from the SAP system to the external PLM system. The name of the extension field must be passed to the target attribute field. You can use extension fields for the DM target structure.

3.13.2 BAdI: Determine External System for COI Outbound Scenario

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to determine target systems that are relevant for customer order item outbound integration.

```
/PLMS/BD_SLS_ORD_I_TGT_SYST_D
```

This BAdI verifies if a system is relevant for outbound integration and returns a list of relevant target systems. The BAdI uses the list of incoming target systems of a customer order item product structure to determine which systems should receive customer order items based on the L1 mapping and integration pattern defined in Customizing.

This BAdI offers a default implementation that is called at runtime but can be overridden. Systems that you define as target systems will receive changes that affect customer order items in the SAP system.

For more information about the standard settings (filters, single or multiple uses), see the [Enhancement Spot Element Definitions](#) tab in the BAdI Builder (transaction [SE18](#)).

Activities

1. Choose [Create](#). Create the implementation of the enhancement spot.
2. Adapt the data from the template as required.
3. Save your entries.

3.13.3 BAdI: COI as Order Item Post-Processing – Read Extension Fields

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to enhance the SAP integration with customer-specific steps after customer order item object processing has been performed successfully for inbound read request types.

```
/PLMS/BD_COI_READ_EXTN_FLD
```

This BAdI can read extension fields to add into the outbound data. Additional extension fields are added to the existing ones.

Note

You cannot use this BAdI to change the payload.

For more information about the standard settings (filters, single or multiple uses), see the *Enhancement Spot Element Definitions* tab in the BAdI Builder (transaction `SE18`).

Activities

1. Choose *Create*.
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.14 DM Customer Order Item Product Structure to SAP Sales Order BOM

This node groups several activities for mapping customer item product structures to SAP sales order BOMs.

3.14.1 Map DM Customer Order Item Product Structure to SAP Sales Order BOM

In this Customizing activity, you make the required settings for customer order item product structure (COIPS) integration.

You define the mapping of Domain Model customer order item product structure types and SAP product types to SAP sales order BOM attributes. The system uses the `COIPS` business object for mapping. Customer order item product structures are created if the BOM for a material needs to be changed to meet special customer requirements in a single sales order. The material BOM itself is not changed.

You can define essential control parameters for sales order BOM, for example, BOM usage conditions. You can create plant-specific COIPS. We recommend that you follow the configuration activities in the given order.

Prerequisites

- Domain Model code lists have been defined in the system.
- Scenarios and versions have been defined in the system.
- You have maintained a change master (CM) profile in the Customizing activity *Maintain Profile* under [▶ Cross-Application Components > Engineering Change Management ▶](#).

- The integration pattern for the relevant COIPS processing is set to `SAP_COIPSasOrderBillOfMaterial`. For more information, see [Integration Patterns \[page 19\]](#).

Note

The customer-specific integration pattern `SAP_COIPSasCustomerImpl` is not supported for the Change Record outbound integration process in PLM system integration (PLMSI) 3.0 FP02.

- In *Define External System*, you have selected the *CO Default Destination* checkbox (see [Define External System \[page 24\]](#)), as the default destinations for CO and COIPS need to be identical. The *COIPS Default Destination* checkbox is no longer available!
- In *Map DM Change Object to SAP Change Record*, in the activity *Define Relation Type*, you have entered a suitable *DM Relation Type* and *Relevance*. You also have selected the checkbox *Send Referenced Object* (see [DM Change Object to SAP Change Record \[page 33\]](#)).

Activities

1. Select a *Scenario ID* and *Scenario Version* to specify the scenario ID and version for which you want to define product version data.
2. *Define Mapping of Customer Order Item Product Structure*: Here you map a Domain Model customer order item product structure to the SAP product type.
 - Enter a Domain Model COIPS type as defined in the Domain Model and map it to an SAP product type.
 - In *BOM Usage*, select the type of area in which a BOM can be used.
 - Select *CAD Indicator* if document info records of this type will be managed by CAD integration.
 - Select *Create Alternative Product Version UoM* to activate the creation of an alternative unit of measure (instead of the base UoM of the BOM) and unit of issue for an item's material if a deviating unit of measure between material and BOM item is received.
 - If *PV Implicit Plant Creation* is enabled, materials are automatically assigned to the plant for a plant-specific BOM, and the product version is created with one or more plant assignments. All plant-specific data is transferred from the template product.
 - Enter a default *Authorization Group* to enable authorization protection for extended table maintenance.
 - Select the *Outbound-Relevant checkbox* to define the mapping in case of outbound requests.
3. *Define Customer Order Item Product Structure Status*: Here, you map the Domain Model status and maturity to SAP order BOM status and maturity. In case of a change, the BOM status will be modified as set here. Select the conditions under which the update happens. Select the *Outbound-Relevant checkbox* to define the relevant SAP status to DM status mapping in case of outbound requests. Confirm your selection and save.
4. *Define Inbound Field Mapping* and *Define Outbound Field Mapping*:
Use field mapping to decide how fields from the source structure are transferred to fields in the target structure. You can use *F4* help to find the respective values. The help provides all fields including fields from the structure and included structures. To define field mapping, you do the following:
 - Choose *Create*, *Update*, or *Modify* in the column *MapMode*, to define what is valid for this line. (*Modify* includes *Create* and *Update*).
 - Define an SAP target structure and SAP target attribute to define which field in the SAP system will be set. Choose *F4* to see the available values. If the *Overwrite* checkbox is selected, the system uses the

stored value instead of a value from the payload. This checkbox is for information purposes only and the setting can't be changed.

- Define a Domain Model source structure (you can also use <COIPS-EXTENSION_FIELDS>, <COMPONENTS-EXTENSION_FIELDS>, see below) and a Domain Model source attribute to define which data shall be mapped to the target system.

Note

- The usage of <EXTENSION_FIELDS> is obsolete, but still possible to use for compatibility reasons.
 - You cannot use <EXTENSION_FIELDS> in combination with another extension field.
 - Extension fields of a child structure cannot be used for the parent structure.
- Define a conversion in case that you want to transform the inbound data before setting it to the target field. The assignment is done using standard technical conversion rules. You can use preconfigured as well as customer-specific or target-data-specific conversions. Define a domain value mapping to do data conversions using tables. You can also define a conversion function. The system offers a set of standard conversion functions.

Example

Use the *Timestamp* conversion function to map date information from a text field into a timestamp in the SAP system, if a data type other than *DATS* is used. This enables the conversion between the long timestamp used in the Meta Domain Model and the SAP format.

You can extend these functions using the enhancement spot */PLMF/ES_FVM_CONVERSION*. [BAdI: Define Custom Field and Value Mapping Conversions \[page 109\]](#).

- Define a default value if you want to set the standard value in case no data is received in the source field or if no source field is mapped. Note that you must define a value in internal format.

Note

If you expect partial payloads, make sure you maintain the settings for *FVM Default Value Mode*.

Field and value mapping supports customer field extensibility by extension fields, for example, to map plant data from product and factory data. You can use <EXTENSION_FIELDS> to map extension fields that are sent from the external PLM system to a standard field or classification in the SAP system. The name of the extension field should be passed to the source attribute field. Moreover, you can use extension fields for the DM target structure.

3.14.2 BAdI: Determine External System for COIPS Outbound Scenario

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to determine target systems that are relevant for customer order item outbound integration.

```
/PLMS/BD_ORD_BOM_TGT_SYST_DET
```

This BAdI verifies if a system is relevant for outbound integration and returns a list of relevant target systems. The BAdI uses the list of incoming target systems of a change record to determine which systems should

receive a customer order item product structure based on the L1 mapping and integration pattern defined in Customizing.

This BAdI returns target systems for which a sent Customer Order Item Product Structure type is marked as outbound-relevant in [Map DM Customer Order Item Product Structure to SAP Sales Order BOM \[page 87\]](#) > *Define Mapping of Cust. Order Item Product Structure*.

This BAdI offers a default implementation that is called at runtime but can be overridden. Systems that you define as target systems will receive changes that affect customer order items in the SAP system.

For more information about the standard settings (filters, single or multiple uses), see the [Enhancement Spot Element Definitions](#) tab in the BAdI Builder (transaction [SE18](#)).

Activities

1. Choose [Create](#). Create the implementation of the enhancement spot.
2. Adapt the data from the template as required.
3. Save your entries.

3.14.3 BAdI: COIPS as Order BOM Post-Processing

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to enhance the SAP integration with customer-specific steps after customer order item product structure objects processing has been performed successfully for inbound maintain request types.

```
/PLMS/BD_OPS_CUST_POSTPROC_STP
```

This BAdI can raise messages that influence the overall status. Additional messages are added to the existing messages and are written to the logs after the SAP process flow. The status is set accordingly based on the error category.

