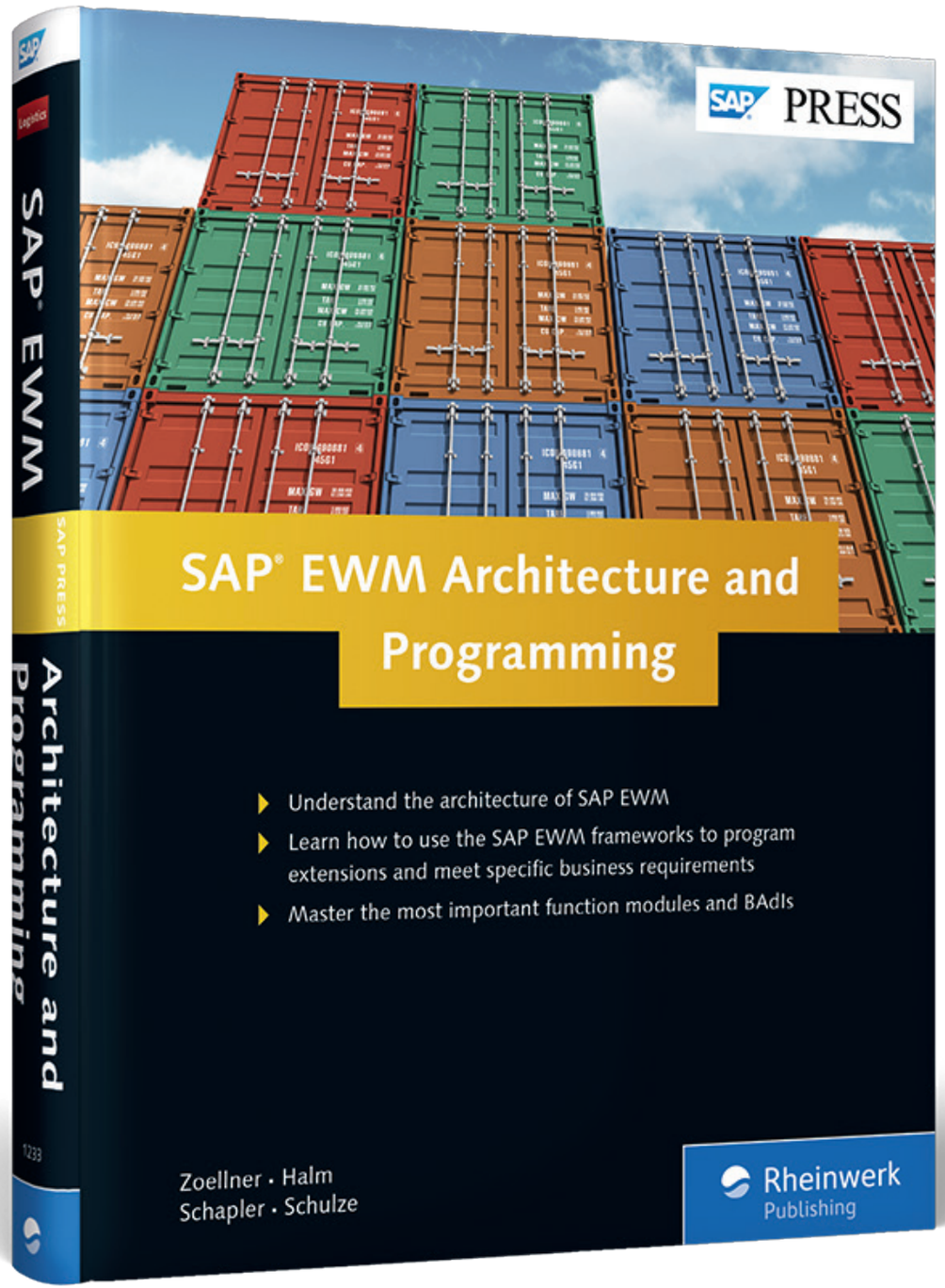
SAP PRESS
First-hand knowledge.
Rheinwerk Publishing



Reading Sample

This selection covers the architecture of SAP EWM, focusing on the delivery processing component, including storage and retrieval, stock transfer, and order management, as well as the warehouse logistics component, including shipping and receiving, warehouse tasks, and warehouse orders.

"Delivery Processing"
"Warehouse Logistics"

Contents

Index

The Authors

Peter Zoellner, Robert Halm, Daniela Schapler, Karen Schulze

SAP EWM Architecture and Programming

494 Pages, 2016, $79.95/€79.95
ISBN 978-1-4932-1233-0

www.sap-press.com/3860

In order to program in SAP EWM, we must first understand the individual software components that are part of the installation in the SAP back-end system. This chapter describes these components in terms of their function and objects, and concludes by discussing their integration with SAP ERP.

2 Architecture

The architecture of SAP Extended Warehouse Management (SAP EWM) is characterized by a variety of application components and objects that exhibit the rich functionality of the system itself. Some of these components will be discussed in detail with respect to their function and their properties in the following sections of this chapter:

- **Delivery processing** (see Section 2.1)
  The delivery process receives requests for storage and retrieval, as well as requirements for the transfer of stock, which result from the order management. It is therefore closely linked to the SAP ERP system and the components "Warehousing" and "Transportation Management."
- **Warehouse logistics** (see Section 2.2)
  The warehouse logistics provides a multiplicity of functions that are necessary for the planning and execution of activities within the warehouse.
- **Inventory management** (see Section 2.3)
  The inventory management allows the management of stock in different spatial and physical states to the level of storage space and parking unit. It is closely linked with inventory management of SAP ERP.
- **Quality management** (see Section 2.4)
  For quality control of inbound goods, SAP EWM provides a close link to Quality Inspection Engine (QIE).
- **Integration with SAP ERP** (see Section 2.5)
  Integration with SAP ERP can be considered a cross-sectional component of SAP EWM. This section presents the numerous interfaces for the interaction between SAP ERP and SAP EWM before they are bundled.

Other components such as labor management, exception handling, master data, and integration with other SAP and non-SAP systems are not considered in this chapter, but they are rudimentarily dotted throughout the other chapters.

2.1 Delivery Processing

The *delivery* in its various forms is one of the key objects in SAP EWM. In this section, we explain how this complex object is constructed.

In general, we can say that for SAP EWM every delivery is a warehouse request (sooner or later), but not every warehouse request is a delivery. You will notice in SAP EWM that *warehouse request* is used as the generic term for delivery. However, within this section, we try to keep them separated to introduce you to different use cases and to avoid confusion.

Economically, the inventory inflows and outflows are usually booked on inbound deliveries and outbound deliveries in SAP EWM. If you have integrated an SAP ERP system to SAP EWM, those deliveries in SAP EWM are always created with reference to a corresponding SAP ERP document.

The delivery in SAP EWM is modeled in several different documents. Depending on function and processing phases, different forms of documents are used that will be categorized by the predefined document categories. We distinguish between *delivery documents*, which from the perspective of communication with SAP ERP or other legacy SAP ERP systems are essential, and warehouse requests (working documents), which are mainly used for processing in SAP EWM.

Table 2.1 provides an overview of all document categories in the delivery/warehouse request processing used in SAP EWM and calls the central transaction for each of them.

Document category	Description	Category	SAP EWM Transaction
IDR	Inbound Delivery Notification	Delivery document	/SCWM/IDN
ODR	Outbound Delivery Request	Delivery document	/SCWM/ODR

Table 2.1 Overview of Document Categories and Associated /SCWM / UI Transactions

Document category	Description	Category	SAP EWM Transaction
POR	Posting Change Request	Delivery document	/SCWM/IM_DR
GRN	Expected Goods Receipt Notification	Delivery document	/SCWM/GRN
PDI	Inbound Delivery	Warehouse request	/SCWM/PRDI
PDO	Outbound Delivery Order	Warehouse request	/SCWM/PRDO
SPC	Posting Change	Warehouse request	/SCWM/IM_PC
EGR	Expected Goods Receipt	Delivery document	/SCWM/GRPE /SCWM/GRPI
WMR	Stock Transfer	Warehouse request	/SCWM/IM_ST
FDO	Outbound Delivery	Delivery document	/SCWM/FD
PWR	Production Material Request	Warehouse request	/SCWM/PMR

Table 2.1 Overview of Document Categories and Associated /SCWM / UI Transactions (Cont.)

In the following section we describe how delivery documents and warehouse requests are arranged in the logistics overall flow and give you an overview of the key technical objects.

Production Material Request [«]

Since SAP EWM release 9.2, the *Production Material Request* (PWR) was introduced as part of the *Advanced Production Integration*. The PWR actually does not really fit into the categories of delivery document or warehouse request. The PWR originates from the production order in SAP ERP. However, the distribution of this object does not use the delivery interfaces and also the strict update restrictions for deliveries from SAP ERP do not apply. Since the PWR is technically treated as a warehouse request in SAP EWM—more than a delivery document—we decided to categorize it as a warehouse request within this section.

2.1.1 Function of the Delivery Processing

The delivery processing provides a number of functions to perform various logistical activities in the warehouse. In particular, it has the task of handling inbound and outbound postings and triggering the feedback to the respective SAP ERP system.

The delivery environment uses documents that are tailor-made for the particular business context. We will elaborate on these different documents in this section. In the documents used, we distinguish between *delivery documents* and *warehouse requests* (working documents).

When distributing SAP ERP deliveries to SAP EWM, the *delivery documents* are created, from which in turn the warehouse requests are produced using a transfer service class `/SCDL/CL_TS_MANAGEMENT` (method `TRANSFER_OBJ`). Delivery documents are used within the warehouse management with SAP EWM as a basis for the imaging and processing of inbound and outbound processes. On the one hand, these documents include the data that is necessary for the planning and scheduling of warehouse activities. On the other hand, data on actions that were carried out by the system during the delivery processing are updated in these documents. Thus you always find an overview of the current processing progress of the relevant document in the system.

The documents to which activities can be carried out within the warehouse (e.g., packaging, putaway, picking, etc.) we call *warehouse requests* (working documents). Most warehouse requests have preceding documents. The preceding documents are a kind of template for the creation of warehouse requests. That is, based on the data of the preceding document, the SAP EWM-specific information from customizing (for example, document categories, item categories, warehouse process types, etc.) and master data (e.g., warehouse product, storage bin, batch, etc.) will be read in order to enrich the warehouse request accordingly.

For a schematic overview of the relationship between delivery documents and warehouse requests, see Figure 2.1; it shows what delivery objects from an SAP ERP system produce which delivery documents and the resulting warehouse requests according to the logistical process.

Regarding the technical document architecture, however, there is no difference between warehouse requests and delivery documents. Technically, all documents in the delivery environment are built identically.

[»] **Delivery Processing in SAP EWM**

For more information about supported processes and requirements for delivery processing in SAP EWM and references to the configuration options, see SAP Help Portal (*http://help.sap.com/EWM92*).

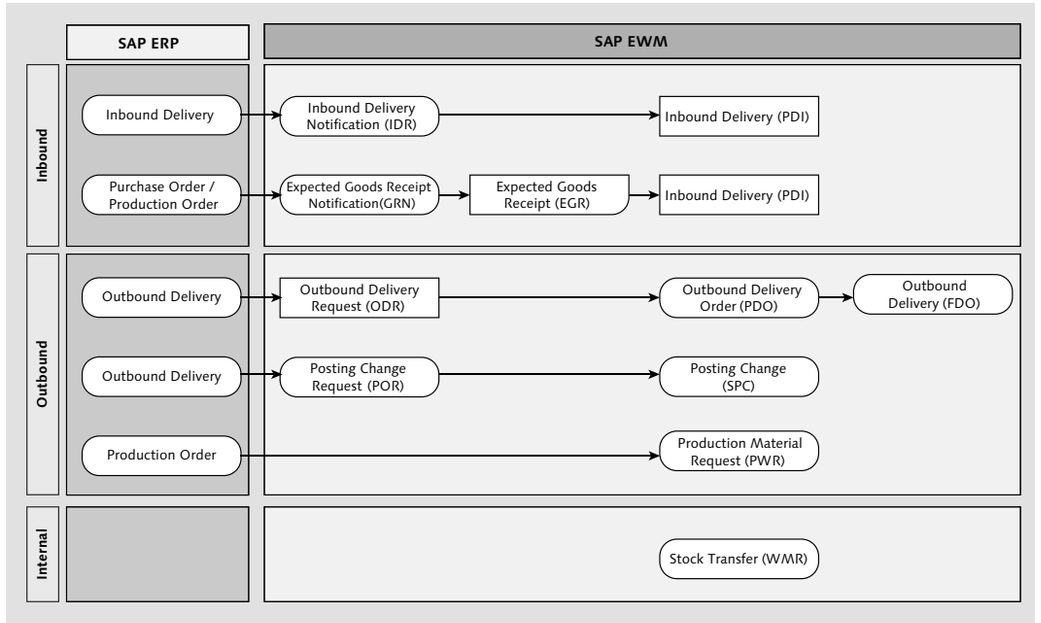


Figure 2.1 Relationship between Delivery Documents and Warehouse Requests

2.1.2 Object Warehouse Request and Data Model

In SAP EWM, the deliveries and warehouse requests are modeled as ABAP Objects. By reading deliveries, the objects (instances) are created that represent the respective deliveries. This object instance can then be accessed by writing, for example, to change delivery dates.

In addition to pure reading of database data, yet more automatic processes occur during the reading process or while the instances are created; among other functions, determinations and validations are performed on the object instances to set dynamic data (for example, aggregated quantities, status, etc.).

Business Object Processing Framework

In this context, we briefly explain the *Business Object Processing Framework* (BOPF) and the way determinations, validations, and actions to SAP EWM delivery objects are used. The BOPF is in its present form a framework for the implementation of business objects, following the principles of *Service-Oriented Architecture* (SOA).

It contains the necessary functions for implementation and supports a uniformly usable modeling standard. The SOA concept is based basically on Enterprise Services, which underlie a strict governance process to ensure the necessary uniform business semantics of existing applications and beyond. The Enterprise Services delivered by SAP are structured along business processes and can be assembled into an automated flow.

The BOPF controls the business logic of the application and covers the deployment, the buffering, and the storage of data.

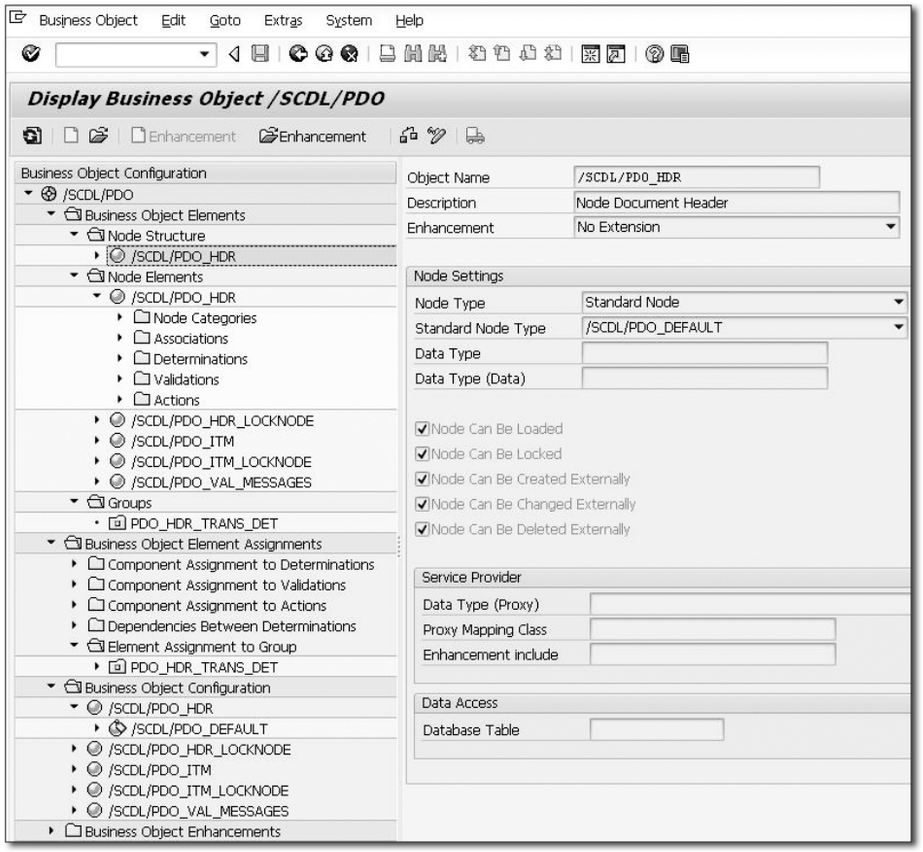


Figure 2.2 BOPF UI to Object /SCDL/PDO

Hence business logic and data management are strictly separated, as well as modification and checking of the data managed. From the software *application*

platform (AP) layer perspective, today's BOPF, as it is used in various SAP products (e.g., SAP Transportation Management), is no longer compatible with the SAP EWM BOPF; however, the basics are still recognizable in delivery processing. Via Transaction /BOPF/CONF_UI, an insight into the clearly structured modeling of the various document categories of SAP EWM delivery processing is provided (see Figure 2.2).

Construction of the Delivery

A delivery always consists of a delivery header and at least one delivery item. The technical relationship is produced by the key holding the `DOCID` (unique internal header identification) and the `ITEMID` (unique internal item identification). In addition, other properties are assigned by default as well:

- Status
- Dates
- Reference documents
- Aggregated quantities
- Locations
- Business partners

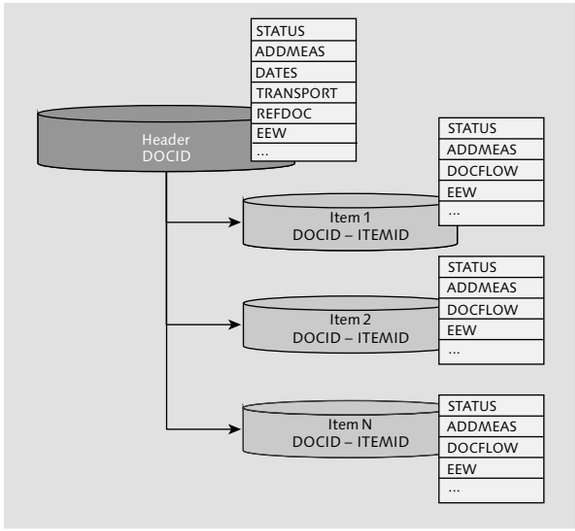


Figure 2.3 Construction of a Delivery Object in SAP EWM

Additional properties that you can use optionally, such as handling units, transportation units, texts, etc., are also assigned from the corresponding applications to the delivery document. Figure 2.3 shows the construction of the delivery.

The document categories at the header level are the item categories at the item level. These attributes are the basis on which the business properties of each position are categorized.

There are the following seven types:

- DLV: Standard Delivery Item
- PAC: Packing Item
- RET: Returns Item
- TXT: Text Item
- VAL: Value Item
- CMP: Component (PWR)
- OCP: Outbound Component (PWR)

In addition, at the item level, among others, the following hierarchy elements are distinguished:

- Main Item (DSP)
- Batch Split Item (BSP)
- Delivery Split Item (FSD)

The split items can occur in SAP EWM, for example, when you post the goods issue only for a specific item and not for the entire delivery (delivery split) or when you pick different batches for one single main item (batch split). Split items are always in a fixed hierarchy to the main item.

Table 2.2 shows the main database tables for header and item data in the delivery processing. For more database tables on the objects mentioned, you can check the package `/SCDL/DATA_MODEL`.

Database Table	Description
`/SCDL/DB_REQH`	Inbound Delivery Notification / Outbound Delivery Request: Header
`/SCDL/DB_REQI`	Inbound Delivery Notification/ Outbound Delivery Request: Item

Table 2.2 Important Database Tables of Delivery

Database Table	Description
`/SCDL/DB_PROCH_I`	Inbound Delivery: Header
`/SCDL/DB_PROCI_I`	Inbound Delivery: Item
`/SCDL/DB_PROCH_O`	Outbound Delivery Order: Header
`/SCDL/DB_PROCI_O`	Outbound Delivery Order: Item
`/SCDL/DB_DLVH_O`	Outbound Delivery: Header
`/SCDL/DB_DLVI_O`	Outbound Delivery: Item
`/SCDL/DB_PROCH_P`	Production Material Request: Header
`/SCDL/DB_PROCI_P`	Production Material Request: Item

Table 2.2 Important Database Tables of Delivery (Cont.)

The structure and the concept of SAP EWM delivery processing allows us to access individual items and extrapolate them without locking the entire warehouse request. This has, in practice, the great advantage of allowing different items of a warehouse request to be worked in parallel—for example, during picking—and still no lock conflicts will be triggered.

From a performance point of view, this also results in a further advantage: processes within the delivery processing and the work at item level only load the respective item and not all of the other items of the relevant warehouse request in the memory.

Service Provider and Aspects

In order to allow all applications in SAP EWM to access a delivery object via a central interface, the *Service Provider* (SP) is used. These SPs provide interface methods that enable the application to read, write, and execute actions to a specific *business object* (BO). Also, the SPs serve to return the requested data from the BO well prepared in the respective structures of the calling application. This architecture allows a generic use of the BO, tailored to the specific requirements of each SAP EWM application.

The two central SPs in SAP EWM are:

- `/SCDL/CL_SP`: SP of the delivery
- `/SCWM/CL_SP`: SP of the *user interface* (UI)

These abstract classes, with their interfaces, build the foundation for the special use cases within the delivery processing. Here the main difference between /SCDL/CL_SP and /SCWM/CL_SP is that the SPs of the UI—depending on the particular transaction—hide diverse fields or components of the BO and they perform additional checks (ABAP logics) before updating the object. Furthermore, the object data through the UI SP will also be enriched by short descriptions. The various forms of the delivery SP are shown graphically in Figure 2.4.

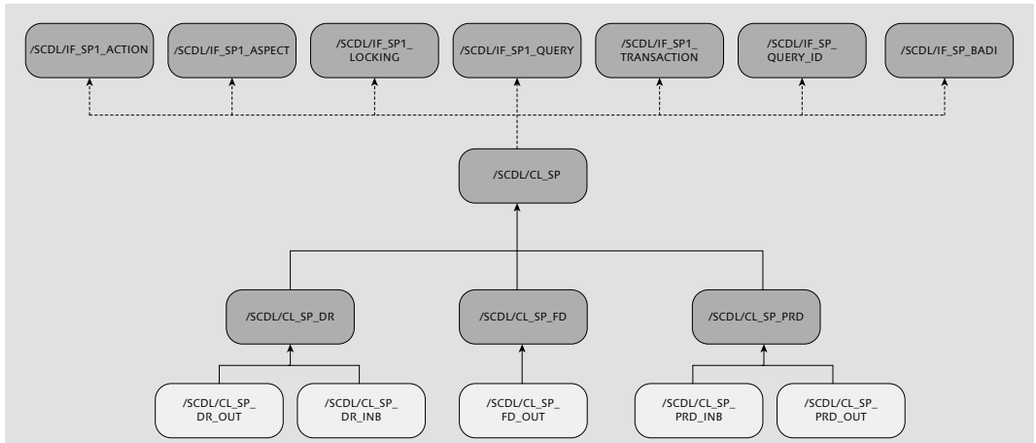


Figure 2.4 Architecture of the SPs /SCDL CL_SP

The structures by which the SPs communicate with the application programs are called aspects. An aspect represents a specific attribute of the BO. There is, for example, an aspect for dates data at header level (/SCDL/S_SP_A_HEAD_DATE) and an aspect for dates data at item level (/SCDL/S_SP_A_ITEM_DATE).

In the architecture of the Enterprise Service we distinguish between *key aspects* and *aspects*. Here, the key aspect always contains the semantic key of an object (similar to a primary key). All aspects must be assigned to a key aspect to be clearly identifiable.

There are no specific repository objects to the respective aspects, so to accommodate the highly dynamic approach of this architecture, the caller has to ensure that the desired context will be passed when calling the SP methods. The appropriate definitions can be found as constants within the interfaces /SCDL/IF_SP_C

and /SCWM/IF_SP_C. The coding example in Listing 2.1 will give you an example of how to use the most important aspects and SP methods. First, we create objects with reference to the classes /SCDL/CL_SP_PRD_OUT and /SCDL/CL_SP_MESSAGE_BOX. We use class /SCDL/CL_SP_PRD_OUT because we will select an outbound delivery order. The message box will be needed to capture any messages, for example, in case of errors.

```
DATA: lo_sp          TYPE REF TO /scdl/cl_sp_prd_out,
      lo_message_box TYPE REF TO /scdl/cl_sp_message_box.
TRY.
    CREATE OBJECT lo_message_box.
    CREATE OBJECT lo_sp
      EXPORTING
        io_message_box = lo_message_box
        iv_doccat      = /scdl/if_dl_doc_c=>sc_doccat_out_prd
        iv_mode        = /scdl/cl_sp=>sc_mode_classic.
ENDTRY.
```

Listing 2.1 Create Object Instance

Now we select an ITEM aspect using an SAP ERP delivery number and the item number. We illustrate how this works in Listing 2.2. In the following section, we explain what is happening with the complex selection criteria.

```
DATA: ls_options      TYPE /scdl/s_sp_query_options,
      ls_inparam      TYPE /scdl/s_sp_q_head,
      lt_selections   TYPE /scdl/t_sp_selection,
      ls_selections   TYPE /scdl/s_sp_selection,
      lt_a_item       TYPE /scdl/t_sp_a_item,
      lv_rejected     TYPE boole_d,
      lt_return_codes TYPE /scdl/t_sp_return_code,
      lt_messages     TYPE /scdl/dm_message_tab.

ls_selections-fieldname = /scdl/if_dl_logfname_c=>sc_refdocno_erp_h.
ls_selections-sign = 'I'.
ls_selections-option = 'EQ'.
ls_selections-low = '8040000094'.
APPEND ls_selections TO lt_selections.

CLEAR ls_selections.
ls_selections-fieldname = /scdl/if_dl_logfname_c=>sc_refitemno_erp_i.
ls_selections-sign = 'I'.
ls_selections-option = 'EQ'.
ls_selections-low = '10'.
APPEND ls_selections TO lt_selections.
```

```
lo_sp->query( EXPORTING
  query = /scdl/if_sp_c=>sc_qry_item
  selections = lt_selections
IMPORTING
  outrecords = lt_a_item
  rejected = lv_rejected ).
IF lv_rejected = abap_true.
  lt_messages = lo_message_box->get_messages( ).
  CALL METHOD /scwm/cl_tm=>cleanup( ).
  EXIT.
ENDIF.
```

Listing 2.2 Example Calling "/SCDL/IF_SP1_QUERY~EXECUTE"

The returning value `lt_a_item` holds both the `DOCID` and the `ITEMID`. Both values are needed to continue using the key aspect `lt_sp_k_item`. Next we can select the aspect for our item, which includes the delivery terms (see Listing 2.3).

```
DATA: ls_sp_k_item       TYPE /scdl/s_sp_k_item,
      lt_sp_k_item       TYPE /scdl/t_sp_k_item,
      lt_a_item_delterms TYPE /scdl/t_sp_a_item_delterm.
FIELD-SYMBOLS:
<fs_a_item> TYPE /scdl/s_sp_a_item.

READ TABLE lt_a_item ASSIGNING <fs_a_item> INDEX 1.
IF sy-subrc EQ 0.
  MOVE-CORRESPONDING <fs_a_item> TO ls_sp_k_item.
  APPEND ls_sp_k_item TO lt_sp_k_item.
ENDIF.

lo_sp->select( EXPORTING
  inkeys = lt_sp_k_item
  aspect = /scdl/if_sp_c=>sc_asp_item_delterm
IMPORTING
  outrecords = lt_a_item_delterms
  rejected = lv_rejected
  return_codes = lt_return_codes ).
IF lv_rejected = abap_true.
  lt_messages = lo_message_box->get_messages( ).
  CALL METHOD /scwm/cl_tm=>cleanup( ).
  EXIT.
ELSEIF lt_return_codes IS NOT INITIAL.
  READ TABLE lt_return_codes TRANSPORTING NO FIELDS
  WITH KEY failed = 'X'.
  IF sy-subrc IS INITIAL.
    lt_messages = lo_message_box->get_messages( ).
    CALL METHOD /scwm/cl_tm=>cleanup( ).
    EXIT.
  ENDIF.
ENDIF.
```

Listing 2.3 Example Calling "/SCDL/IF_SP1_ASPECT ~ SELECT"

If you have extended the delivery item with its own customer fields and you want to read or update these fields, you use the aspect `SC_ASP_ITEM_EEW_PRD` instead of `SC_ASP_ITEM_DELTERM`. If you do not want to provide a specific item of the delivery or if you first want to determine the corresponding items, you use the call in Listing 2.4.

```
lo_sp->select_by_relation( EXPORTING
  relation = /scdl/if_sp_c=>sc_rel_head_to_item
  inrecords = lt_sp_k_head
  aspect = /scdl/if_sp_c=>sc_asp_head
IMPORTING
  outrecords = lt_a_item
  rejected = lv_rejected
  return_codes = lt_return_codes ).
```

Listing 2.4 Example Calling "/SCDL/IF_SP1_ASPECT~SELECT_BY_RELATION"

We recommend using aspects only if you have to select certain pieces of information of a delivery object or if you want to update certain customer fields. You should have these features override standard values only in exceptional cases and only if you are absolutely sure what effect this will have, otherwise, data inconsistencies may occur.