Note

You cannot use this BAdI to change the payload.

This BAdI has no default implementation. For more information about the standard settings (filters, single or multiple uses), see the [Enhancement Spot Element Definitions](#) tab in the BAdI Builder (transaction [SE18](#)).

Activities

1. Choose [Create](#).
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.15 DM Variant Definition to SAP Class

This node groups activities for mapping variant definitions to SAP classes.

3.15.1 Map DM Variant Definition to SAP Class

In this Customizing activity, you make the required settings to define the mapping of Domain Model variant definitions to SAP classes. We recommend that you follow the configuration steps in the given order.

Prerequisites

- You have made all relevant settings in Customizing for [Domain Model Code Lists \[page 11\]](#). Domain Model code lists have been defined. You can add additional values as needed before you start mapping, or during the mapping process.
- Scenarios and versions have been configured in Customizing for [Define Scenario and Version \[page 17\]](#).
- You have maintained a change master (CM) profile in Customizing for [Maintain Profile](#) under [Cross-Application Components > Engineering Change Management](#). Most objects that are received from the external PLM system include a valid-from date. The SAP system uses this date in combination with the data from the change master profile to create the change master for the object. Confirm and save your selections.
- The integration pattern for the relevant object type is set to `SAP_VDasClassification`. For more information, see [Integration Patterns \[page 19\]](#).

Activities

1. Select a [Scenario ID](#) and [Scenario Version](#) to specify the scenario ID and version for which you would like to define and map variant definition data.
2. [Define Variant Definition Type](#): Here, you define a variant definition type by class type and change master profile. The default class type is 300. You can use different class types if suitable custom class types are configured in the system.

Note

You can't use the same class type for variant definition and classification.

If you activate [Value Restriction](#), the system will create a constraint net, variant table, and constraints during the processing of an inbound variant option that was received with reference to a variant definition.

3. [Define Mapping of Variant Definition Status](#): Here, you define which Domain Model variant definition status code is mapped to which class status for a class type. In case of a change, the class status will be modified as set here. Select the conditions under which the update happens.

4. *Define Field Value Mapping:*

Use field mapping to decide how fields from the source structure are transferred to fields in the target structure. You can use *F4* help to find the respective values. The help provides all fields including fields from the structure and included structures. To define field mapping, you do the following:

- Choose *Create*, *Update*, or *Modify* in the column *MapMode*, to define what is valid for this line. (*Modify* includes *Create* and *Update*).
- Define an SAP target structure and SAP target attribute to define which field in the SAP system will be set. Choose *F4* to see the available values. If the *Overwrite* checkbox is selected, the system uses the stored value instead of a value from the payload. This checkbox is for information purposes only and the setting can't be changed.
- Define a Domain Model source structure (you can also use `<EXTENSION_FIELDS>`, see below) and Domain Model source attribute to define which data will be mapped to the target system.
- Define a conversion in case you want to transform the inbound data before setting it to the target field. The assignment is done using standard technical conversion rules. You can use preconfigured conversions, and conversions that are specific to your customer or target data. Define a domain value mapping to do data conversions using tables. You can also define a conversion function. The system offers a set of standard conversion functions.

Example

If a data type other than *DATS* is used, use the *Timestamp* conversion function to map date information from a text field into a timestamp in the SAP system. This enables the conversion between the long timestamp used in the Meta Domain Model and the SAP format.

You can extend these functions through the enhancement spot `/PLMF/ES_FVM_CONVERSION`. [BADl: Define Custom Field and Value Mapping Conversions \[page 109\]](#).

- Define a default value if you want to set the standard value in case no data is received in the source field or if no source field is mapped. Note that you must define a value in internal format.

Note

If you expect partial payloads, make sure you maintain the settings for *FVM Default Value Mode*.

Field and value mapping supports customer field extensibility by extension fields, for example, to map plant data from product and factory data. You can use `<EXTENSION_FIELDS>` to map extension fields that are sent from the external PLM system to a standard field or classification in the SAP system. The name of the extension field must be passed to the source attribute field. You can use extension fields for the DM target structure.

3.15.2 Maintain Class ID Number Range

In this Customizing activity, you make the required settings for the number ranges used for generating unique names for variant definitions in order to restrict available values.

Prerequisites

At least one interval should be created. Without an interval for number ranges, variant definitions cannot be created and the restriction of available values for characteristics is not possible.

Note

If the length of a variant definition ID exceeds the length of the *Class Number* (KLASSE) field of 18 characters, the ID cannot be used for a variant definition. In this case, the number range must be defined in the Customizing activity *Maintain Characteristic ID Number Range*. This is then used to determine the variant definition ID name.

Activities

1. *Display Intervals*: Here, you can see the existing intervals.
2. *Change Intervals*: Here, you can add a new interval, and change or delete existing intervals for a number range.
3. *Change NR Status*: Here, you specify the status for a number range.

3.15.3 BAdI: VD as Class Post-Processing

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to enhance the SAP integration with customer-specific steps after variant definition objects processing has been performed successfully for inbound maintain request types.

```
/PLMF/BD_VD_CUST_POSTPROC_STEP
```

This BAdI can raise messages that influence the overall status. Additional messages are added to the existing messages and are written to the logs after the SAP process flow. The status is set accordingly based on the error category.

Note

You cannot use this BAdI to change the payload.

This BAdI has no default implementation. For more information about the standard settings (filters, single or multiple uses), see the *Enhancement Spot Element Definitions* tab in the BAdI Builder (transaction SE18).

Activities

1. Choose *Create*.
2. Enter the custom enhancement implementation you have created.

3. Save your entries.

3.16 DM Variant Option Rule to SAP Object Dependency

This node groups activities for mapping variant option rules to SAP object dependencies.

3.16.1 Define Variant Configuration Engine

In this Customizing activity, you set the default processing mode for variant configuration.

Prerequisites

- Scenarios and versions have been configured in the system, in [Define Scenario and Version \[page 17\]](#).
- The integration pattern for the relevant variant configuration type is set to `SAP_VORasDependency`. For more information, see [Integration Patterns \[page 19\]](#).

Activities

1. Select a *Scenario ID* and *Scenario Version* to specify the scenario ID and version for which you would like to define the default variant configuration processing mode.
2. Select one of the following processing modes:
 - *LO-VC – SAP* – Standard Variant Configuration Profile: This mode is the default value for the configuration.
 - *Advanced VC* – Advanced Variant Configuration in SAP S/4HANA: This uses the same database tables for configuration and master data, but has some additional features, for example, this mode allows you to customize sales, planning, production, and engineering with process and data integration.

3.16.2 Map DM Variant Option Rule to SAP Object Dependency

In this Customizing activity, you make the required settings for variant option rule integration.

You define the mapping of Domain Model variant option rules to the SAP object dependency types. All available engineering options for a configurable product version are defined and the variant definition, and its associated engineering rules, are handed over from the external PLM system. The release of a variant definition starts the data transfer.

Prerequisites

- Scenarios and versions have been defined in the system, in Customizing activity [Define Scenario and Version \[page 17\]](#).
- The integration pattern for the relevant variant option rule type is set to `SAP_VORasDependency`. For more information, see [Integration Patterns \[page 19\]](#).
- You have maintained the variant definitions included in the incoming payload in the SAP system. For more information on variant definition mapping, see [Map DM Variant Definition to SAP Class \[page 91\]](#).

Activities

1. Select a *Scenario ID* and *Scenario Version* to specify the scenario ID and version for which you would like to define and map variant option rule data.
2. *Define Variant Option Rules*: Select a variant option rule type and maintain a change master profile. You need to maintain a CM profile for each rule type.

Note

The validity of variant option rules can be ended historically as part of a variant configuration (VC) model transfer. A *Discontinue* flag in the inbound payload defines the valid-from date from which the VOR will be discontinued.

Note

The variant option rule type *Default* (SAP procedures) is created with reference to a configurable class. If a new material is created with reference to the same class, the material configuration profile retains the rule assignment, even if its valid-from date exceeds the rule's discontinuation date. This is a limitation within PLM system integration.

3. *Define Mapping of Rules Status*: Map a DM status code to the corresponding dependency status, for example, *Locked*.

Note

The procedure created through integration should not be modified manually, especially not the priority of rules. If the objects are modified, the consistency of data cannot be guaranteed.

3.16.3 Maintain Constraint Net ID Number Range

In this Customizing activity, you make the required settings for the number ranges used for generating unique names for constraints and constraint nets in order to restrict available values.

Prerequisites

At least one interval should be created. Without an interval for number ranges, constraints and constraint nets cannot be created and the restriction of available values for characteristics is not possible.

Activities

1. *Display Intervals*: Here, you can see the existing intervals.
2. *Change Intervals*: Here, you can add a new interval, and change or delete existing intervals for a number range.
3. *Change NR Status*: Here, you specify the status for a number range.