In general, however, a delivery object needs various information from various aspects. To this end, there are other functions, which we will explain in the next section.

Additional Information [«]

- For more information about the data model, as well as the technical objects, please check the SAP Service Marketplace (*http://service.sap.com*).
- In the guidelines attached to SAP Note 1414179, there are detailed technical descriptions on the used class interfaces and the service classes of delivery processing, as well as numerous coding examples.

Using the Delivery Query

To select delivery documents or warehouse requests without having to use individual aspects directly, SAP provides the method QUERY. This method is part of the three main classes of the object manager /SCWM/CL_DLV_MANAGEMENT. Depending on the document category you want to select, you will need to use the designated class. Inbound delivery notifications, for example, are read with a class other than inbound deliveries. Table 2.3 shows the class reference you should use in order to select the objects according to each document category using the method QUERY.

Document category	Class
IDR	/SCWM/CL_DLV_MANAGEMENT_DR
ODR	/SCWM/CL_DLV_MANAGEMENT_DR
POR	/SCWM/CL_DLV_MANAGEMENT_DR
EGR	/SCWM/CL_DLV_MANAGEMENT_DR
PDI	/SCWM/CL_DLV_MANAGEMENT_PRD
PDO	/SCWM/CL_DLV_MANAGEMENT_PRD
SPC	/SCWM/CL_DLV_MANAGEMENT_PRD
WMR	/SCWM/CL_DLV_MANAGEMENT_PRD
FDO	/SCWM/CL_DLV_MANAGEMENT_FD
PWR	/SCWM/CL_DLV_MANAGEMENT_PRD

Table 2.3 Document Categories and Associated Query Classes

The coding example in Listing 2.5 will help you understand how you can use the delivery query to read the header and item information of an outbound delivery order for an SAP ERP delivery number on complex selection criteria (LT_SELECTION).

```
DATA:
  lo_delivery       TYPE REF TO /scwm/cl_dlv_management_prd,
  lo_message        TYPE REF TO /scwm/cl_dm_message_no,
  lt_message        TYPE /scdl/dm_message_tab,
  lx_delivery       TYPE REF TO /scdl/cx_delivery,
  lt_selection      TYPE TABLE OF /scwm/dlv_selection_str,
  ls_selection      TYPE /scwm/dlv_selection_str,
  ls_addmeas_detail TYPE /scdl/dl_addmeas_key_str,
  ls_read_options   TYPE /scwm/dlv_query_contr_str,
  ls_include        TYPE /scwm/dlv_query_incl_str_prd,
  lt_headers        TYPE /scwm/dlv_header_out_prd_tab,
  lt_items          TYPE /scwm/dlv_item_out_prd_tab.
```



```
* get instance
lo_delivery = /scwm/cl_dlv_management_prd=>get_instance( ).

* build up range of select options
ls_selection-fieldname =
/scdl/if_dl_logfname_c=>sc_refdocno_erp_h.
ls_selection-sign = 'I'.
ls_selection-option = 'EQ'.
ls_selection-low = '8040000090'.
APPEND ls_selection TO lt_selection.

* set read options
ls_read_options-data_retrival_only = abap_true.
ls_read_options-mix_in_object_instances = /scwm/if_dl_c=>sc_mix_in_
load_instance.

ls_include-item_addmeas = abap_true.
ls_include-item_status = abap_true.
ls_include-item_hierarchy = abap_true.

* call DLV query
TRY.
    CALL METHOD lo_delivery->query
      EXPORTING
        iv_doccat       = /scdl/if_dl_c=>sc_doccat_out_prd
        it_selection    = lt_selection
        is_read_options = ls_read_options
        is_include_data = ls_include
      IMPORTING
        et_headers      = lt_headers
        et_items        = lt_items
        eo_message      = lo_message.
    IF lo_message IS BOUND.
      lt_message = lo_message->get_messages( ).
    ENDIF.
  CATCH /scdl/cx_delivery INTO lx_delivery.
    IF lx_delivery->mo_message IS BOUND.
      lo_message->add( lx_delivery->mo_message ).
    ENDIF.
ENDTRY.
```

Listing 2.5 Example Calling "Delivery Query"

If you would like to use complex selection criteria, you must know the association between the corresponding database fields—in our example, the SAP ERP delivery number—and the logical field names. For this mapping, see the IMG on EWM • Cross-Process Settings • Delivery Processing • Extend Delivery Processing • Define Logical Field Names.

Alternatively, the constants for the logical field names are to be found in the following interfaces:

- /SCDL/IF_DL_LOGFNAME_C
- /SCWM/IF_DL_LOGFNAME_C

As a result, the system provides even further delivery-related information next to the header or item data. Depending on the class used, this can be, for example, assigned handling units or transportation units.

The selection of delivery objects can have a strong impact on the performance of a process in certain circumstances. That depends not only on how many documents and items there are to be read but also on what states or values from the database information must be calculated at runtime. Special header and item information—for example, status values and quantity roles—are dynamic (transient state) and have to be calculated at runtime based on other static values (persistent state). You can find an overview of the characteristics of a status here in IMG EWM • Cross-Process Settings • Delivery Processing • Status Management • Define Status Profiles.

Note that there could be different configurations to a status at header and item level. For example, for outbound delivery orders, the status DPI (Picking) at item level is persistent. Whereas at the header level, it is calculated at runtime from the status of the items; therefore, the status here is transient.

In order to control the selection behavior of the delivery query to the specific needs, SAP delivers the following structures as import parameters:

- IS_READ_OPTIONS – controls the selection behavior
- IS_INCLUDE_DATA – include certain aspects
- IS_EXCLUDE_DATA – exclude certain aspects

The IS_EXCLUDE_DATA parameter is, however, obsolete and only mentioned here to provide comprehensive information. You should not use it in implementations.

In your programming, you should closely examine in each context exactly what you need the delivery object for (read or write accesses). In addition, check if the data in the selection result are really necessary.

Query Control during Data Acquisition [Ex]

In an evaluation for special deliveries, you want to determine for which of these documents the goods issue has already been posted. The status to check for at item level is called DGI and is persistent. You call therefore the delivery query for a read-only access and you need only ET_ITEMS as a return parameter. A creation of an object instance and the calculation of dynamic values are not necessary. Accordingly, you can use the indicator DATA_RETRIVAL_ONLY in the structure IS_READ_OPTIONS and you will set structural value IS_INCLUDE_DATA-ITEM_STATUS.

A detailed description of how to use the special reading options can be found in the documentation of the method QUERY in the class /SCWM/CL_DLV_MANAGEMENT_PRD. In any case, you should always set the appropriate reading options before every call of the delivery query due to performance reasons.

Another fundamental part of deliveries are the respective item quantities. SAP EWM provides various functions for editing delivery items, which the status management cannot solely manage. For example, one would like to know exactly how much has not yet been posted by the system during a partial goods receipt—that means how much is still open. This aspect will be displayed using single quantity roles.

You can find the current quantities of a delivery item in the delivery UI under the table tab ADDL QUANTITIES. Similar to the status management, the quantity offsetting also contains persistent and transient values. For example, the quantity role W1 (Picking Planned) will be calculated from the document flow of an item at runtime.

The delivery document flow is technically a separate framework, which is why we will not go into it in greater detail at this point. Basically, it represents node relationships that contain information on specific actions to a delivery item. The document flow used a very flat database layer (/1DF/* tables) up until SAP EWM release 7.02. Release SAP EWM 9.0 introduced the advanced document flow. Thus here the information is not stored in the /1DF/* tables anymore but in a few generic ones (one/two) to improve performance.

An overview of the properties of all quantities roles as well as their determination, see the IMG EWM • Cross-Process Settings • Delivery Processing • Quantity Offsetting.

The example in Listing 2.6 shows how to evaluate item hierarchy, status, and quantity roles. We use the already selected table `lt_items` and examine first whether there are delivery split items.

```
DATA: ls_addmeas TYPE /scdl/dl_addmeas_str,
      ls_status  TYPE /scdl/dl_status_str.
DATA: ls_hierarchy TYPE /scdl/dl_hierarchy_str,
      lv_cat       TYPE /scdl/dl_hierarchy_category,
      lv_skip      TYPE boolean,
      lo_corr      TYPE REF TO /scwm/cl_dlv_correlation.
FIELD-SYMBOLS: <item>   TYPE /scwm/dlv_item_out_prd_str.

LOOP AT lt_items ASSIGNING <item>.
* check hierarchy
  lo_corr = /scwm/cl_dlv_correlation=>get_instance( ).
  LOOP AT <item>-hierarchy INTO ls_hierarchy.
    CLEAR lv_skip.
    TRY.
        CALL METHOD lo_corr->get_hier_cat
          EXPORTING
            iv_hierarchy_type = ls_hierarchy-hierarchy_type
          IMPORTING
            ev_hierarchy_cat  = lv_cat.
      CATCH /bopf/cx_frw .
    ENDTRY.

* skip split items
    IF lv_cat = /scdl/if_dl_hierarchy_c=>sc_cat_ssp AND
       ls_hierarchy-parent_object IS NOT INITIAL.
      lv_skip = abap_true.
      EXIT.
    ENDIF.
  ENDLOOP.

  IF lv_skip = abap_true.
    DELETE lt_items.
    CONTINUE.
  ENDIF.
```

Listing 2.6 Evaluate Delivery Item Hierarchy

Having sorted out all the items that have already been posted to a goods issue, we now look at the status `DPI` (Picking) for the item that has not yet been completely picked (see Listing 2.7).

```
  READ TABLE <item>-status INTO ls_status
      WITH KEY
        status_type  = /scdl/if_dl_status_c=>sc_t_picking.
```



```
  IF ls_status-status_value = /scdl/if_dl_status_c=>sc_v_not_relevant.
    DELETE lt_items.
    CONTINUE.
  ELSEIF ls_status-status_value NE /scdl/if_dl_status_c=>sc_v_finished.
    MESSAGE e012(zySAP EWM) WITH
    <item>-itemno. "not yet picked completely
    CONTINUE.
  ENDIF.
```

Listing 2.7 Evaluate Delivery Item Status

Finally, we check on the quantity role `PA` (Pack) for the item that has not yet been completely packed after picking (see Listing 2.8).

```
  READ TABLE <item>-addmeas INTO ls_addmeas
  WITH KEY
  qty_role = /scdl/if_dl_addmeas_c=>sc_qtyrole_pack
  qty_category = /scdl/if_dl_addmeas_c=>sc_qtycat_open.
  IF sy-subrc = 0 AND ls_addmeas-qty NE 0.
    MESSAGE e013(zyewm) WITH
    <item>-itemno. "not yet packed completely
  ENDIF.
ENDLOOP.
```

Listing 2.8 Evaluate Delivery Item Quantity Role

This section provided an overview with the help of some practical examples about how to select the delivery objects and to perform certain checks according to various criteria.

2.1.3 Integration with Other SAP EWM Components

In Section 2.1.1, we outlined the central functionality of the delivery processing, and in Section 2.1.2, we described how SPs of the respective /SCWM/ application programs manage and edit the delivery objects. The most important one of these SAP EWM components in this context is the WT processing for warehouse requests. Through this component, the special WTs for putaway and picking will be created with reference to a warehouse request. In contrast to that, there are also WTs not related to a warehouse request. But for these tasks, we use other functions of the WT processing.

The following coding example (see Listing 2.9) shows how easy it is to use the function module `/SCWM/TO_CREATE_WHR` to create the WTs for warehouse requests. Again we use the already selected table `lt_items`. As we work in the field of

/SCWM/ components, we also need to pass the appropriate warehouse number (lv_lgnum).

```
DATA: ls_create_whr TYPE /scwm/s_to_prepare_whr_int,
      lt_create_whr TYPE /scwm/tt_to_prep_whr_int,
      lv_lgnum      TYPE /scwm/lgnum,
      lv_tanum      TYPE /scwm/tanum,
      lt_ltap_vb    TYPE /scwm/tt_ltap_vb,
      lt_bapiret    TYPE bapirettab,
      lv_severity   TYPE bapi_mtype.
FIELD-SYMBOLS: <item>   TYPE /scwm/dlv_item_out_prd_str.

* call central cleanup
/scwm/cl_tm=>cleanup(
        EXPORTING
          iv_lgnum  = lv_lgnum).

* transfer DLV keys
LOOP AT lt_items ASSIGNING <item>.
  CLEAR ls_create_whr.
  ls_create_whr-rdocid  = <item>-docid.
  ls_create_whr-ritmid  = <item>-itemid.
  ls_create_whr-rdoccat = <item>-doccat.
  APPEND ls_create_whr TO lt_create_whr.
ENDLOOP.

* trigger WT creation
CALL FUNCTION '/SCWM/TO_CREATE_WHR'
  EXPORTING
    iv_lgnum       = lv_lgnum
    iv_bname       = sy-uname
    it_create_whr  = lt_create_whr
    iv_update_task = abap_false
    iv_commit_work = abap_true
  IMPORTING
    et_ltap_vb     = lt_ltap_vb
    et_bapiret     = lt_bapiret
    ev_severity    = lv_severity.
```

Listing 2.9 Sample Call /SCWM/TO_CREATE_WHR

The return parameter lt_ltap_vb provides the WTs created. Tables lt_ bapiret and lv_severity contain the creation log, as well as the aggregated message type (E, I, etc.). If you want to implement in a project your own UI transaction for this application, for example, because the end user needs to see the created WTs before saving, use the function module /SCWM/TO_PREP_WHR_UI_INT instead of /SCWM/TO_CREATE_WHR. The program logic is basically identical. The difference is

that in the case of /SCWM/TO_PREP_WHR_UI_INT the WTs will initially be created only within in the SAP memory (internally) and then, depending on the action of the user, either be published and saved or deleted via rollback In case you are using the function module /SCWM/TO_PREP_WHR_UI_INT, you are completely responsible for the entire update (commit) control. To guide you to the appropriate logic, just take a look at function module /SCWM/TO_CREATE_WHR. Here the function module /SCWM/TO_PREP_WHR_INT takes care of both the internal WT creation and, afterwards, depending on the reported result, the update logic.