3.16.4 Maintain Dependency Number Range

In this Customizing activity, you make the required settings for the number ranges used for generating unique names for variant option rules in order to restrict available values.

Prerequisites

At least one interval should be created. Without an interval for number ranges, variant option rules cannot be created and the restriction of available values for dependencies is not possible.

Note

As the dependency object is created based on the number range and in order for variant option rules to be successfully executed, the number range must be defined in the Customizing activity *Maintain Characteristic ID Number Range*.

Activities

1. *Display Intervals*: Here, you can see the existing intervals.
2. *Change Intervals*: Here, you can add a new interval, and change or delete existing intervals for a number range.
3. *Change NR Status*: Here, you specify the status for a number range.

3.16.5 Maintain Variant Table ID Number Range

In this Customizing activity, you make the required settings for number ranges for naming variant tables. The numbers are used to generate unique names for variant tables, which are created for constraints and constraint nets in order to restrict available values.

Prerequisites

At least one interval should be created. Without an interval for number ranges, variant tables cannot be created and the restriction of available values for characteristics is not possible.

Activities

1. *Display Intervals*: Here, you can see the existing intervals.
2. *Change Intervals*: Here, you can add a new interval, and change or delete existing intervals for a number range.
3. *Change NR Status*: Here, you specify the status for a number range. The *External* checkbox determines whether the number range interval is external or internal. The SAP system issues the numbers for internal number range intervals automatically. This number is between the from-number and the to-number. The last number issued is logged in the current number level.
You need to enter a number for external number issue. The number you enter needs to be between the from-number and the to-number.

3.16.6 BAdI: Set Reference Value for VOR Conditions and Expressions

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to set the reference value for variant option rule (VOR) conditions and expressions.

```
/PLMF/BD_VOR_PARSER
```

The BAdI implementation enhances the converted internal format.

Note

The incoming production operation data must contain at least an expression in the external format.

Activities

1. Choose [Create](#).
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.16.7 BAdI: VOR as Characteristic Post-Processing

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to enhance the SAP integration with customer-specific steps after variant option objects processing has been performed successfully for inbound maintain request types.

```
/PLMS/BD_VDR_POSTPROC_STEP
```

This BAdI can raise messages that influence the overall status. Additional messages are added to the existing messages and are written to the logs after the SAP process flow. The status is set accordingly based on the error category.

Note

You cannot use this BAdI to change the payload.

This BAdI has no default implementation. For more information about the standard settings (filters, single or multiple uses), see the [Enhancement Spot Element Definitions](#) tab in the BAdI Builder (transaction SE18).

Activities

1. Choose [Create](#).
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.17 DM Variant Option to SAP Characteristic

This node groups activities for mapping variant options to SAP characteristics.

3.17.1 Define Variant Option Boolean Values

Make the required settings to define variant option Boolean values and their descriptions.

Activities

Enter Boolean values and descriptions. The entries are used in [Map DM Variant Option to SAP Characteristic \[page 99\]](#) to fill SAP Boolean True and False values.

When a payload is sent for variant option, the Boolean value for the payload is created from this customizing activity.

3.17.2 Map DM Variant Option to SAP Characteristic

In this Customizing activity, you make the required settings to define the mapping of Domain Model variant options to SAP characteristics. We recommend that you follow the configuration steps in the given order.

Prerequisites

- You have made all relevant settings in Customizing for [Domain Model Code Lists \[page 11\]](#). Domain Model code lists have been defined. You can add additional values as needed before you start mapping, or during the mapping process.
- Scenarios and versions have been configured in Customizing for [Define Scenario and Version \[page 17\]](#).
- You have maintained a change master (CM) profile in Customizing for [Maintain Profile](#) under **► Cross-Application Components** **► Engineering Change Management** **▢**. Most objects that are received from the external PLM system include a valid-from date. The SAP system uses this date in combination with the data from the change master profile to create the change master for the object. Confirm and save your selections.
- The integration pattern for the relevant object type is set to `SAP_VOasCharacteristic`. For more information, see [Integration Patterns \[page 19\]](#).

Activities

1. Select a *Scenario ID* and *Scenario Version* to specify the scenario ID and version for which you would like to define and map variant option data.
2. *Define Variant Option Type*: Here, you define a variant option type by change master profile. You first define the default values. Then you maintain descriptions for these values in the subsequent steps. Save your entries.

Note

Set the *Change Master Profile* as selected in *Define Mapping of Change Object Type* under Customizing for *Define Change Object*. This ensures that the business objects created use the same profile and are enabled to attach an ECN.

The *ID Schema for VC* controls the logic of creating characteristic names and characteristic values:

- External: name and value are created from the payload (*External ID 4 Char. Name and Char. Value*).
 - Internal: name and value are created from the number range (*Internal ID for Char. Name and Char. Value*).
 - Internal: name is created from the number range and value from the payload (*Internal ID for Char. Name and External ID for Char. Value*).
 - You need to set the respective intervals in the Configuration activities *Maintain Characteristic Value Number Range* and *Maintain Characteristic ID Number Range*.
3. *Define Mapping of Variant Option Status*: Here, you define which Domain Model variant option status code maps to which characteristic status for a characteristic type. In case of a change, the characteristic status will be modified as set here. Select the conditions under which the update happens.
 4. *Define Boolean True Descriptions* and *Define Boolean False Descriptions*: Here, you specify the true and false values that are set in the SAP system for variant options of type Boolean that are loaded from the external PLM system. Choose a language and maintain language-dependent descriptions for the Boolean value.
 5. *Define Field Value Mapping*:
Use field mapping to decide how fields from the source structure are transferred to fields in the target structure. You can use *F4* help to find the respective values. The help provides all fields including fields from the structure and included structures. To define field mapping, you do the following:
 - Choose *Create*, *Update*, or *Modify* in the column *MapMode*, to define what is valid for this line. (*Modify* includes *Create* and *Update*).
 - Define an SAP target structure and SAP target attribute to define which field in the SAP system will be set. Choose *F4* to see the available values. If the *Overwrite* checkbox is selected, the system uses the stored value instead of a value from the payload. This checkbox is for information purposes only and the setting can't be changed.
 - Define a Domain Model source structure (you can also use `<EXTENSION_FIELDS>`, see below) and Domain Model source attribute to define which data will be mapped to the target system.
 - Define a conversion in case you want to transform the inbound data before setting it to the target field. The assignment is done using standard technical conversion rules. You can use preconfigured as well as customer-specific or target-data-specific conversions. Define a domain value mapping to do data conversions using tables. You can also define a conversion function. The system offers a set of standard conversion functions.

❖ Example

If a data type other than *DATS* is used, use the *Timestamp* conversion function to map date information from a text field into a timestamp in the SAP system. This enables the conversion between the long timestamp used in the Meta Domain Model and the SAP format.

You can extend these functions through the enhancement spot */PLMF/ES_FVM_CONVERSION*. [BA1: Define Custom Field and Value Mapping Conversions \[page 109\]](#).

- Define a default value if you want to set the standard value in case no data is received in the source field or if no source field is mapped. Note that you must define a value in internal format.

ⓘ Note

If you expect partial payloads, make sure you maintain the settings for *FVM Default Value Mode*.

Field and value mapping supports customer field extensibility by extension fields, for example, to map plant data from product and factory data. You can use `<EXTENSION_FIELDS>` to map extension fields that are sent from the external PLM system to a standard field or classification in the SAP system. The name of the extension field must be passed to the source attribute field. You can use extension fields for the DM target structure.

ⓘ Note

SAP can handle a maximum of 512 variant options in a selection condition.

3.17.3 Maintain Characteristic ID Number Range

In this Customizing activity, you configure the number ranges used for generating unique names for variant options in order to restrict available values.

Prerequisites

At least one interval should be created. Without an interval for number ranges, variant options cannot be created and the restriction of available values for characteristics is not possible.

Note: If the length of a variant option ID exceeds the length of the *Characteristic Name* (ATNAM) field of 30 characters, the ID cannot be used for a variant option. In this case, the number range must be defined in the Customizing activity [Maintain Range: Variant Option ID Number Range](#). This is then used to determine the variant option ID name.

Activities

1. *Display Intervals:* Display the existing intervals.
2. *Change Intervals:* Add a new interval, and change or delete existing intervals for a number range.

3. *Change NR Status*: Specify the status for a number range.

3.17.4 Maintain Characteristic Value Number Range

In this Customizing activity, you make the required settings for the number ranges used for generating unique values for variant options in order to restrict available values.

Prerequisites

At least one interval should be created. Without an interval for number ranges, variant options cannot be created and the restriction of available values for characteristics is not possible.

Note

If internal numbering is used in the creation of variant options of type string, values are replaced with the pattern PLMSI plus the number from the number range /PLMF/VVN. In this case, the number range must be defined in the Customizing activity [Maintain Characteristic ID Number Range \[page 101\]](#).

Activities

1. *Display Intervals*: Here, you can see the existing intervals.
2. *Change Intervals*: Here, you can add a new interval, and change or delete existing intervals for a number range.
3. *Change NR Status*: Here, you specify the status for a number range.

3.17.5 BAdI: VO as Characteristic Post-Processing

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to enhance the SAP integration with customer-specific steps after variant option objects processing has been performed successfully for inbound maintain request types.

```
/PLMF/BD_VO_CUST_POSTPROC_STEP
```

This BAdI can raise messages that influence the overall status. Additional messages are added to the existing messages and are written to the logs after the SAP process flow. The status is set accordingly based on the error category.

Note

You cannot use this BAdI to change the payload.

This BAdI has no default implementation. For more information about the standard settings (filters, single or multiple uses), see the *Enhancement Spot Element Definitions* tab in the BAdI Builder (transaction SE18).

Activities

1. Choose *Create*.
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.18 DM Vendor to SAP Business Partner

This node groups several activities for mapping Domain Model vendor data to SAP business partner data.

3.18.1 Map DM Vendor to SAP Business Partner

In this Customizing activity, you make the required settings for vendor integration.

You define the mapping of Domain Model vendor types to SAP business partner types. Also, you can define essential control parameters for vendors, for example, vendor status, SAP status code, and other data. We recommend that you follow the configuration steps in the given order.

Prerequisites

- Domain Model code lists have been defined in the system, in the Customizing activity [Domain Model Code Lists \[page 11\]](#).
- Scenarios and versions have been configured in the system, in the Customizing activity [Define Scenario and Version \[page 17\]](#).
- Business Partner to vendor integration has been enabled in Customizing: [Cross-Application Components](#) > [Master Data Synchronization](#) > [Synchronization Control](#) > [Synchronization Control](#) > [Activate Synchronization Options](#) > .
- The integration pattern for the relevant vendor type is set to *SAP_VDRasVendor*. For more information, see [Integration Patterns \[page 19\]](#).

Activities

1. Select a *Scenario ID* and *Scenario Version* to specify the scenario and version for which you want to define and map vendor data. Choose *Continue*.
2. *Define Mapping of Vendor Type*: Here, you define the values for vendor mapping, for example, vendor category, status profile, and vendor role, for a certain vendor type.

The selected *SAP Business Partner Grouping* uses the defined number range from SAP business partner Customizing. This number range can be used as external or internal numbering and must correspond to the chosen *Inbound Communication ID Mode*.

Note

The Domain Model contains one main address. The integration supports the bi-directional transfer of vendor data with the corresponding external ID or internal ID, depending on the customization. It is not possible to transfer any other role bi-directionally, apart from vendor role. The status management for the vendor is maintained through the corresponding status profile.

It is not possible to derive multiple vendors based on multiple Domain Model vendor addresses. If multiple addresses are needed for a vendor, the vendor needs to be created with multiple IDs.

Note

To create a vendor with an external number coming from an external PLM system, external number assignment needs to be activated in the Customizing activity [Cross-Application Components > Master Data Synchronization > Customer/Vendor Integration > Business Partner Settings > Settings for Vendor Integration > Field Assignment for Vendor Integration > Assign Keys > Define Number Assignment for Direction BP to Vendor](#).

Note

The selected *Document Link Type* must correspond to the chosen role, for example *Vendor*. You can set the business partner role in the Customizing activity [Cross-Application Components > Master Data Synchronization > Customer/Vendor Integration > Business Partner Settings > Settings for Customer Integration/Settings for Vendor Integration > Set BP Role Category for direction BP to Vendor](#) and [Set BP Role Category for direction BP to Customer](#).

You can define default vendor types in [Domain Model Code Lists \[page 11\]](#).

3. *Define Vendor Status*: Here, you map a Domain Model status code to an SAP vendor status. In case of a change, the vendor status will be modified as set here. Select the conditions under which the update happens. In case of outbound request, use the *Outbound-Relevant* checkbox to define the relevant SAP status to DM status mapping.
4. *Define Inbound Field Mapping* and *Define Outbound Field Mapping*: Use field mapping to decide how fields from the source structure are transferred to fields in the target structure. You can use the *F4* help to find the respective values. The help provides all fields, including fields from the structure and included structures. To define field mapping, do the following:
Choose *Create*, *Update* or *Modify* in the column *MapMode*, to define what is valid for this line. (*Modify* includes *Create* and *Update*).
Define an SAP target structure and SAP target attribute to define which field in the SAP system will be set. Choose *F4* to see the available values. If the *Override* checkbox is selected, the system uses the stored

value instead of a value from the payload. This checkbox is for information purposes only and the setting can't be changed.

Define a Domain Model source structure (you can also use <EXTENSION_FIELDS> or <MAIN_ADDR-EXTENSION_FIELDS>, see below) and Domain Model source attribute to define which data will be mapped to the target system.

Define a conversion if you want to transform the inbound data before setting it to the target field. The assignment is done using standard technical conversion rules. You can use preconfigured conversions, and conversions that are specific to your customer or target data. Define a domain value mapping to do data conversions using tables. You can also define a conversion function. The system offers a set of standard conversion functions.

Example: If a data type other than *DATS* is used, use the *Timestamp* conversion function to map time and date information from a text field into a timestamp in the SAP system. This enables the conversion between the long timestamp used in the Meta Domain Model and the SAP format. You can extend these functions with the enhancement spot */PLMF/ES_FVM_CONVERSION*. For more information, see [BAdI: Define Custom Field and Value Mapping Conversions \[page 109\]](#).

Define a default value if you want to set the standard value in case no data is received in the source field or if no source field is mapped. Note that you must define a value in internal format. Note: If you expect partial payloads, make sure you maintain the settings for FVM Default Value Mode.

Field and value mapping supports customer field extensibility by extension fields using <EXTENSION_FIELDS> or <MAIN_ADDR-EXTENSION_FIELDS> structures, for example, to map address data from main address extension fields data.

5. **Define Conditions for Outbound:** You can use the fields *Vendor Table Name* and *Attribute Name* to define which fields in the vendor tables trigger a data transfer to the external PLM system when they are updated in the SAP system.
 - **Rule Group ID:** Defines a group of rules. Based on the rule group ID, the system determines the correct target systems for outbound data transfer. Target systems can be defined in the configuration activity [Define Rule Group \[page 25\]](#).
 - Enter a number in the *Counter for Outbound Attribute* field. You can maintain multiple entries for the same combination of *Vendor Table Name* and *Attribute* to specify different values.
 - **Value From:** If the configuration contains a value in the *Value From* field, the actual original value is ignored, and only a change in the attribute is considered relevant.
 - **Value From Any:** If *Value From Any* is selected, the actual original value is ignored, and only a change in the attribute is considered relevant.
 - **Value To:** If the configuration contains a value in the *Value To* field, the change must set this value for the defined field to be valid.
 - **Value To Any:** If *Value To Any* is selected, the actual target value is ignored, and only a change in the attribute is considered relevant.

3.18.2 BAdI: Determine External System for Vendor Outbound Scenario

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to determine target systems relevant for the Vendor outbound integration.

```
/PLMS/BD_BP_TGT_SYST_DET
```

This BAdI verifies if a system is relevant for outbound integration and returns a list of relevant target systems:

- Target systems for which the sent DM vendor type is marked as outbound-relevant in Customizing for [▶ Map DM Vendor to SAP Business Partner ▶ Define Mapping of Vendor Type ▶](#), see [Map DM Vendor to SAP Business Partner \[page 103\]](#).
- Target systems for which changes have been made to a sent vendor.

This BAdI offers a default implementation that is called at runtime but can be overridden. Systems that you define as target systems will receive changes that affect vendor data in the SAP system.

For more information about the standard settings (filters, single or multiple uses), see the [Enhancement Spot Element Definitions](#) tab in the BAdI Builder (transaction [SE18](#)).

Activities

1. Choose [Create](#). Create the implementation of the enhancement spot.
2. Adapt the data from the template as required.
3. Save your entries.

3.18.3 BAdI: Vendor as BP Post-Processing

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to enhance the SAP integration with customer-specific steps after vendor objects processing has been performed successfully for inbound maintain request types.

```
/PLMF/BD_VDR_CUST_PO
```

This BAdI can raise messages that influence the overall status. Additional messages are added to the existing messages and are written to the logs after the SAP process flow. The status is set accordingly based on the error category.

Note

You cannot use this BAdI to change the payload.

This BAdI has no default implementation. For more information about the standard settings (filters, single or multiple uses), see the [Enhancement Spot Element Definitions](#) tab in the BAdI Builder (transaction [SE18](#)).