Transfer Parameters of the Function Module /SCWM/TO_CREATE_WHR [«]

The function module /SCWM/TO_CREATE_WHR has various parameters that are all well documented in the function module itself. In the coding example (Listing 2.9), we only used the most important parameters.

The following additional components of SAP EWM also use delivery objects:

- Packing
- Shipping and Receiving (S&R)
- SAP EWM Master Data
- Value-Added Services (VAS)
- Kitting

The corresponding service classes are grouped under /SCWM/CL_DLV* in package /SCWM/DELIVERY. Here in this package, the core functions of the delivery processing are developed.

In the other chapters in this book, we will provide you with some implementation examples in relation with the delivery processing. For example, in Section 3.6.6, we will describe how the delivery processing integrates with the special packaging features.

Delivery Assignment in Inventory Management [«]

A major advantage of SAP EWM is the quant separation, which is also due to the delivery assignment. This allows you, for example, to move already picked goods back and forth easily within SAP EWM without losing the reference to the delivery. The goods will remain clearly identifiable throughout the system.

In addition, the delivery processing includes interfaces for the integration of the following components:

- Quality Inspection Engine (QIE)
- Post Processing Framework (PPF)
- Route Determination
- Global Trade Services (SAP GTS)
- Transportation Management (SAP TM)

In addition to the various interfaces and calling options of delivery objects, SAP EWM also offers a number of BAdIs which allow you to carry out your own validations or, within the delivery processing, to fill customer-specific fields. In Chapter 6, we give a complete overview of all BAdIs in SAP EWM. At various BAdIs, SAP delivers sample implementations to give you a programming template for orientation.

For further information, visit the SAP Service Marketplace (*http://service.sap.com*) where you will find many SAP Notes with practical tips for troubleshooting and coding examples (for example, 1064376, 1436107, 1462811, 1451135).

2.2 Warehouse Logistics

The warehouse logistics includes all movements of goods—both within the warehouse complex as well as around the warehouse. SAP EWM supports numerous processes such as the arrival of goods in the warehouse, goods receiving, and the putaway process. It also supports internal storage processes, such as inventory, quality control, supply, and reorganization.

The most important processes are picking and shipping. As the basis for almost all logistics processes, warehouse logistics thereby serves the delivery request. It is the pool and thus the request document for the warehouse on which planning and reporting take place, as the warehouse request in SAP EWM serves the *delivery* (see Section 2.1). On the delivery you can see the current processing status and move in all dependent documents.

For Shipping & Receiving (S&R), complete deliveries, respective items, or handling units (HU), are assigned to a transportation unit (TU). You can group TUs by vehicles. The central document in SAP EWM is the warehouse task (WT) in which

activities within the warehouse are performed and documented. It serves the movement of stock and/or HUs according to the data specified in the delivery items. But it's also possible to create and process them without a delivery reference for internal warehouse processes.

In addition to the WT, more processes can be carried out. For example, inbound quality checks can be carried out (see Section 2.4). VAS can be integrated in the inbound and outbound process. In the outbound process, the delivery items are grouped in waves. WTs are created for a complete wave and bundled to warehouse orders according to certain customizable criteria.

2.2.1 Shipping and Receiving

An incoming transport, such as a truck, often contains more than one delivery. To illustrate this fact in the system, we can create in SAP EWM a separate object—the TU—and assign deliveries, delivery items, or HUs to the TU activity.

When you create a TU, you will always produce an activity to the TU—the TU itself is not editable. An activity (for TU, vehicle, or gate) is always valid for a specific time frame, which is defined in the activity. To a TU (it's defined by the external number + carrier), there may be several *planned* activities but only one *active* activity; an activity is activated by the arrival at the checkpoint. Activities of one or more TUs can be assigned to a vehicle.

The use of the object "vehicle" will only bring an added value when more than one TU is used, for example, a truck with a trailer. If a delivery is distributed to multiple TUs, this can be represented in the system by direct assignments of the delivery items or HUs to these TUs. This takes place primarily in the goods issue process during loading when the final deliveries are not yet known and processing takes place on warehouse requests.

Yard Management in SAP EWM **[«]**

The "yard" is the name we call the area outside of the warehouse complex, where the arriving and departing TUs are handled. With yard management in SAP EWM, you can not only register the incoming TUs at the gate but also assign locations and move them within the yard. You can occupy parking spaces and create WTs for moving the TU to its target (mainly a door). This provides an opportunity for parking management and administration of additional TUs, for example, swap bodies or trailers that must be moved with their own resources.

The doors of the warehouse, where TUs are loaded and unloaded, must be assigned to a storage bin in the warehouse. These door bins are necessary to create WTs to the door. But it does not involve physically existing places where goods are actually stored. Working with the yard management, the doors must also be assigned to a storage place outside of the warehouse—the place where the TU is off target. The two bins assigned to a door (outside and inside) connect the yard to the warehouse complex.

The movement of a TU with a WT to a door automatically creates or activates a door activity for the TU. Only one TU can be docked at a door. For this reason, we recommend the use of yard management for timely postings only. Without yard management, you must create and activate door activities manually in TU transaction. S&R works on activating and deactivating with time stamps (current date and time).

The use of TUs also provides the ability to post the stock to the TU location. You can therefore post goods receipts sooner without taking up storage space. Outbound process staging areas will not only be cleared from the goods issue but also by loading the TU, thus you can manage staging areas better. Of course, loading is possible without TUs; in this case, stock is moved to the door bin. The stock transparency is thus no longer ensured, since physically the door bin does not exist and stock cannot be checked. Accordingly, it is not clear where in fact the goods are located.

The activity of a TU contains information about when the TU is expected in and at what time it will leave the warehouse again. The activity may provide additional information, such as the driver and means of transport (e.g., license plate), including other identifications and seal information. But above all, it includes status information.

The activity of a TU is activated by the user action Arrival at Checkpoint. On activated yard management, the HU of the TU is moved to the checkpoint bin. From that point in time, the TU can be moved with WTs within the yard to a door where it is unloaded or loaded. If no free door exists, it can be temporarily moved to a parking lot.

Receiving

In inbound processing, customizing decides if the goods receipt (GR) is posted to the TU location or door bin or to the staging area and thus if goods must be

unloaded or not. Customizing for GR posting can be found in the IMG on SCM Extended Warehouse Management • Cross-Process Settings • Shipping and Receiving • General Settings • Control of Goods Movements. If GR is posted to a TU or door bin, unloading is required with WT. From goods receipt to staging area, no unloading is required, but it is supported with simple unloading. After unloading the TU, it can leave the door and depart from the checkpoint or it can be used for loading a shipping activity.

Status

Object TU and vehicle contain a number of statuses to document the current state of the activity. For example: *arrival at checkpoint*, *docked to door*, *begin loading*, and *loading end*. Existing both on the object "delivery" and on the object TU, status is synchronized between the two objects. The TU informs the delivery upon arrival at checkpoint about the status change, which sets the status in yard in the delivery. If the TU is already planned for a door, the door information is passed to the delivery. If no open WT exists on the delivery item, it is written to the delivery. Staging area determination is called to find an optimized staging area for the door. When changing the TU, the synchronization between the TU and the delivery takes place directly. For this reason, the deliveries will be blocked and must be changeable. Otherwise, the change is not possible for the TU.

Status changes on the delivery, as with changes in loading or goods movement status, lead to a recalculation of the corresponding status in the TU activity. Changes in quantity and repacking eventually lead to a change in capacitance of the TU activity. Changes in delivery, such as status or quantity changes that affect the transport, are not updated directly in the TU, but they result in scheduling a PPF action that synchronizes the TU (see Figure 2.5).

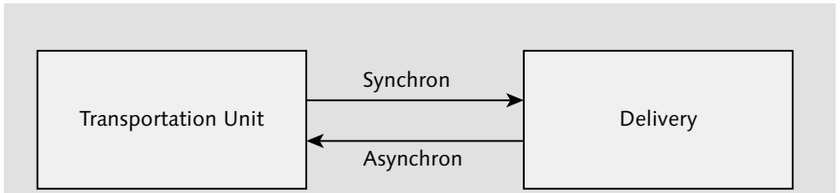


Figure 2.5 Synchronization between Delivery and TUs

[»] **Transport Units Contain Only References**

TUs do not contain their own quantity, product, or HU information; only references to the delivery number, respective item, and possibly to HUs.

For this reason, the UI of the TU always shows the current state of the HU data. These data do not necessarily correspond with the data of the received HUs after unloading.

Shipping

For shipping, a TU activity can arrive at a checkpoint that only picks up goods or a TU can be reused that previously delivered incoming goods and is still active in the yard or at the door. For such a TU, the shipping activity can be activated and the system automatically closes the incoming activity. If the incoming activity is located at a door, create and activate a door activity for the outgoing activity also. With active yard management, the location of the TU does not change.

Since the transported objects need not be planned, they can be assigned by loading spontaneously; as opposed to inbound, it is not automatically determined by the system when the loading is complete. Status Loading Completed must always be set manually; if loading started, completion is a prerequisite before the activity can leave the checkpoint again.

WTs are necessary for the unplanned loading of TUs, which provide the assignment of HUs to TU.

Data Model

S&R is part of the namespace /SCWM/. The responsibility is summarized in the component SCM EWM-SR and its subcomponents. Beside the object's TU, vehicle, and door, this component also handles staging and door determination, as well as loading and unloading functionality. Both functions are relevant for delivery processing, regardless of the use of TU.

The package `/SCWM/SHP_RCV` is divided into five subpackages. In addition, package `/SCWM/YARD_MGMT` is valid for yard management functionality. Table 2.4 lists the packages that are relevant for S&R.

The processing of transactional data is done via the classes shown in Figure 2.6.

Package	Function
`/SCWM/SHP_RCV_CORE`	Processing of TU, vehicle, door, and staging area determination
`/SCWM/SHP_RCV_CUST`	Customizing tables, maintenance views, functions for reading customizing and F4-help
`/SCWM/SHP_RCV_PPF`	Implementing PPF
`/SCWM/SHP_RCV_UI`	Transactions for processing of TU, vehicle, and door activities
`/SCWM/ERP_TM_INTEGRATION`	Customizing and IDocs for integration of SAP ERP shipment (LE-TRA)
`/SCWM/YARD_MGMT`	Definition of checkpoints, transaction for check-in and checkout, yard moves

Table 2.4 Packages for S&R

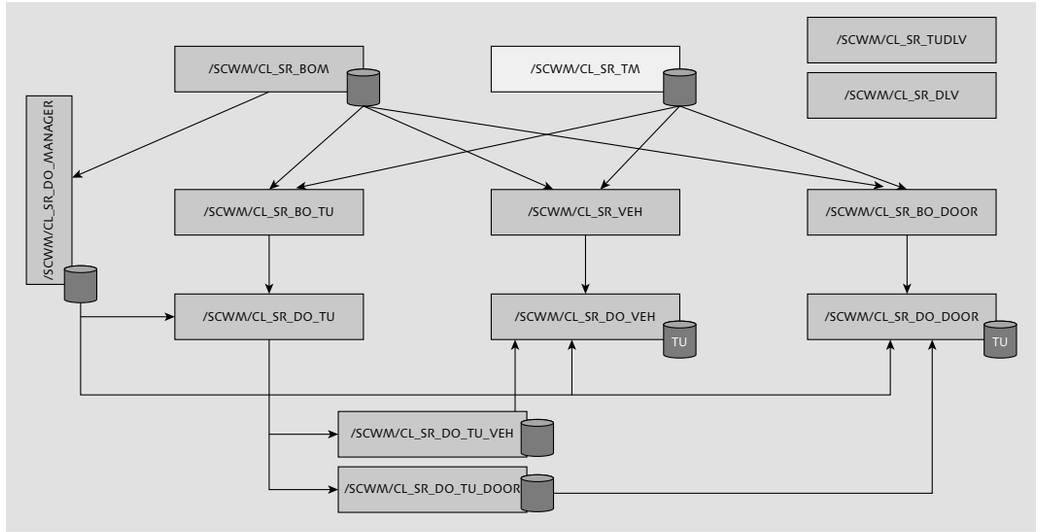


Figure 2.6 Class Model of S&R

Each activity will be managed by a separate instance of that particular class. The data can be found in the classes `/SCWM/CL_SR_DO..` The business logic is executed through the associated classes `/SCWM/CL_SR_BO..`, which hold a reference to the current instance of data. The property manager `/SCWM/CL_SR_BOM` manages the instances of business objects `/SCWM/CL_SR_BO..`, etc.

The links between TU and vehicle or gate are managed in their own classes, which are referenced in the class /SCWM/CL_SR_DO_TU. For performance reasons, keep the corresponding instances of door and vehicle keys with a table of the associated TUs.