Activities

1. Choose [Create](#).
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.19 DM View Structure Version to SAP PEO MRS

This node groups activities for mapping view structure versions to SAP PEO manufacturing reference structures.

3.19.1 Maintain MRS Object ID Number Range

In this Customizing activity, you configure the number ranges used for generating unique values for MRS object IDs to restrict the available values.

Prerequisites

At least one interval should be created. Without an interval for number ranges, View Structure Version as PEO MRS cannot be created.

Activities

1. *Display Intervals*: Display the existing intervals.
2. *Change Intervals*: Add a new interval, and change or delete existing intervals for a number range.
3. *Change NR Status*: Specify the status for a number range.

3.19.2 Map DM View Structure Version to SAP PEO MRS

In this Customizing activity, you configure the settings for view structure version integration.

You define the mapping of Domain Model View Structure Version (VSV) types to SAP PEO MRS types. The Manufacturing Reference Structure (MRS) is a separate hierarchy that can be linked to the BOM hierarchy to provide additional contextual information (for example, spacial and functional information) for BOM components during the production process. We recommend that you follow the configuration steps in the given order.

Prerequisites

- Domain Model code lists have been defined in the Customizing activity [Domain Model Code Lists \[page 11\]](#).
- Scenarios and versions have been configured in the Customizing activity [Define Scenario and Version \[page 17\]](#).

- The integration pattern for the relevant view structure version type is set to `SAP_VSVasPEOMRS`. For more information, see [Integration Patterns \[page 19\]](#).

Activities

1. Select a *Scenario ID* and *Scenario Version* to specify the scenario and version for which you would like to define VSV data. Choose *Continue*.
2. *Define Mapping of View Structure Version Type*: Here, you define the values for view structure version mapping, for example, *MRS Usage* and *Authority Group*.
3. *Define Inbound Field Mapping*: Use field mapping to decide how fields from the source structure are transferred to fields in the target structure. You can use the F4 help to find the respective values. This help function provides all fields, including those from the structure and any included structures. To define field mapping, you do the following:
 - Choose *Create*, *Update* or *Modify* in the column *MapMode*, to define what is valid for this line. (*Modify* includes *Create* and *Update*).
 - Define an SAP target structure and an SAP target attribute to define which field in the SAP system will be set. Choose *F4* to see the available values.
 - Define a Domain Model source structure (you can also use `<EXTENSION_FIELDS>`, see below) and a Domain Model source attribute to define which data will be mapped to the target system.
 - Define an *FMV conversion* if you want to transform the inbound data before setting it to the target field. The assignment is done using standard technical conversion rules. You can use preconfigured, customer-specific, or target-data-specific conversions. Define a domain value mapping to perform data conversions using tables. You can also define a conversion function. The system offers a set of standard conversion functions. For example, if a data type other than DATS is used, use the timestamp conversion function to map date information from a text field into a timestamp in the SAP system. This enables the conversion between the long timestamp used in the Meta Domain Model and the SAP format. You can extend these functions through the enhancement spot `/PLMF/ES_FVM_CONVERSION`. For more information, see [BAdI: Define Custom Field and Value Mapping Conversions \[page 109\]](#).
 - Define a default value if you want to set the standard value in case no data is received in the source field or if no source field is mapped. Note that you must define a value in internal format.

3.19.3 BAdI: VSV as PEO MRS Post-Processing

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to enhance the SAP integration with customer-specific steps after view structure version as PEO MRS objects processing has been performed successfully for inbound maintain request types.

```
/PLMF/BD_VSV_CUST_POSTPROC_STP
```

This BAdI can raise messages that influence the overall status. Additional messages are added to the existing messages and are written to the logs after the SAP process flow. The status is set accordingly based on the error category.

Note

You cannot use this BAdI to change the payload.

This BAdI has no default implementation. For more information about the standard settings (filters, single or multiple uses), see the *Enhancement Spot Element Definitions* tab in the BAdI Builder (transaction SE18).

Activities

1. Choose *Create*.
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.20 Business Add-Ins (BAdIs) for PLM System Integration

3.20.1 BAdI: Define Custom Field and Value Mapping Conversions

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to extend the preconfigured field and value mapping conversion functions for the integration.

```
/PLMF/ES_FVM_CONVERSION
```

This BAdI offers standard field and value mapping conversions that you can use as a template. You can choose one or several of the offered conversion possibilities from the value help in the activities *Define Inbound Field Mapping* and *Define Outbound Field Mapping* in Customizing activities, for example *Map DM Product Version to SAP Material Master* [page 63]. For more information about the standard settings (filters, single or multiple uses), see the *Enhancement Spot Element Definitions* tab in the BAdI Builder (transaction SE18).

The following standard BAdIs are available:

- `/PLMF/BI_FVM_CONV_CONDENSE`: Field and Value Mapping: Condense Values: Use this BAdI to remove leading and trailing spaces.
- `/PLMF/BI_FVM_CONV_ALPHA`: Field and Value Mapping: Alpha Conversion: Use this BAdI to convert alphanumeric data to/from alpha. (For inbound messages, missing zeros are added to the input; for outbound messages, zeros from internal table data are removed.)
- `/PLMF/BI_FVM_CONV_UPPER`: Field and Value Mapping: Convert to Upper Case: Use this BAdI to convert all letters in the character string of incoming values from the payload to upper case.
- `/PLMF/BI_FVM_CONV_HOURS`: Field and Value Mapping: Convert Hours: Use this BAdI to convert data in ISO format ValueDuration to/from hours (float).
- `/PLMF/BI_FVM_CONV_MINUTES`: Field and Value Mapping: Convert Minutes: Use this BAdI to convert data in ISO format ValueDuration to/from minutes (float).
- `/PLMF/BI_FVM_CONV_SECONDS`: Field and Value Mapping: Convert Seconds: Use this BAdI to convert data in ISO format ValueDuration to/from seconds (float).

- `/PLMF/BI_FVM_CONV_TZNTSTEMPL`: Field and Value Mapping: Convert Timestamps: If a data type other than `DATS` is used, use this BAdI to convert date information from a text field to the correct format, to map the data into a timestamp in the SAP system.

ⓘ Note

For each object type, for example, `PV`, these implementations use the same Domain Model structures (e.g., `/PLMF/S_IN_PV`) as source and target structures as outlined in the objects-specific configuration activities `Define Inbound Field Mapping` and `Define Outbound Field Mapping` for the respective object type.

Activities

1. Create the implementation of the enhancement spot.
2. Adapt the data from the template as required.

Example

You need a custom field mapping, for example, the field `Labor` for a product version. You want to create an implementation that derives the value through a custom table access.

3.20.2 BAdI: Define PEO Snapshot & Snapshot Revision Scope

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to determine the scope of the creation of snapshots and snapshot revisions.

```
/PLMS/BD_CR_PREP_SNAPSHOT
```

Determine the scope of the creation of snapshots and snapshot revisions by defining `T_EBOM_IN` and `T_MBOM_IN` of API `MPE_CREATE_SNPSHT_AND_PREP_CHG`.

SAP provides a default implementation that adds the version BOM, that is in snapshot to `T_EBOM_IN` or `T_MBOM_IN`, based on BOM type.

Activities

1. Choose `Create`.
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.20.3 BAdIs for Changing Message and Conversion

You can use these Customizing activities to enhance the SAP integration with customer-specific steps for defining change messages and conversions.

With the BAdI: Change Message Request/Response, you can make customer-specific changes to data before SAP's standard logic processes it. This example implementation lets you modify data by configuring BRF applications and functions in the Customizing activity 'Define Change Message BRF+ applications'. The input and output data structures for the implementation must be used.

3.20.3.1 BAdI: Change Message Request/Response

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). It implements custom logic to adjust the data in the inbound maintain request before processing by SAP standard protocols, and modifies data within the response prior to its delivery to the external system.

```
/PLMF/BD_DM_MESSAGE_IN
```

You can use this BAdI to change values for any objects contained in the payload received from external PLM systems to maintain requests or responses. The BAdI includes an example implementation that allows data changes through BRF applications and functions defined in the Customizing activity "Define Change Message BRF+ applications".

If the inbound message was changed after the execution of the BAdI logic, it is uniquely identified by the custom message ID. During processing, all related system messages are logged in SLG1, including those for the original message ID.

For more information about standard settings (filters, single or multiple uses), see the [Enhancement Spot Element Definitions](#) tab in the BAdI Builder (transaction [SE18](#)).

BAdI Settings

This is a single-use BAdI.

Activities

1. Choose [Create](#).
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.20.3.2 Define Change Message BRF+ Applications

BRFplus (Business Rule Framework Plus) is a framework for managing business logic within SAP that is incorporated in the ABAP stack. It allows efficient and simple definition and processing of complex business rules. See also [Business Rule Framework plus \(BRFplus\)](#). Define the BRFplus applications with their corresponding functions for the BAdI example implementation `/PLMF/BI_DM_MESSAGE_IN_IMPL` of *BAdI: Change Message Request/Response*.