The link between TUs and delivery is managed in class /SCWM/CL_SR_TUDLV. It is independent of the other classes and may also be used without them if no other information about the TU or vehicle is required. Class /SCWM/CL_SR_DLV is the interface of the TU for the delivery. It informs the delivery about changes in its assignment and status.

The class /SCWM/CL_SR_DO_MANAGER exists only for historical reasons and must not be used directly. Only the methods for reading/setup of all relevant objects will remain and must only be used by the business object manager class /SCWM/CL_SR_BOM.

The three objects in S&R consist of the object header and any number of activities in this header. For an entry in table /SCWM/TUNIT or /SCWM/VEHICLE at least one entry in its activity table /SCWM/TU_SR_ACT respective to /SCWM/VEH_SR_ACT must exist. All additional tables contain entries relevant to the activity. The tables of the objects are listed in Table 2.5.

Object	Table	Activity	Additional tab
Transport Unit	/SCWM/TUNIT	/SCWM/TU_SR_ACT	/SCWM/TU_STATUS /SCWM/TU_IDENT /SCWM/TUNIT_SEAL /SCWM/TU_VEH /SCWM/TU_DOOR /SCWM/TU_DLV
Vehicle	/SCWM/VEHICLE	/SCWM/VEH_SR_ACT	/SCWM/VEH_STATUS /SCWM/VEH_IDENT
Door	/SCWM/TDOOR	/SCWM/DOOR_SR_ACT	

Table 2.5 Tables of Objects in S&R

TU and vehicle header are deleted if the last activity is deleted.

The object "door" has a special status because it is defined in customizing. Entry must exist when creating an activity, and door activities cannot be created as

standalone. They only exist with a reference to a TU activity. This means for any entry in /SCWM/DOOR_SRACT an entry in /SCWM/TU_DOOR must exist.

For just about any action that can be performed for an activity, the BAdI /SCWM/EX_SR_ACTION_TU for TUs, /SCWM/EX_SR_ACTION_VEH for vehicles, and /SCWM/EX_SR_ACTION_DOOR for doors is called. Those BAdIs can be used for custom checks and trap implementation.

Since the three objects are independent and some changes have an impact on more than one object, the consistency of the objects must be ensured when the data changes. For this reason, if any method of /SCWM/CL_SR_BO_.. class (briefly: BO method) is performed, a copy of instance /SCWM/CL_SR_DO.. (briefly: DO) is saved before the first data change. In addition, they inform the class /SCWM/CL_SR_TM that it has produced such a copy.

If an action is successful, the application that called the action—such as "upon arrival at the checkpoint"—calls class /SCWM/CL_SR_TM to initialize and inform all stored BO instances to delete the copy.

If within an action in one of the objects an error is raised, class /SCWM/CL_SR_TM is also called. It will inform all stored BO instances to discard the current DO reference and replace it with the copied one. Furthermore, it is ensured that if an error occurs, data will be the same on all objects like before the action started.

2.2.2 Warehouse Task and Warehouse Order

Goods movements within the warehouse are mapped on the object WT. For goods movements, such as goods receipt, goods issues, and posting changes, the system generates completed WTs without a warehouse order for documentation.

Rearrangements within the warehouse consist of scheduled and completed WTs. When creating a scheduled task, a warehouse order is automatically generated in accordance with the warehouse order creation rules defined in customizing. The warehouse order can be assigned to a resource group and is used to edit and confirm assigned WTs.

The purpose of a WT can be identified by its warehouse process category (TRART). The possible modes of transport are listed in Table 2.3. The data available on the WT depends on this process category. See Table 2.9.

Warehouse process category	Description
1	Putaway
2	Picking
3	Internal movement
4	Inventory
5	Goods receipt
6	Goods issue
7	Posting change

Table 2.6 Warehouse Process Categories in SAP EWM

Storage and retrieval can be done in several steps. This can be due to the process (in this case, the process will be guided by process-oriented storage control) or warehouse layout (which is defined by layout-oriented storage control). The various storage process steps of the process-oriented storage control are usually optional and some are feasible at any point in the process. Most can be defined as rule-based, which means they can be included in the process profile, but they are carried out only on the existence of a dependent document.

These steps are divided in two stages and must be completed manually after the transfer, while single-stage process steps are completed automatically with the confirmation of the WT. At each process step, the system determines whether the completion of the process step leads automatically to create the WT of the next step or whether this takes place at a later date. To use the process-oriented storage control it is necessary to work with HUs, which serve as carriers of the storage control information.

In the following section, data determination for storage and retrieval is discussed in more detail.

Warehouse Process Category 1 (Putaway)

The putaway storage process consists of HU and product WTs. The final putaway can be done with both types of WTs. So it's also possible to putaway mixed HUs. Product WTs always contain an inbound delivery reference, whereas HU WTs only refer to a document if it is unique.

If WTs are created prior to GR, the source location is determined from first delivery item within the tasks (HU). After GR, the source location is determined by the

location of HU or stock that should be moved. Destination data comes from storage process or putaway strategy (for final putaway task).

In the goods receipt process, the last step is always the final storage. However, the WTs for this process step can be created at any time during storage process. This is done on the identification PRODUCT / HU WT in the allocation form of the storage process to a storage process. In this case, the WTs are created with the status WAITING and assigned to a warehouse order. In the next section, a multistage goods receipt process is described, and the roles of the objects "WT" and "HU" will be explained.

1. **Move transport unit to door**
   Upon arrival of a transport unit, these can be moved with active yard management using WTs in the yard. This WTs move a HU named by the TU. As a source location, the actual location of the TU HU is determined. Either obtain the destination from the storage process type or enter it manually. Confirmation of this WT will dock the TU to a door. If no yard management is used, then process the door docking manually.
2. **Unloading**
   If the goods receipt of a delivery item is received in the staging area, unloading is not mandatory, but it can be done by so-called simple unloading. In the delivery, a document flow is generated by the unloaded amount and UNLOADED status set for the HUs.

   If the goods receipt of the delivery item is posted to the TU or a door bin, HUs or unpacked stock must be unloaded with WTs. Unloading is an integral part of the goods receipt process. Concerning unloading, there are basically the following two options:

   - **Unloading without storage process**
     For unpacked goods, the system creates a WT with the storage process type of the delivery item. In this case, the WT represents unloading and putaway. This is also possible for HUs when creating the putaway WT. Confirmation of the unloading will also confirm the putaway step if no manual interaction with change of destination is performed. From the unloading UI, it's also possible to unload the HU to the staging bin of the delivery item.
   - **Unloading with storage process**
     The first warehouse process step of the storage control in the HU must be the unloading step. If WT creation is called for, this HU, the system automatically

creates a HU WT from the delivery item GR location to the staging area of the inbound delivery.

For unloading, WTs using a warehouse order creation rule of the type LOAD/UNLOAD makes sense. You can create it by choosing the IMG path EXTENDED WAREHOUSE MANAGEMENT • CROSS PROCESS SETTINGS • WAREHOUSE ORDER • DEFINE WAREHOUSE ORDER CREATION RULE. Use the creation type LOAD/UNLOAD for the corresponding warehouse order creation rule (Figure 2.7). If the unloading WTs are created automatically by the system from PPF or manually for a complete delivery or TU, all WTs are created in parallel. In this case, all WTs are assigned to one warehouse order. Depending on the number of workers configured, additional warehouse orders without WTs are created to allow parallel unloading in Radio Frequency (RF). Confirmation of an unloading WT will reassign the WT from the main warehouse order to the warehouse order of the worker. This means that multiple users can process the same worklist without a predefined unloading sequence. In the end, the identity of the user who unloaded the single HU is documented.

Warehouse Order Creation Rules

War...	WOCR	Description	Creat.Cat.
0001	HE01	Heavy Items	Pick Path
0001	HU01	Complete Pallet Withdrawal	Pick Path
0001	KO01	Minimum 5 Items per Consolidation Group	Consolidation Group
0001	LI01	Light Items to Max. 25 kg Total Weight	Pick Path
0001	OFTC	Flow-Through Recipient-Driven	Consolidation Group
0001	OFTP	Flow-Through Product-Driven	Distributive Putaway
0001	OMDX	Merchandise Distribution Cross-Docking	Pick Path
0001	PU01	Putaway	Pick Path
0001	UL01	Unload	Load/Unload

Figure 2.7 Warehouse Order Creation Category within Warehouse Order Creation Rule

3. **Counting**
   Counting may be included as a rule-based step in the storage controller. If an HU item is relevant for counting by an existing counting document for the packed delivery item, an HU WT is created to a counting station. There the content of the HU is counted and the step is then completed. If the counting step is not identified as rule-based, the HU is brought to the counting station regardless of the existence of a counting document.

For delivery items with a counting and quality step or a VAS, the counting step must be included in the storage control prior to those steps. For storage processes that contain only deconsolidation and putaway after unloading, counting may be completed implicitly by those WTs.

4. **Quality inspection**
   The quality control may be included as another rule-based step in the storage process. If a quality inspection document exists for the content of the HU, a WT is created for the inspection bin determined by the inspection document (inspection rule). However, the HU is only moved to the quality work center of the first relevant quality inspection. If the HU contains several test-relevant items with different quality work centers, the routing between them must be done manually. If no work center can be determined, no HU WT is created and the step must be completed at the current bin.

   After completion of the quality inspection, the HU must be marked COMPLETED. The system does not check whether there is still noninspected stock in the HU.

5. **Value-Added Service**
   The storage process step for VAS should be defined as rule-based also. This step is relevant if at least one item of the HU contains a VAS order. A HU WT is created to a work center for VAS. After completion of the VAS, the user must mark the HU as COMPLETED.

   Although the VAS step can be implemented at any point in the storage process, processing a product after a possible quality inspection makes more sense.

6. **Deconsolidation**
   Deconsolidation is an optional step in storage control that may be marked as rule-based. To use deconsolidation step it's mandatory to create putaway WT in an earlier defined process step. Those putaway WT are then created with status WAITING.

   An HU is relevant for deconsolidation if there is more than one WT for the content of the HU with a different consolidation group. Content of the HU has to be repacked so that the resulting HU can be used for putaway. The deconsolidation step must also be COMPLETED manually.

7. **Putaway**
   Putaway is always the last step in the goods receipt process. You can create the WT for this step at any time in the process. In the putaway step, you create a putaway WT or activate an already existing putaway WT with the status WAITING.

Warehouse Process Category 2 (Picking)

The outbound process always starts with a product WT with reference to the outbound delivery order, which is called picking. For this product WT, source stock determination is done by warehouse process type (WPT) from the delivery item and picking strategy. Destination data comes from outbound delivery order.

Upon creation of picking WTs, they are bundled to warehouse orders with warehouse order creation rules. In picking, the warehouse order creation provides the most opportunities for optimization and processing of different scenarios (see Table 2.7). For this reason, it makes sense to create as many WTs as possible together. For this, SAP EWM offers the wave functionality: with waves, delivery order items that are to be picked in a certain period are pooled, and with the wave release, the WTs for all items within this pool are created.

Creation Category	Description
Consolidation group	If possible, all WTs for one consolidation group are picked together. So all products going to the same customer are held (packed) together at an early stage. The picking path is a bit longer, but you can possibly skip an additional repacking.
Pick path	WTs are sorted by pick path and then bundled so that the shortest paths are covered, and the pickers are working in different areas. But products have to be brought to a packing station to consolidate deliveries.
Pick pack pass – system	The WTs are bundled according to the consolidation group within an activity area. The order of the warehouse orders of a consolidation group is defined in customizing, and only one warehouse order is active. Others are waiting. Confirming a warehouse order activates the next warehouse order of the same consolidation group in the next activity area. For this scenario, several activity areas must be assigned to the activity area for warehouse order creation.
Pick pack pass – user	In this scenario, the WTs are also bundled on consolidation group per activity area, but all warehouse orders are active.

Table 2.7 Warehouse Order Creation Categories for Picking

Process-Oriented Storage Control in Outbound Processing

In this section, a multistage outbound process is described, and the roles of objects WT and HU are explained. The five stages, in order, are:

1. **Picking**
   WT creation for picking is completed like the outbound delivery order item and picking strategies described earlier.
2. **Value-added services**
   The storage process step for VAS should be maintained as rule-based within the storage process. In this case, the step is only relevant if a VAS document exists for the outbound delivery order item. If VAS is the second step within the storage process, picking WTs are already created to the VAS work center. On confirmation of picking WTs, picking HUs must be used. If VAS is not the second step within the storage process, an additional HU WT is created to the VAS work center.
3. **Packing**
   If WTs are bundled by pick path, different pickers pick stock for one delivery. For shipping, those stock must be consolidated and packed at a packing work center. In this case, picking WTs are created directly to the packing work center. With the VAS step, the HUs are moved from the VAS work center to the pack station with HU WT if needed. The packing step in the storage process must be completed manually.
4. **Staging**
   In the staging step, the HUs are moved to the staging area.
5. **Loading**
   Loading creates HU WTs from the staging area to the door of the outbound delivery order item. Confirmation of any WT to a door bin is defined as loading WT and creates a document flow entry in the delivery order items and a status on the HU in case of a HU move. If a TU is docked to the door when confirming the WT in RF, destination of the WT is changed directly to the TU.

Data Model

The processing of WTs can be found in package `/SCWM/CORE`. Warehouse order creation is assigned to package `/SCWM/WHO`.

Planned WTs are stored in table /SCWM/ORDIM_O with an identical entry in /SCWM/ORDIM_L. On confirming or canceling this planned WT, the entry in /SCWM/ORDIM_O is deleted and a new entry is created in table /SCWM/ORDIM_C. This ensures that the table of open WTs is kept small and accesses accordingly perform well. WTs always references a warehouse order in table /SCWM/WHO.

Table 2.8 lists the tables in which the data for WT processing is stored.

Table	Usage
/SCWM/ORDIM_O	Planned WTs with status open or waiting
/SCWM/ORDIM_OS	Serial number for open WTs
/SCWM/ORDIM_L	Log-table for planned WTs. It's a copy of the original entry in /SCWM/ORDIM_O which documents the initial values. It is not updated at any point in time
/SCWM/ORDIM_LS	Serial numbers for log-table
/SCWM/ORDIM_C	Completed WTs
/SCWM/ORDIM_CS	Serial numbers for completed WTs
/SCWM/ORDIM_H	Stock information for confirmed HU WTs with mixed stock
/SCWM/ORDIM_HS	Serial numbers on mixed HU WT referencing /SCMW/ORDIM_H
/SCWM/ORDIM_E	Exception codes for completed WTs

Table 2.8 Tables of WTs

The amount to be moved with a WT can be confirmed partially. This is called splitting the WT. It creates a confirmed WT entry with the confirmed partial quantity. This is, for example, necessary if the amount is divided between more than one destination (pick) HU or if the source HU is a nested HU and picking is done from part of a sub-HU or from some complete sub-HUs. The table key of /SCWM/ORDIM_O and /SCWM/ORDIM_L is the warehouse number and WT. Tables of confirmed WTs contain an additional item number.

The attributes of the WT can be divided in the following groups:

- Organizational data: user name, creation date, resource, queue, confirmation date, etc.
- Stock attributes: product, quantity, batch, stock category, etc.

- Source location: description of the location in the warehouse where stock is currently placed. Source storage bin, source HU, sources resource, or source TU.
- Destination location: description of the location in the warehouse where the stock must be put. Destination storage bin, destination HU, destination resource, or destination TU.
- Additional data: more general attributes, such as FLGHUTO, which indicates the move of the complete source HU, or field HOMVE, which signifies for the product WTs that the complete stock of source HU should be moved and if the move of complete HU is possible.

Organizational data attributes are always filled. Stock attributes are only filled for product moves, stock changes, and HU moves with unique stock. Source and destination data depend on the type of WT. What data is filled on which posting (transportation type and product or HU posting) can be found in Table 2.9. The transport types shown are defined in table /SCWM/T333A.

Type	TRART	Source	Destination	Stock	Additional
Product GR	5		X	X	
HU GR	5		X	Optional	FLGHUTO
Product GI	6	X		X	
HU GI	6	X		Optional	FLGHUTO
Posting change	7	X	 X	X X	
Product put-away, picking, internal move	1, 2, 3	X	X	X	HOMVE HUENT
HU put-away, picking, internal move	1, 2, 3	X	X	Optional	FLGHUTO MOVEHU
Inventory of complete HU	4	X	X	Optional	FLGHUTO
Product inventory	4	X	 X	X X	

Table 2.9 Provided Data within WT Depending on Type of Task

From and to data include the location and the HU. Besides the location, the affected GUID_HU is always filled. HU identification remains for the initial dummy HUs, which makes it easy to see whether this is a real or a virtual HU. Flag FLGHUTO indicates whether the task was created as an HU posting. If destination storage type does not allow real HUs, only the content of the HU is moved, which can be identified on flag MOVEHU. If a product WT for a complete HU is created, it is identified by flag HOMVE. In this case, the complete HU can be moved like a HU WT and flag HUENT is set to indicate the HU move.

WTs can have different statuses (see Table 2.10).

Status	Usage
Open	WT waiting for processing
Waiting	WT that reserves source stock and capacity on destination but cannot be processed
Confirmed	WT was processed successfully
Canceled	Planning was canceled

Table 2.10 Status of a WT

[»] **Planned Warehouse Tasks are Never Deleted**

Planned WTs can only be canceled but not deleted. Confirmed WTs and stock postings cannot be canceled. They must be posted in the opposite direction with a new WT.

Source-Data-Determination

For putaway WTs for inbound delivery items without GR, the source bin is determined from the inbound delivery item's GR location. After GR, the source location is taken from stock with reference to this inbound delivery item. For manual product WTs (Transaction /SCWM/ADPROD) or HU WTs (Transaction /SCWM/ADHU), stock must be selected by the user to be able to create a WT.

For pick WTs, source data is determined by available quantities (table /SCWM/AQUA) with picking strategy and packaging specification for the product. An exception to this rule is the WT creation for a cross-dock scenario. In this case, the outbound delivery order already contains information about the inbound delivery and WT searches for inbound delivery stock reference (where putaway has not been completed). Source location determination is done in function group /SCWM/REM_BIN_DET.

Figure 2.8 shows the customizing of the picking strategy.

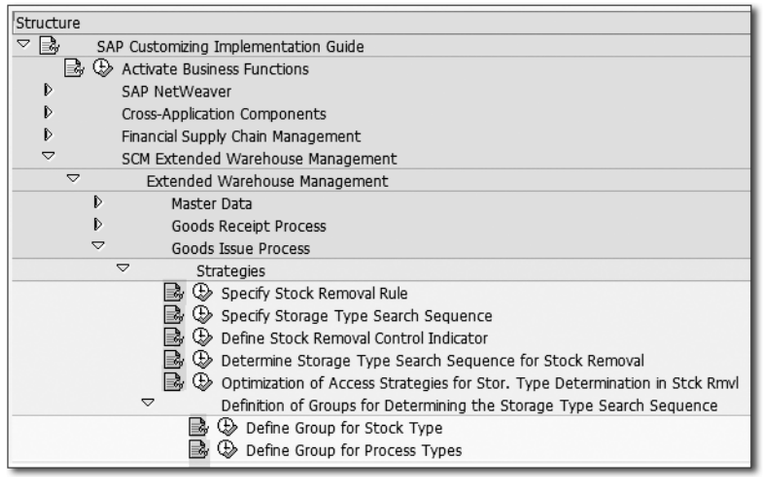


Figure 2.8 Customizing for Picking Strategies

The picking strategy defined in customizing is used to determine the source storage bin respective source HU. If a quantity classification is defined in source storage type, the system calls packaging specification determination with type 0WHT and determines the relevant level for operations with the quantity classification of the storage type. Upon successful level determination, the operational unit of measure is set by the packspec level. An optional rounding of the WT quantity can be done (e.g., to ensure that only whole boxes are picked), or the quantity can be reduced to one unit of the operating unit (e.g., in a bulk storage where one WT must be created per pallet).

With the BAdIs listed in Figure 2.9, picking strategy and thus affected source location determination can be customized specific to the process. For example, the operational *unit of measure* (UOM) can be set if stock is stored in a different UOM than in the standard storage type. An example of implementation can be found in Section 4.

To allow parallel WT creation, the source storage bin is locked with a shared enqueue. This allows any number of parallel processes to access this data. An exclusive lock is used for the quantity of the created WT to ensure availability. For product WTs with a given source HU, the HU also receives a shared lock. This is also true for repacking at a work center. When moving a complete HU or

repacking a complete HU, the HU is gets an exclusive enqueue. Therefore repacking stock must be saved prior to repacking or moving the HU.

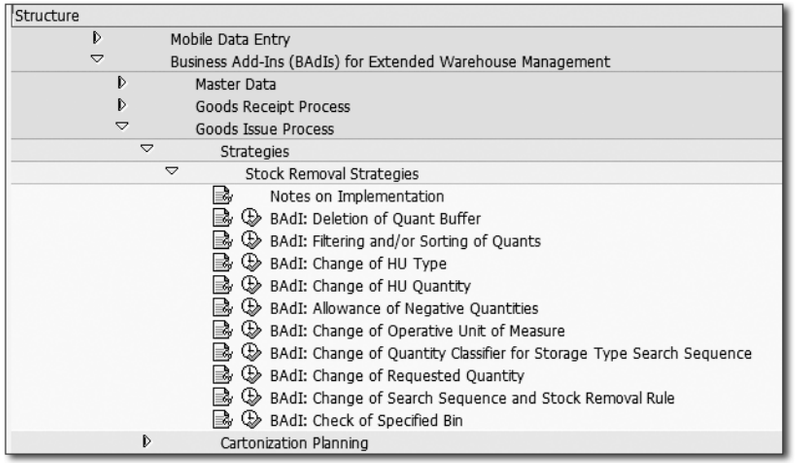


Figure 2.9 Business Add-Ins to Influence Picking Strategy and Source Location Determination

Destination Data Determination

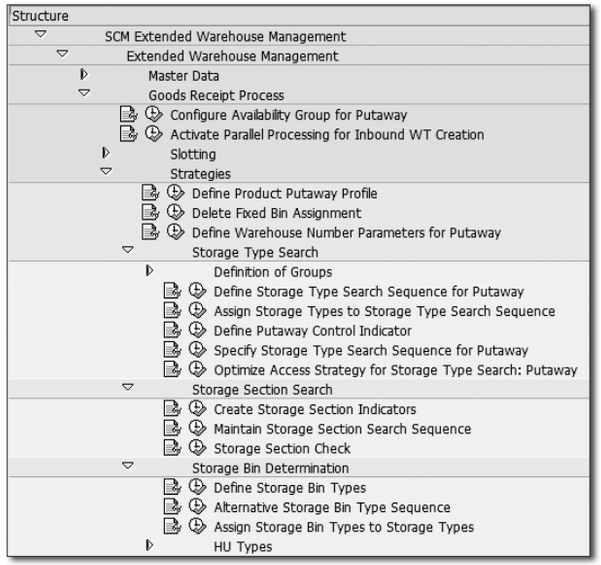


Figure 2.10 Customizing of Putaway Stategy

Destination data for final putaway is determined by the putaway strategy defined by the product. The definition of putaway strategy is completed in the customizing of the goods receipt process (see Figure 2.10). For other processes, destination data is taken from the delivery item or storage process step (e.g., work center), or is defined manually.

Examination of destination data is done in function group `/SCWM/PUT_BIN_DET`. Here, according to the settings on the storage type (see Figure 2.11), the mixed storage and capacity check is called. On moving a complete HU (HU WT or product WT for complete HU quantity), the capacity of the HU header is used for capacity check. Mix stock and capacity check are done in function group `/SCWM/HUFUNC`. Function module `/SCWM/TO_HU_INT` is called. The destination HU (as long as it's no dummy (see Section 2.3.2)) is locked exclusively by an enqueue. Destination bins are also locked if mix stock and capacity check must be done. Deactivation of capacity check and capacity update can be useful for work center and staging bins.

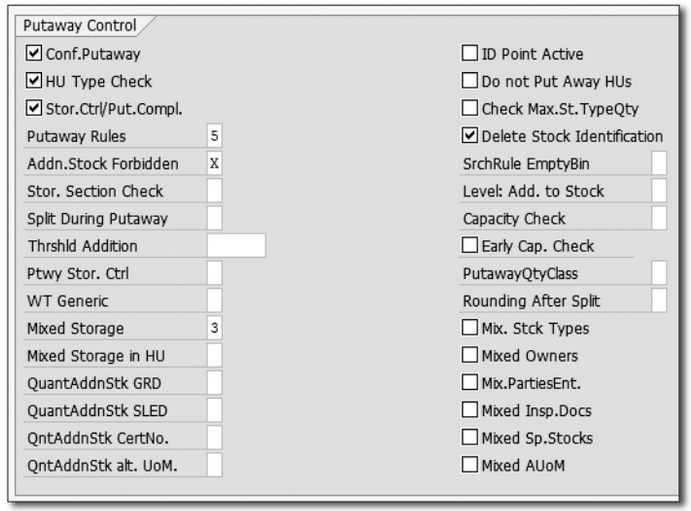


Figure 2.11 Setup for Mixed Storage and Capacity Check within Storage Type

Pure product WTs determine a packaging specification with condition type 0WHT. Capacity information is determined up to the relevant level in the packaging specification, which is defined by the quantity classification for putaway (general quantity classification if no specific is defined). Only HU relevant levels are

used. If you cannot find the packaging specification, use the capacity information if the product master of base UOM is used. Products with catch weight are the exception. In that case, the quantity of capacity relevant UOM is used.

An influence of all parameters of the putaway strategy is possible via BAdI implementation. An overview of the available BAdIs is given in Figure 2.12.

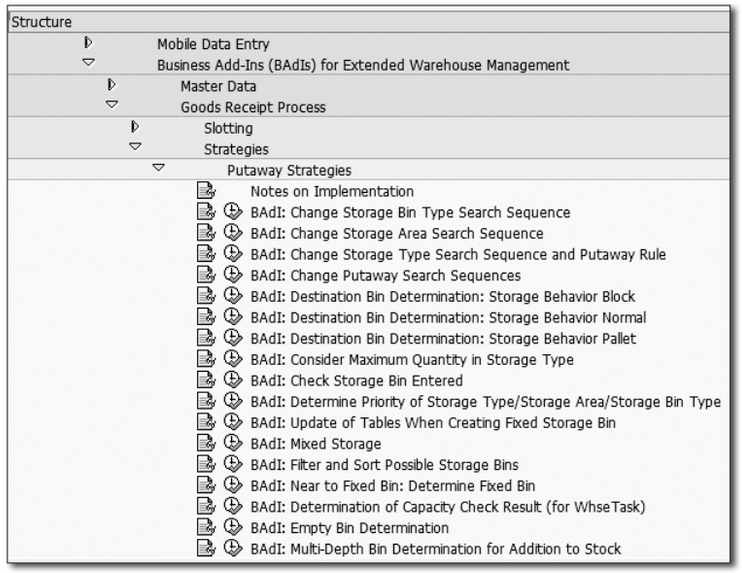


Figure 2.12 Business Add-Ins for Putaway Strategy

2.3 Inventory Management

Inventory data and hierarchies in SAP EWM are stored on the Lean Inventory Management Engine (LIME). The structure model of SAP EWM and LIME contains three object types:

- Location
- HU
- Stock

The objects of these types form a hierarchy, with stock always stored in HUs. HUs can contain other HUs and an HU is always anchored on a location. An example is shown in Figure 2.13.