Prerequisites

You have activated the example implementation of [BAdI: Change Message Request/Response \[page 111\]](#) (`/PLMF/BD_DM_MESSAGE_IN`).

Use

If you have activated the example implementation `/PLMF/BI_DM_MESSAGE_IN_IMPL` of [BAdI: Change Message Request/Response \[page 111\]](#), you can use this Customizing activity to configure BRFplus applications and functions per DM object type. The BRFplus application and function interfaces must apply to the input and output data structures used in the example implementation `/PLMF/BI_DM_MESSAGE_IN_IMPL`.

These BRFplus applications and functions are called in the example implementation for the following:

- [Object type](#)
Defines the Domain Model object type the BRFplus application is applicable for.
- [BRF+ Change Request Application](#)
Defines the BRFplus application that is called for requests.
- [BRF+ Change Request Function](#)
Defines the BRFplus function from the BRF+ application that is called for requests.
- [BRF+ Change Response Application](#)
Determines the BRFplus application that is called for responses.
- [BRF+ Change Response Function](#)
Determines the BRFplus function from the BRF+ application that is called for responses.

3.20.4 BAdIs for Defining Custom Logic

You can use these BAdIs to enhance the SAP integration with customer-specific steps for defining custom logic.

3.20.4.1 BAdI: Change Object (CO) Custom Logic

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to implement customer-specific logic for the change object (CO) DM object.

```
/PLMF/BD_CO_IP_CUSTOM_IMPL
```

SAP offers standard and custom-specific integration patterns for the change object integration process. You can select the needed integration pattern for the defined change object Domain Model type for a used scenario under *Assign Integration Pattern* in the Customizing for *Define Scenario and Version*. Using the `SAP_COasCustomerImplementation` integration pattern and the implementation of this BAdI allow you to replace the standard implementation for the change object integration with your own coding to implement your logic. The BAdI implementation supports inbound and outbound integration directions.

This BAdI contains the following methods:

- `Maintain`: for the implementation of customer-specific logic for inbound maintain requests.
- `Read`: for the implementation of customer-specific logic for inbound read requests and outbound maintain requests.
- `Check_Event_Relev_By_SAP_Obj`: for the implementation of customer-specific logic for outbound maintain requests to determine the custom integration pattern relevance for outbound integration by the SAP object type and SAP object image.
- `Determine_DM_Key_By_SAP_Key`: for the implementation of customer-specific logic for outbound maintain requests to determine the DM key by the SAP key.
- `Determine_Receiver_For_Maint`: for the implementation of customer-specific logic for outbound maintain requests to determine the system the object will be integrated with.
- `Get_Event_Data_Type_Name`: for the implementation of customer-specific logic for outbound maintain requests to determine the event data type name for the custom SAP object type.
- `Get_SAP_Object_Type`: for specifying the SAP object type when implementing a custom integration pattern. The SAP object type could be a PLM SAP object type, or the customer-specific SAP object type starting with the prefix Z.
- `Get_SAP_Object_Type_Shlp`: for the implementation of customer-specific logic for outbound maintain requests to determine the search help for the custom SAP object type.
- `Process_After_Outbound_Call`: for the implementation of customer-specific logic for outbound maintain requests in the post-processing step (the method is called after the request was sent).

For more information about the standard settings (filters, single or multiple uses), see the *Enhancement Spot Element Definitions* tab in the BAdI Builder (transaction SE18).

BAdI settings:

- Single - Use BAdI

Activities

1. Choose *Create*.
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.20.4.2 BAdI: Customer Order Item (COI) Custom Logic

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to implement customer-specific logic for the customer order item (COI) DM object.

```
/PLMS/BD_COI_IP_CUSTOM_IMPL
```

SAP offers standard integration patterns that enable you to select different target applications for a defined integration object. You can choose to replace the standard implementation for the integration of a certain object type with your own coding to implement your own logic. To do so, you can select the integration pattern we offer for custom implementations, *SAP_COIasCustomerImpl*. When a payload is sent, this BAdI implementation is called if you have chosen an integration pattern of type *SAP_COIasCustomerImpl*.

This BAdI contains the following methods:

- *Check_Event_Relev_By_SAP_Obj*: for the implementation of customer-specific logic for outbound maintain requests to determine the integration pattern relevance by the SAP object type.
- *Determine_DM_Key_By_SAP_Key*: for the implementation of customer-specific logic for outbound maintain requests to determine the DM key by the SAP key.
- *Determine_Receiver_For_Maint*: for the implementation of customer-specific logic for outbound maintain requests to determine the system the object will be integrated with.
- *Get_Event_Data_Type_Name*: for the implementation of customer-specific logic for outbound maintain requests to determine the event data structure for the SAP object.
- *Get_SAP_Object_Type*: for specifying the SAP object type when implementing a custom integration pattern. The SAP object type could be a PLM SAP object type, or the customer-specific SAP object type starting with the prefix z.
- *Get_SAP_Object_Type_Shlp*: for the implementation of customer-specific logic for outbound maintain requests to determine the search help for the SAP object type.
- *Process_After_Outbound_Call*: for the implementation of customer-specific logic for outbound maintain requests in the post-processing step.

For more information about the standard settings (filters, single or multiple uses), see the *Enhancement Spot Element Definitions* tab in the BAdI Builder (transaction SE18).

BAdI settings:

- Single - Use BAdI

Activities

1. Choose *Create*.
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.20.4.3 BAdI: Customer Order Item Product Structure (COIPS) Custom Logic

/PLMS/BD_OPS_IP_CUSTOM_IMPL

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to implement customer-specific logic for the customer order item product structure (COIPS) DM object.

SAP offers standard integration patterns that enable you to select different target applications for a defined integration object. You can choose to replace the standard implementation for the integration of a certain object type with your own coding to implement your own logic. To do so, you can select one of integration patterns we offer for custom implementations, `SAP_COIPSasCustomerImpl`. When a payload is sent, this BAdI implementation is called if you have chosen an integration pattern of type `SAP_COIPSasCustomerImpl`.

This BAdI contains the following methods:

- `Read`: for the implementation of customer-specific logic for inbound read requests.
- `Maintain`: for the implementation of customer-specific logic for inbound maintain requests.
- `Check_Event_Relev_By_SAP_Obj`: for the implementation of customer-specific logic for outbound maintain requests to determine the integration pattern relevance by the SAP object type.
- `Determine_DM_Key_By_SAP_Key`: for the implementation of customer-specific logic for outbound maintain requests to determine the DM key by the SAP key.
- `Determine_Receiver_For_Maint`: for the implementation of customer-specific logic for outbound maintain requests to determine the system the object will be integrated with.
- `Get_Event_Data_Type_Name`: for the implementation of customer-specific logic for outbound maintain requests to determine the event data structure for the SAP object.
- `Get_SAP_Object_Type`: for specifying the SAP object type when implementing a custom integration pattern. The SAP object type could be a PLM SAP object type, or the customer-specific SAP object type starting with the prefix `Z`.
- `Get_SAP_Object_Type_Shlp`: for the implementation of customer-specific logic for outbound maintain requests to determine the search help for the SAP object type.
- `Process_After_Outbound_Call`: for the implementation of customer-specific logic for outbound maintain requests in the post-processing step.

This BAdI has no default implementation. For more information about the standard settings (filters, single or multiple uses), see the *Enhancement Spot Element Definitions* tab in the BAdI Builder (transaction `SE18`).

BAdI settings:

- Single - Use BAdI

Activities

1. Choose *Create*.
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.20.4.4 BAdI: Document Version (DV) Custom Logic

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to implement customer-specific logic for the document version (DV) DM object.

```
/PLMF/BD_DV_IP_CUSTOM_IMPL
```

SAP offers standard and custom-specific integration patterns for the document version integration process. You can select the needed integration pattern for the defined document version Domain Model type for a used scenario under *Assign Integration Pattern* in the Customizing for *Define Scenario and Version*. Using the `SAP_DVasCustomerImplementation` integration pattern and the implementation of that BAdI allow you to replace the standard implementation for the document version integration with your own coding to implement your logic. The BAdI implementation supports inbound and outbound integration directions.