Contents

Foreword 13
Introduction 15
Acknowledgments 19

1 Flexible Warehouse Management with Extended Warehouse Management 21

1.1 Warehouse Management with Standardized Software: Configuration and Enhancement 21
1.2 Flexibility of Extended Warehouse Management 22
1.2.1 Activity Areas: Flexible Assignment of Storage Bins to Work Areas for Warehouse Operators 23
1.2.2 Storage Control: Modeling Multistep Warehouse Processes 24
1.3 Summary 26

2 Architecture 29

2.1 Delivery Processing 30
2.1.1 Function of the Delivery Processing 31
2.1.2 Object Warehouse Request and Data Model 33
2.1.3 Integration with Other SAP EWM Components 47
2.2 Warehouse Logistics 50
2.2.1 Shipping and Receiving 51
2.2.2 Warehouse Task and Warehouse Order 57
2.3 Inventory Management 70
2.3.1 Locations 72
2.3.2 Handling Units 73
2.3.3 Stock 74
2.4 Quality Management 80
2.4.1 Function 80
2.4.2 Objects and Data Model 88
2.4.3 Integration 89
2.5 Integration with SAP ERP 91
2.5.1 SAP ERP Systems 92
2.5.2 Non-SAP ERP Systems 119
2.6 Summary 125

3 Frameworks and Development Tools in Extended Warehouse Management 127

3.1 Warehouse Management Monitor 127
3.1.1 Basics and Technical Structure 128
3.1.2 Enhancement Options 136
3.2 Easy Graphics Framework and Measurement Services 138
3.2.1 Foundations 139
3.2.2 Custom Development: Basic Measurement Service 141
3.2.3 Adjust Chart Template 148
3.2.4 Custom Development: Easy Graphics Framework Object Service Provider 149
3.3 Radio Frequency Framework 157
3.3.1 Basics 158
3.3.2 Radio Frequency Framework and the Web Dynpro ABAP Transaction 163
3.3.3 Create a New Display Profile 165
3.3.4 Create a New Radio Frequency Menu 169
3.3.5 Specification and Design of a New Radio Frequency Transaction 170
3.3.6 Realization of the New Radio Frequency Transaction in the System 173
3.3.7 Realization of a Verification Profile 187
3.3.8 Value Helps in Radio Frequency with Function Key F8 191
3.3.9 Realization of Lists 192
3.3.10 Exception Handling in Radio Frequency 198
3.3.11 Process Functions in Background Mode in Radio Frequency Transactions 204
3.3.12 Enhance Standard Radio Frequency Transactions and the Use of Radio Frequency Business Add-Ins 204
3.4 Post Processing Framework 206
3.4.1 What is the Post Processing Framework 206
3.4.2 Extended Warehouse Management and Post Processing Framework 208
3.4.3 Enhancement Options of the Post Processing Framework 215
3.5 Easy Enhancement Workbench 220
3.5.1 Introduction to Easy Enhancement Workbench 220
3.5.2 Extended Warehouse Management and Easy Enhancement Workbench 222

3.5.3 Extended Warehouse Management and Easy Enhancement Workbench Structure Enhancements 226
3.6 Work Center 230
3.6.1 Basics and Architecture 230
3.6.2 Enhancing the Entry Screen 237
3.6.3 Custom Development: Change Icons in the Tree Control 238
3.6.4 Custom Development: New Tab in Scanner Area 239
3.6.5 Custom Development: Close Handling Unit 244
3.6.6 Custom Development: Packing of Inbound Deliveries 247
3.7 Summary 251

4 Extending Processes in the Preconfigured Warehouse W001 253

4.1 Introduction to Preconfigured Warehouse for Extended Warehouse Management 254
4.1.1 Functional Scope of the Preconfigured Warehouse 255
4.1.2 Overview of Custom Developments within the Processes 260
4.1.3 Installation of Preconfigured Warehouse 261
4.1.4 Testing Preconfigured Warehouse with Solution Manager 262
4.2 Inbound Process with Repacking for Putaway (106) 265
4.2.1 Process Description 106 266
4.2.2 Custom Development 106A: Permit Activation of the Transportation Unit Only after Entering the License Plate and Pager 268
4.2.3 Custom Development 106B: Delay Inbound Delivery with Missing Batch 274
4.2.4 Custom Development 106C: Putaway Depending on Quarantine Period 282
4.2.5 Custom Development 106D: Enhancing the Warehouse Monitor 287
4.3 Inbound Process without Packing Information (107) 301
4.3.1 Process Description 107 302
4.3.2 Custom Development 107A: Automatic Handling Unit Creation without Packaging Specifications 303
4.3.3 Extension 107B: Simplify the Screen Flow for Radio Frequency Putaway 314

4.4 Outbound Process Using Pick Handling Units as Shipping Handling Units (108) 320
4.4.1 Process Description 108 321
4.4.2 Custom Development 108A: Enhancing the Delivery Interface by Custom Data 323
4.4.3 Custom Development 108B: Transfer of Custom Data from ODO to Warehouse Task 328
4.4.4 Custom Development 108C: Showing Custom Data in the Form View of the ODO Item 330
4.4.5 Custom Development 108D: Determination and Transfer of Handling Unit Type from Packaging Specification to Pick Warehouse Task 333
4.4.6 Custom Development 108E: Determination of the Operative Unit of Measure by Packaging Specification of Goods Receipt 338
4.4.7 Custom Development 108F: Prohibit Goods Issue for Incomplete Packing 342
4.5 Outbound Process Using Wave, Pick Handling Units, Packing, Staging, and Loading (109) 345
4.5.1 Process Description 109 345
4.5.2 Custom Development 109A: Take over Transportation Unit after Unloading 348
4.5.3 Custom Development 109B: Print Picking Labels on a Mobile Printer 362
4.6 Periodic Physical Inventory (110) 373
4.6.1 Process Description 110 373
4.6.2 Custom Development 110A: Enhancing the Goods Movement Interface by Additional Data 374
4.7 Summary 382

5 Functions Modules and Methods for Extended Warehouse Management 383

5.1 Transaction Manager for a Logical Unit of Work in Extended Warehouse Management 384
5.2 Service Class for Filling Stock Fields 386
5.3 Date and Time for Time Zone of Warehouse Number 387
5.4 Cross Application Constants 388
5.5 Create and Extend Application Log 389
5.6 Read Warehouse Requests 391

5.7 Change Warehouse Requests 393
5.8 Read Warehouse Product Master 395
5.9 Create/Change Warehouse Product Master Records 396
5.10 Service Class Batch Management 398
5.11 Read Warehouse Task 400
5.12 Create/Confirm and Cancel Warehouse Task 400
5.13 Select Handling Units from DB 402
5.14 Create and Change Handling Units 404
5.15 Get Stock 405
5.16 Posting Change of Stock 407
5.17 Select, Release, Split, and Merge Waves 410
5.18 Get Transportation Units 414
5.19 Change TU 416
5.20 Read and Determine Packaging Specifications 417
5.21 Create/Change/Copy/Delete Packaging Specifications 418
5.22 Service Class for Radio Frequency Framework 418
5.23 Summary 419

6 Useful Business Add-Ins within Extended Warehouse Management 421

6.1 Warehouse Request 423
6.1.1 Enhancement Spot /SCWM/ES_ERP_MAPIN 424
6.1.2 Enhancement Spot /SCWM/ES_ERP_ERROR_HANDLING 425
6.1.3 Enhancement Spot /SCWM/ES_ERP_INT_CONF 425
6.1.4 Enhancement Spot /SCWM/ES_ERP_PROD 426
6.1.5 Enhancement Spot /SCWM/ES_ERP_PRIOP 427
6.1.6 Enhancement Spot /SCDL/TS_EXT 427
6.1.7 Enhancement Spot /SCWM/ES_DLV_DET 429
6.1.8 Enhancement Spot /SCWM/ES_CORE_CONS 430
6.1.9 Enhancement Spot /SCWM/ES_DLV_EGR2PDI 430
6.1.10 Enhancement Spot /SCWM/ES_ERP_MAPOUT 431
6.2 Wave 434
6.2.1 Composite Enhancement Spot /SCWM/ES_WAVE 434
6.3 Warehouse Task 436
6.3.1 Enhancement Spot /SCWM/ES_CORE_CR – Create Warehouse Task 437
6.3.2 Enhancement Spot /SCWM/ES_CORE_RMS – Stock Removal Strategy 441

6.3.3 Enhancement Spot /SCWM/ES_CORE_PTS – Putaway Strategies 447
6.3.4 Enhancement Spot /SCWM/ES_RSRC_QU – RF: Resource Management Queue Determination 456
6.3.5 Enhancement Spot /SCWM/ES_CORE_WT_RT – Extract Time Determination in Warehouse Document 456
6.3.6 Enhancement Spot /SCWM/ES_CORE_CO – Confirmation of Warehouse Task 457
6.4 Warehouse Order 460
6.4.1 Enhancement Spot /SCWM/ES_WHO 462
6.5 Exception Handling 469
6.5.1 Enhancement Spot /SCWM/ES_EXCP_EXC – Exception Handling 469
6.6 Summary 472

Appendices 473

A Programming Guidelines in Extended Warehouse Management for Enhancements 473
A.1 General Guidelines 473
A.2 Naming Conventions for Data Dictionary 475
A.3 Naming Conventions for Advanced Business Application Programming Objects 476
A.4 Naming Conventions for Variables and Parameter 477
A.5 Performance Guidelines 477
A.6 Summary 479
B The Authors 481

Index 483

Index

A

ABAP dictionary, 179
ABAP logic, 38, 131, 207, 214, 287
ABAP object navigator, 193
ABAP workbench, 289, 294
Action definition, 207, 209, 212–213, 217–218
Action merging, 211
Action profiles, 207
Activity, 51
Activity areas, 23
ADD_NEW_FIELDS, 222
Advanced production integration, 31
Aggregated quantities, 35
ALV grid control, 128
Append structure, 327
Application, 158
Application log, 285, 389
Application parameter, 161, 178
Application platform (AP), 35
Architecture, 29
ARM, 86
Aspects, 38
Asynchronous, 274
Asynchronous processing, 206
Authorization control, 136
Automatic replenishment, 255, 260
Availability check, 107
Availability group, 124
Available and physical stock, 226
Available stock, 406

B

Background job, 410
BAdI, 234, 237, 245, 308, 351, 354, 382
  /SCDL/TS_DATA_COPY, 327, 427
  /SCDL/TS_DATA_COPY_O, 427
  /SCWM/CL_EX_ERP_GOODSMVT_EXT, 376
  /SCWM/EX_CORE_CO_CHECK_CONF, 457
  /SCWM/EX_CORE_CO_HU_SAVE, 458
  /SCWM/EX_CORE_CO_IMPORT, 458
BAdI (Cont.)
  /SCWM/EX_CORE_CO_POST, 458
  /SCWM/EX_CORE_CO_SN_FORCE, 459
  /SCWM/EX_CORE_CO_UNP_OUTHU, 459
  /SCWM/EX_CORE_CONS, 430
  /SCWM/EX_CORE_CR_ABORT, 437
  /SCWM/EX_CORE_CR_AQUA_DATA, 437
  /SCWM/EX_CORE_CR_DEL_ITM, 438
  /SCWM/EX_CORE_CR_INT_CR, 438
  /SCWM/EX_CORE_CR_SCRAP_ZERO, 439
  /SCWM/EX_CORE_CR_SN_COMBINE, 440
  /SCWM/EX_CORE_CR_STOCK_ID, 440
  /SCWM/EX_CORE_CR_UPD_TAB_DI, 441
  /SCWM/EX_CORE_PTS_BTSQ, 447
  /SCWM/EX_CORE_PTS_CAPACHECK, 454
  /SCWM/EX_CORE_PTS_DET_PRIO, 447
  /SCWM/EX_CORE_PTS_EMPTY_BIN, 455
  /SCWM/EX_CORE_PTS_FILT_SORT, 448
  /SCWM/EX_CORE_PTS_MD_ADDBIN, 455
  /SCWM/EX_CORE_PTS_MIX, 448
  /SCWM/EX_CORE_PTS_NBIN_BLK, 450
  /SCWM/EX_CORE_PTS_NBIN_NRM, 449
  /SCWM/EX_CORE_PTS_NBIN_PAL, 449
  /SCWM/EX_CORE_PTS_NEAR_FB, 450
  /SCWM/EX_CORE_PTS_SECSQ, 451
  /SCWM/EX_CORE_PTS_SMAQ, 452
  /SCWM/EX_CORE_PTS_SRTSQ, 452
  /SCWM/EX_CORE_PTS_TYPSQ, 453
  /SCWM/EX_CORE_PTS_UPD_TAB, 453
  /SCWM/EX_CORE_PTS_VERIF, 454
  /SCWM/EX_CORE_RMS_DELETE, 441
  /SCWM/EX_CORE_RMS_DETERMINE, 442
  /SCWM/EX_CORE_RMS_HU_QUAN, 443
  /SCWM/EX_CORE_RMS_HUTYP, 442
  /SCWM/EX_CORE_RMS_NEGATIVE, 443
  /SCWM/EX_CORE_RMS_OPUNIT, 340, 444
  /SCWM/EX_CORE_RMS_QCLA_STR, 445
  /SCWM/EX_CORE_RMS_QUANTITY, 445
  /SCWM/EX_CORE_RMS_STRATEGY, 446
  /SCWM/EX_CORE_RMS_VERIFY, 446
  /SCWM/EX_CORE_WT_RT, 456
  /SCWM/EX_DLV_AVAIL_CHECK, 429
  /SCWM/EX_DLV_DET_ADDMEAS, 429