This BAdI contains the following methods:

- `Maintain`: for the implementation of customer-specific logic for inbound maintain requests.
- `Read`: for the implementation of customer-specific logic for inbound read requests and outbound maintain requests.
- `Check_Event_Relev_By_SAP_Obj`: for the implementation of customer-specific logic for outbound maintain requests to determine the custom integration pattern relevance for outbound integration by the SAP object type and SAP object image.
- `Determine_DM_Key_By_SAP_Key`: for the implementation of customer-specific logic for outbound maintain requests to determine the DM key by the SAP key.
- `Determine_Receiver_For_Maint`: for the implementation of customer-specific logic for outbound maintain requests to determine the target systems the object will be integrated with.
- `Get_Event_Data_Type_Name`: for the implementation of customer-specific logic for outbound maintain requests to determine the event data type name for the Custom SAP object type.
- `Get_SAP_Object_Type`: for specifying the SAP object type when implementing a custom integration pattern. The SAP object type could be a PLM SAP object type, or the customer-specific SAP object type starting with the prefix Z.
- `Get_SAP_Object_Type_Shlp`: for the implementation of customer-specific logic for outbound maintain requests to determine the search help for the Custom SAP object type.
- `Process_After_Outbound_Call`: for the implementation of customer-specific logic for outbound maintain requests in the post-processing step (the method is called after the request has been sent).

For more information about the standard settings (filters, single or multiple uses), see the *Enhancement Spot Element Definitions* tab in the BAdI Builder (transaction SE18).

BAdI settings:

- Single - Use BAdI

Activities

1. Choose *Create*.
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.20.4.5 BAdI: Document Version File (DVF) Custom Logic

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to implement customer-specific logic for the DM object document version file (DVF).

```
/PLMF/BD_DVF_IP_CUSTOM_IMPL
```

SAP offers standard and custom-specific integration patterns for the document version file read process. You can select the needed integration pattern for the document version file for a used scenario under [Assign Integration Pattern](#) in the Customizing for [Define Scenario and Version](#). Using the `SAP_DVFasCustomerImpl` integration pattern and the implementation of that BAdI allow you to replace the standard implementation for the document version file read process with your own coding to implement your logic. The BAdI implementation supports inbound integration direction.

This BAdI contains the following method:

- `Read`: for the implementation of customer-specific logic for inbound read requests and outbound maintain requests.

For more information about the standard settings (filters, single or multiple uses), see the Enhancement Spot Element Definitions tab in the BAdI Builder (transaction [SE18](#)).

BAdI settings:

- Single-Use BAdI

Activities

1. Choose [Create](#).
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.20.4.6 BAdI: File Confirmation (FC) Custom Logic

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to implement customer-specific logic for the file confirmation (FC) DM object.

```
/PLMF/BD_FC_IP_CUSTOM_IMPL
```

SAP offers standard and custom-specific integration patterns for the file upload confirmation integration process. You can select the needed integration pattern for the file upload confirmation for a used scenario under [Assign Integration Pattern](#) in the Customizing for [Define Scenario and Version](#). Using the `SAP_FCasCustomerImplementation` integration pattern and the implementation of that BAdI allow you to replace the standard implementation for the file upload confirmation integration process with your coding to implement your logic. The BAdI implementation supports inbound integration direction.

This BAdI contains the following method:

- `Maintain`: for the implementation of customer-specific logic for inbound maintain requests.

For more information about the standard settings (filters, single or multiple uses), see the [Enhancement Spot Element Definitions](#) tab in the BAdI Builder (transaction `SE18`).

BAdI settings:

- Single - Use BAdI

Activities

1. Choose [Create](#).
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.20.4.7 BAdI: Inspection Characteristic (IC) Custom Logic

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to implement customer-specific logic for the inspection characteristic (IC) DM object.

```
/PLMF/BD_IC_IP_CUSTOM_IMPL
```

SAP offers standard and custom-specific integration patterns for the inspection characteristic integration process. You can select the needed integration pattern for the inspection characteristic Domain Model type for a used scenario under [Assign Integration Pattern](#) in the Customizing for [Define Scenario and Version](#). Using the `SAP_ICasCustomerImplementation` integration pattern and the implementation of that BAdI allow you to replace the standard implementation for the inspection characteristic integration process with your coding to implement your logic. The BAdI implementation supports inbound integration direction.

This BAdI contains the following method:

- `Maintain`: for the implementation of customer-specific logic for inbound maintain requests.

For more information about the standard settings (filters, single or multiple uses), see the [Enhancement Spot Element Definitions](#) tab in the BAdI Builder (transaction `SE18`).

BAdI settings:

- Single - Use BAdI

Activities

1. Choose [Create](#).
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.20.4.8 BAdI: Production Operation List (POL) Custom Logic

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to implement customer-specific logic for the production operation list (POL) DM object.

```
/PLMF/BD_POL_IP_CUSTOM_IMPL
```

SAP offers standard and custom-specific integration patterns for the production operation list integration process. You can select the needed integration pattern for the production operation list Domain Model type for a used scenario under *Assign Integration Pattern* in the Customizing for *Define Scenario and Version*. Using the `SAP_POLasCustomerImplementation` integration pattern and the implementation of that BAdI allow you to replace the standard implementation for the production operation list integration process with your coding to implement your logic. The BAdI implementation supports inbound integration direction.

This BAdI contains the following method:

- `Maintain`: for the implementation of customer-specific logic for inbound maintain requests.

For more information about the standard settings (filters, single or multiple uses), see the *Enhancement Spot Element Definitions* tab in the BAdI Builder (transaction `SE18`).

BAdI settings:

- Single - Use BAdI

Activities

1. Choose *Create*.
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.20.4.9 BAdI: Product Version (PV) Custom Logic

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to implement customer-specific logic for the product version (PV) DM object.

```
/PLMF/BD_PV_IP_CUSTOM_IMPL
```

SAP offers standard and custom-specific integration patterns for the product version integration process. You can select the needed integration pattern for the defined product version Domain Model type for a used scenario under *Assign Integration Pattern* in the Customizing for *Define Scenario and Version*. Using the `SAP_PVasCustomerImplementation` integration pattern and the implementation of that BAdI allow you to replace the standard implementation for the product version integration with your coding to implement your logic. The BAdI implementation supports inbound and outbound integration directions.

This BAdI contains the following methods:

- **Maintain**: for the implementation of customer-specific logic for inbound maintain requests.
- **Read**: for the implementation of customer-specific logic for inbound read requests and outbound maintain requests.
- **Check_Event_Relev_By_SAP_Obj**: for the implementation of customer-specific logic for outbound maintain requests to determine the custom integration pattern relevance for outbound integration by the SAP object type and SAP object image.
- **Determine_DM_Key_By_SAP_Key**: for the implementation of customer-specific logic for outbound maintain requests to determine the DM key by the SAP key.
- **Determine_Receiver_For_Maint**: for the implementation of customer-specific logic for outbound maintain requests to determine the target systems the object will be integrated with.
- **Get_Event_Data_Type_Name**: for the implementation of customer-specific logic for outbound maintain requests to determine the event data type name for the custom SAP object type.
- **Get_SAP_Object_Type**: for specifying the SAP object type when implementing a custom integration pattern. The SAP object type could be a PLM SAP object type, or the customer-specific SAP object type starting with the prefix z.
- **Get_SAP_Object_Type_Shlp**: for the implementation of customer-specific logic for outbound maintain requests to determine the search help for the custom SAP object type.
- **Process_After_Outbound_Call**: for the implementation of customer-specific logic for outbound maintain requests in the post-processing step (the method is called after the request has been sent).

For more information about the standard settings (filters, single or multiple uses), see the [Enhancement Spot Element Definitions](#) tab in the BAdI Builder (transaction SE18).

BAdI settings:

- Single - Use BAdI

Activities

1. Choose [Create](#).
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.20.4.10 BAdI: Product Version Key Reservation (PV KR) Custom Logic

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to implement customer-specific logic for the product version key reservation (PV KR) DM object.

```
/PLMF/BD_PV_KR_IP_CUSTOM_IMPL
```

SAP offers standard and custom-specific integration patterns for the product version key reservation integration process. You can select the needed integration pattern for the defined product version key

reservation Domain Model type for a used scenario under [Assign Integration Pattern](#) in the Customizing for [Define Scenario and Version](#). Using the `SAP_PVKRasCustomerImplementation` integration pattern and the implementation of that BAdI allow you to replace the standard implementation for the product version key reservation integration process with your own coding to implement your logic. The BAdI implementation supports inbound integration direction.

This BAdI contains the following methods:

- `Maintain`: for the implementation of customer-specific logic for inbound maintain requests.

For more information about the standard settings (filters, single or multiple uses), see the [Enhancement Spot Element Definitions](#) tab in the BAdI Builder (transaction `SE18`).

BAdI settings:

- Single - Use BAdI

Activities

1. Choose [Create](#).
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.20.4.11 BAdI: Structure Version (SV) Custom Logic

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to implement customer-specific logic for the structure version (SV) DM object.

```
/PLMF/BD_SV_IP_CUSTOM_IMPL
```

SAP offers standard and custom-specific integration patterns for the structure version integration process. You can select the needed integration pattern for the defined structure version Domain Model type for a used scenario under [Assign Integration Pattern](#) in the Customizing for [Define Scenario and Version](#). Using the `SAP_SVasCustomerImplementation` integration pattern and the implementation of that BAdI allow you to replace the standard implementation for the structure version integration with your coding to implement your logic. The BAdI implementation supports inbound integration direction.

This BAdI contains the following methods:

- `Maintain`: for the implementation of customer-specific logic for inbound maintain requests.
- `Read`: for the implementation of customer-specific logic for inbound read requests.

For more information about the standard settings (filters, single or multiple uses), see the [Enhancement Spot Element Definitions](#) tab in the BAdI Builder (transaction `SE18`).

BAdI settings:

- Single - Use BAdI

Activities

1. Choose [Create](#).
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.20.4.12 BAdI: Variant Definition (VD) Custom Logic

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to implement customer-specific logic for the variant definition (VD) DM object.

```
/PLMF/BD_VD_IP_CUSTOM_IMPL
```

SAP offers standard and custom-specific integration patterns for the variant definition integration process. You can select the needed integration pattern for the variant definition for a used scenario under [Assign Integration Pattern](#) in the Customizing for [Define Scenario and Version](#). Using the `SAP_VDasCustomerImplementation` integration pattern and the implementation of that BAdI allow you to replace the standard implementation for the variant definition integration process with your coding to implement your logic. The BAdI implementation supports inbound integration direction.

This BAdI contains the following methods:

- `Maintain`: for the implementation of customer-specific logic for inbound maintain requests.

For more information about the standard settings (filters, single or multiple uses), see the [Enhancement Spot Element Definitions](#) tab in the BAdI Builder (transaction `SE18`).

BAdI settings:

- Single - Use BAdI

Activities

1. Choose [Create](#).
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.20.4.13 BAdI: Vendor (VDR) Custom Logic

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to implement customer-specific logic for the vendor (VDR) DM object.

```
/PLMS/BD_VDR_IP_CUSTOM_IMPL
```

SAP offers standard and custom-specific integration patterns for the vendor integration process. You can select the needed integration pattern for the defined vendor Domain Model type for a used scenario under [Assign Integration Pattern](#) in the Customizing for [Define Scenario and Version](#). Using the `SAP_VDRasCustomerImplementationintegration` pattern and the implementation of that BAdI allow you to replace the standard implementation for the vendor integration with your own coding to implement your logic. The BAdI implementation supports inbound and outbound integration directions.

This BAdI contains the following methods:

- `Read`: for the implementation of customer-specific logic for outbound maintain requests.
- `Maintain`: for the implementation of customer-specific logic for inbound maintain requests.
- `Check_Event_Relev_By_SAP_Obj`: for the implementation of customer-specific logic for outbound maintain requests to determine the custom integration pattern relevance for outbound integration by the SAP object type and SAP object image.
- `Determine_DM_Key_By_SAP_Key`: for the implementation of customer-specific logic for outbound maintain requests to determine the DM key by the SAP key.
- `Determine_Receiver_For_Maint`: for the implementation of customer-specific logic for outbound maintain requests to determine the target systems the object will be integrated with.
- `Get_Event_Data_Type_Name`: for the implementation of customer-specific logic for outbound maintain requests to determine the event data type name for the Custom SAP object type.
- `Get_SAP_Object_Type`: for specifying the SAP object type when implementing a custom integration pattern. The SAP object type could be a PLM SAP object type, or the customer-specific SAP object type starting with the prefix `z`.
- `Get_SAP_Object_Type_Shlp`: for the implementation of customer-specific logic for outbound maintain requests to determine the search help for the Custom SAP object type.
- `Process_After_Outbound_Call`: for the implementation of customer-specific logic for outbound maintain requests in the post-processing step (the method is called after the request has been sent).

For more information about the standard settings (filters, single or multiple uses), see the [Enhancement Spot Element Definitions](#) tab in the BAdI Builder (transaction `SE18`).

BAdI settings:

- Single - Use BAdI

Activities

1. Choose [Create](#).
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.20.4.14 BAdI: Variant Option (VO) Custom Logic

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to implement customer-specific logic for the variant option (VO) DM object.

```
/PLMF/BD_VO_IP_CUSTOM_IMPL
```

SAP offers standard and custom-specific integration patterns for the variant option integration process. You can select the needed integration pattern for the variant option for a used scenario under [Assign Integration Pattern](#) in the Customizing for [Define Scenario and Version](#). Using the `SAP_VOasCustomerImplementation` integration pattern and the implementation of that BAdI allow you to replace the standard implementation for the variant option integration process with your coding to implement your logic. The BAdI implementation supports inbound integration direction.

This BAdI contains the following methods:

- `Maintain`: for the implementation of customer-specific logic for inbound maintain requests.

For more information about the standard settings (filters, single or multiple uses), see the [Enhancement Spot Element Definitions](#) tab in the BAdI Builder (transaction `SE18`).

BAdI settings:

- Single - Use BAdI

Activities

1. Choose [Create](#).
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.20.4.15 BAdI: Variant Option Rule (VOR) Custom Logic

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to implement customer-specific logic for the variant option rule (VOR) DM object.

```
/PLMF/BD_VOR_IP_CUSTOM_IMPL
```

SAP offers standard and custom-specific integration patterns for the variant option rule integration process. You can select the needed integration pattern for the variant option rule domain model type for a used scenario under [Assign Integration Pattern](#) in the Customizing for [Define Scenario and Version](#). Using the `SAP_VORasCustomerImplementation` integration pattern and the implementation of that BAdI allow you to replace the standard implementation for the variant option rule integration process with your coding to implement your logic. The BAdI implementation supports inbound integration direction.

This BAdI contains the following method:

- `Maintain`: for the implementation of customer-specific logic for inbound maintain requests.

For more information about the standard settings (filters, single or multiple uses), see the [Enhancement Spot Element Definitions](#) tab in the BAdI Builder (transaction `SE18`).

BAdI settings:

- Single - Use BAdI

Activities

1. Choose *Create*.
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

3.20.4.16 BAdI: View Structure Version (VSV) Custom Logic

This Business Add-In (BAdI) is used in the component PLM System Integration (PLMSI). You can use this BAdI to implement customer-specific logic for the view structure version (VSV) DM object.

```
/PLMS/BD_VSV_IP_CUSTOM_IMPL
```

SAP offers standard and custom-specific integration patterns for the view structure version integration process. You can select the needed integration pattern for the defined view structure version Domain Model type for a used scenario under *Assign Integration Pattern* in the Customizing for *Define Scenario and Version*. Using the `SAP_VSVasCustomerImplementation` integration pattern and the implementation of that BAdI allow you to replace the standard implementation for the view structure version integration with your coding to implement your logic. The BAdI implementation supports inbound and outbound integration directions.

This BAdI contains the following methods:

- `Read`: for the implementation of customer-specific logic for outbound maintain requests.
- `Maintain`: for the implementation of customer-specific logic for inbound maintain requests.
- `Check_Event_Relev_By_SAP_Obj`: for the implementation of customer-specific logic for outbound maintain requests to determine the custom integration pattern relevance for outbound integration by the SAP object type and SAP object image.
- `Determine_DM_Key_By_SAP_Key`: for the implementation of customer-specific logic for outbound maintain requests to determine the DM key by the SAP key.
- `Determine_Receiver_For_Maint`: for the implementation of customer-specific logic for outbound maintain requests to determine the target systems the object will be integrated with.
- `Get_Event_Data_Type_Name`: for the implementation of customer-specific logic for outbound maintain requests to determine the event data type name for the custom SAP object type.
- `Get_SAP_Object_Type`: for specifying the SAP object type when implementing a custom integration pattern. The SAP object type could be a PLM SAP object type, or the customer-specific SAP object type starting with the prefix `Z`.
- `Get_SAP_Object_Type_Shlp`: for the implementation of customer-specific logic for outbound maintain requests to determine the search help for the custom SAP object type.
- `Process_After_Outbound_Call`: for the implementation of customer-specific logic for outbound maintain requests in the post-processing step (the method is called after the request has been sent).

This BAdI has no default implementation. For more information about the standard settings (filters, single or multiple uses), see the *Enhancement Spot Element Definitions* tab in the BAdI Builder (transaction `SE18`).

BAdI settings:

- Single - Use BAdI

Activities



1. Choose *Create*.
2. Enter the custom enhancement implementation you have created.
3. Save your entries.

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.

About the icons:

- Links with the icon : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
 - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
 - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon : You are leaving the documentation for that particular SAP product or service and are entering an SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.

The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Bias-Free Language

SAP supports a culture of diversity and inclusion. Whenever possible, we use unbiased language in our documentation to refer to people of all cultures, ethnicities, genders, and abilities.

© 2025 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.

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.

Please see <https://www.sap.com/about/legal/trademark.html> for additional trademark information and notices.