BAdI (Cont.)
/SCWM/EX_DLV_DET_AFTER_CHANGE, 429
/SCWM/EX_DLV_DET_AFTER_SAVE, 429
/SCWM/EX_DLV_DET_AT_SAVE, 429
/SCWM/EX_DLV_DET_GM_BIN, 429
/SCWM/EX_DLV_DET_HIER_CORR, 429
/SCWM/EX_DLV_DET_LOAD, 429
/SCWM/EX_DLV_DET_POD_REL, 429
/SCWM/EX_DLV_DET_PROCTYPE, 429
/SCWM/EX_DLV_DET_REJ, 429
/SCWM/EX_DLV_DET_ROUTE, 429
/SCWM/EX_DLV_EGR2PDI_AFTERCOPY, 430
/SCWM/EX_DLV_EGR2PDI_COMPARE, 430
/SCWM/EX_DLV_EGR2PDI_TEXT, 430
/SCWM/EX_DLV_EGR2PDI_VAL, 430
/SCWM/EX_DLV_GM, 343
/SCWM/EX_DLV_UI_SCREEN, 331
/SCWM/EX_ERP_ERROR_QUEUE, 425
/SCWM/EX_ERP_GOODSMVT_EXT, 373
/SCWM/EX_ERP_INT_CONF, 425
/SCWM/EX_ERP_MAPIN_ID_REPLACE, 424
/SCWM/EX_ERP_MAPIN_ID_SAVEREPL, 424
/SCWM/EX_ERP_MAPIN_OD_SAVEREPL, 424
/SCWM/EX_ERP_MAPOUT_DELINFO, 431
/SCWM/EX_ERP_MAPOUT_ID_CONFDEC, 431
/SCWM/EX_ERP_MAPOUT_ID_REPLACE, 431
/SCWM/EX_ERP_MAPOUT_ID_REPLICA, 431
/SCWM/EX_ERP_MAPOUT_ID_SPLIT, 432
/SCWM/EX_ERP_MAPOUT_OD_CHANGE, 432
/SCWM/EX_ERP_MAPOUT_OD_CONFDEC, 432
/SCWM/EX_ERP_MAPOUT_OD_REPLICA, 432
/SCWM/EX_ERP_MAPOUT_OD_SPLIT, 432
/SCWM/EX_ERP_PRIOP, 427
/SCWM/EX_ERP_PROD, 426
/SCWM/EX_EXCP_EXC_BLKBINS, 469
/SCWM/EX_EXCP_EXC_FLT, 470
/SCWM/EX_EXCP_EXC_FLT_FOLLOUP, 471
/SCWM/EX_EXCP_EXC_FUNC, 471
BAdI (Cont.)
/SCWM/EX_HU_BASICS_HUHDR, 334
/SCWM/EX_MAPIN_OD_SAVEREPL, 323
/SCWM/EX_MSL_FILL_FD, 432
/SCWM/EX_MSL_FILL_PRD_INB, 432
/SCWM/EX_MSL_FILL_PRD_OUTB, 432
/SCWM/EX_MSL_FILL_SPC, 432
/SCWM/EX_MSL_MESSAGE_SORT, 432
/SCWM/EX_RF_FLOW_POST, 205
/SCWM/EX_RF_FLOW_PRE, 205
/SCWM/EX_RSRC_QU_DET, 456
/SCWM/EX_TOWHR_PTO_CREA, 328
/SCWM/EX_VAL_PROD_REF, 433
/SCWM/EX_WAVE_PLAN, 434
/SCWM/EX_WAVE_SAVE, 435
/SCWM/EX_WHO_CREATE, 462
/SCWM/EX_WHO_DSTGRP, 462
/SCWM/EX_WHO_FLT_IL, 463
/SCWM/EX_WHO_FLT_SL, 464
/SCWM/EX_WHO_HDR_PROCTY, 465
/SCWM/EX_WHO_LIM_OVERRULE, 465
/SCWM/EX_WHO_PACK_CHECK, 468
/SCWM/EX_WHO_PACK_DIM, 467
/SCWM/EX_WHO_PACK_DSTGRP, 467
/SCWM/EX_WHO_PACKING, 468
/SCWM/EX_WHO_PMAT_CHECK, 466
/SCWM/EX_WHO_PMAT_DET, 466
/SCWM/EX_WHO_SORT, 334, 463
/SCWM/EX_WRKC_PACK, 235
/SCWM/EX_WRKC_PACK_QTY_PROPOSE, 236
/SCWM/EX_WRKC_UI_AFTER_SAVE, 236, 244
/SCWM/EX_WRKC_UI_DEST_BIN, 235
/SCWM/EX_WRKC_UI_DETA_SCREENS, 236
/SCWM/EX_WRKC_UI_DETAIL_TABS, 236
/SCWM/EX_WRKC_UI_DETERMINE_HU, 235
/SCWM/EX_WRKC_UI_FLAG_REPACK, 235
/SCWM/EX_WRKC_UI_GET_WEIGHT, 235
/SCWM/EX_WRKC_UI_GUI_STATUS, 236
/SCWM/EX_WRKC_UI_HU_CHANGED, 235
/SCWM/EX_WRKC_UI_PAMT_FR_IDENT, 236
/SCWM/EX_WRKC_UI_PO_PROP, 236
/SCWM/EX_WRKC_UI_PRODUCTMASTER, 236

/SCWM/EX_WRKC_UI_SCAN_SCREENS, 236, 243
/SCWM/EX_WRKC_UI_TREE_CONTROL, 236, 238
/SCWM/EX_WRKC_UI_WHTA_DCONS, 235
SMOD_V50B0001, 323
BAdI builder, 275, 335
BAPI_BATCH_SAVE_REPLICA, 117
BAPI_DELIVERY_GETLIST, 279
BAPI_MATERIAL_AVAILABILITY, 107
BAPI_OUTB_DELIVERY_CONFIRM_DEC, 106
BAPI_OUTB_DELIVERY_REJECT, 106
BAPI_OUTB_DELIVERY_SPLIT_DEC, 106
Barcode, 179, 188, 315
Batch, 115, 274, 399
characteristics, 287, 300, 399
class, 399
maintenance, 274
management, 274, 398
master, 276, 292
split item, 36, 342
status, 399
BMS, 140
Buffer, 292
Business context, 199
Business documents (BDocs), 221
Business logic, 55, 157
Business object (BO), 37
Business object processing framework (BOPF), 33
Business partner, 115
Business partner relationships (BUPR), 222
Business system group, 275

C

Capacity check, 69
Central model class, 244
CF, 282
Change request, 103, 106
Changing indicator, 270
Checkpoint group, 162, 178
/SCWM/WAVE, 434
/SCWM/WHO, 461
CIF model, 283
CL_MAP, 119
Class
/SCDL/CL_BO_MANAGEMENT, 393
/SCDL/CL_SP_PRD_OUT, 393
/SCWM/CL_BATCH_APPL, 398
/SCWM/CL_DLV_MANAGEMENT_DR, 391
/SCWM/CL_DLV_MANAGEMENT_FD, 391
/SCWM/CL_DLV_MANAGEMENT_PRD, 391
/SCWM/CL_DLVPACK_IBDL, 234
/SCWM/CL_EG_SP_MS, 140
/SCWM/CL_EGF_CHART_DATA, 140
/SCWM/CL_EI_WRKC_UI_AFTER_SAVE, 245
/SCWM/CL_EXCEPTION_APPL, 201
/SCWM/CL_LOG, 389
/SCWM/CL_MON_STOCK, 405
/SCWM/CL_RF_BLL_SRVC, 418
/SCWM/CL_SR_BO, 416
/SCWM/CL_SR_BOM, 416
/SCWM/CL_SR_DB_TU, 414
/SCWM/CL_SR_TU_QUERY, 414
/SCWM/CL_UI_STOCK_FIELDS, 386
/SCWM/CL_WM_PACKING, 180, 234, 244, 404
Class characteristics, 287
Class reference, 42
CLEANUP, 385
COMMIT, 385
Complex selection criteria, 39, 42–43
Composite enhancement spot
/SCWM/ESC_CORE, 437
/SCWM/ESC_MAIN, 421
/SCWM/ESC_WAVE, 434
Condition technique, 335
Confirmed WT, 66
Consolidation group, 62, 430
Constants, 388
/SCDL/IF_DL_C, 389
/SCWM/IF_DL_C, 389
WMEGC, 388, 401
WMESR, 388
Control parameters, 292
Core interface, 115, 121, 257
Core RF transactions, 205
Counting, 60
CRM condition technique, 214
Custom data, 424
Custom development, 253, 260, 268

Custom field extensions, 220
Custom fields, 226
Custom message type, 432
Customer relationship management (SAP CRM), 86, 91, 220, 222
Customizing includes, 221
Customizing tables, 26
Cycle counting, 255

D

Database inconsistencies, 384
Database selection, 388
Deconsolidation, 61
Delivery, 30, 50, 227
confirmation, 106
header, 35
interface, 92, 95, 121
item, 35
note, 321
object, 41
order, 228
processing, 29
query, 42–45
split, 103, 105
split item, 36, 46
synchronization, 355
terms, 40
Delivery documents, 30, 32, 42
Delivery request, 424
Design, 172
Destination data, 68
Destination HU, 239–240
Destination location, 65
Destination stock, 87
Difference Analyzer, 377
Display profile, 158
DLV, 282
Document architecture, 32
Document category, 30, 36, 42
Document flow, 45
Door, 56, 226, 268, 321, 348
Door bins, 52
Door determination, 54
Drill-down, 137
Dynamic processing, 281

E

Easy enhancement, 136
Easy enhancement workbench (EEW), 127, 136, 220, 222, 226
EEWB, 221
EEWC, 221
structure, 288
Easy graphics framework (EGF), 127, 251
object, 140, 148
object service providers, 149
Empty transaction, 174
Enhancement spot
/SCDL/TS_EXT, 427
/SCWM/ES_CORE_CO, 457
/SCWM/ES_CORE_CONS, 430
/SCWM/ES_CORE_CR, 437
/SCWM/ES_CORE_PTS, 447
/SCWM/ES_CORE_RMS, 441
/SCWM/ES_CORE_WT_RT, 456
/SCWM/ES_DLV_DET, 429
/SCWM/ES_DLV_EGR2PDI, 430
/SCWM/ES_DLV_GM, 343
/SCWM/ES_DLV_TOWHR, 329
/SCWM/ES_DLV_UI_SCREEN, 332
/SCWM/ES_ERP_ERROR_HANDLING, 425
/SCWM/ES_ERP_GOODSMVT, 376
/SCWM/ES_ERP_INT_CONF, 425
/SCWM/ES_ERP_MAPIN, 326, 424
/SCWM/ES_ERP_MAPOUT, 431
/SCWM/ES_ERP_PRIOP, 427
/SCWM/ES_ERP_PROD, 426
/SCWM/ES_EXCP_EXC, 469
/SCWM/ES_HU_BASICS, 335
/SCWM/ES_RSRC_QU, 456
/SCWM/ES_WHO, 334, 462
/SCWM/EX_CORE_RMS, 340
Enterprise service, 38
Entitled, 396
Entry screen, 237
ERP and EWM stock models, 118
ERP integration, 256, 262
ERP version control, 92
Exception code, 87, 198, 201
DIFF, 199
LIST, 199
NEXT, 198
SKFD, 199

Exception handling, 198
Exceptions, 181
Expected goods receipt (EGR), 31, 104, 227
Expected goods receipt notification, 31
Extending processes, 253

F

Field control, 292
Final delivery, 424
Flexibility of software, 15, 22, 127
Floor plan manager (FPM), 229
Flow chart, 164
Follow-on documents, 428
Follow-up processing, 91
Freight order, 226
Function builder, 279
Function code, 161, 177
Function group, 161
/SCWM/RF_INQUIRY, 193
/SCWM/UI_PACKING, 240
ZBKS_WAVE_KPI, 142
ZRF_SORT, 174
ZUI_PACKING, 239
Function module, 48, 102, 136, 383, 387, 395–396, 400, 402, 407, 410, 417–418, 437
/SCWM/API_PACKSPEC_CHANGE, 418
/SCWM/API_PACKSPEC_COPY, 418
/SCWM/API_PACKSPEC_CREATE, 418
/SCWM/API_PACKSPEC_DELETE, 418
/SCWM/API_PACKSPEC_GETLIST, 417
/SCWM/API_PACKSPEC_READ, 417
/SCWM/CALL_PACKUI, 237
/SCWM/CONVERT_DATE_TIME, 387
/SCWM/CONVERT_TIMESTAMP, 387
/SCWM/HU_SELECT_GEN, 402
/SCWM/MATERIAL_READ_MULTIPLE, 395
/SCWM/MATERIAL_READ_RANGE, 395
/SCWM/MATERIAL_READ_SINGLE, 395
/SCWM/MATERIAL_WHST_MAINT_MULT, 396
/SCWM/PACKING_UI, 233
/SCWM/PS_FIND_EVALUATE_MULTI, 417
/SCWM/RF_*_EXCEPTION, 202
/SCWM/RF_EAN128_SPLIT_VALID, 188
/SCWM/RF_MATNR_VALID, 190
Function module (Cont.)
/SCWM/RF_PICK_BIN_CHECK, 187
/SCWM/RF_PICK_MATID_CHECK, 190
/SCWM/RSRC_USER_DEF_SET_GET, 192
/SCWM/STOCK_CHANGE, 407
/SCWM/TO_CANCEL, 400
/SCWM/TO_CONFIRM, 400
/SCWM/TO_CREATE, 400
/SCWM/TO_CREATE_MOVE_HU, 400
/SCWM/TO_GET_WIP, 400
/SCWM/TO_READ_MULT, 400
/SCWM/TO_READ_SINGLE, 400
/SCWM/WAVE_MERGE_EXT, 410
/SCWM/WAVE_RELEASE_EXT, 410
/SCWM/WAVE_SELECT_EXT, 410
/SCWM/WAVE_SPLIT_EXT, 410
/SCWM/WAVEHDR_O_MON, 143
/SCWM/WHO_CREATE, 462

G

Global trade services (SAP GTS), 50
Goods issue (GI), 77, 206, 321
Goods movement interface, 92, 108, 374, 376
Goods receipt (GR), 52, 58, 122
GUI status, 163
GUID (Global Unique Identifier), 71, 73, 387

H

Handheld device, 157
Handling unit (HU), 73, 226, 402, 404
Hanging queue entries, 425
Header identification, 35
Hierarchy elements, 36
Hotspots, 131
HTML templates, 180
HU type, 320, 333

I

Identification type, 269
Implementation Guide (IMG), 223, 230, 235, 243, 421

Inactive delivery, 274
Inbound delivery, 31, 247, 302, 394
  change, 103
  notification, 30, 274
  replication, 102
Inbound message processing, 96
Inbound process, 254, 264
Include
  /SCWM/IRF_SSCR, 174
Index tables, 77
Indirect labor task, 226
Inspection, 84, 90
  document, 82, 86
  lot, 113
  rules, 80, 82
Inspection object type (IOT), 88
Integration, 47, 89
Integration of non-SAP ERP systems, 91
Integration with SAP ERP, 29, 91
Intermediate document (IDoc), 55, 121
Internal action, 87
Internal process codes, 199
Inventory management, 29, 70, 407
Item category, 36
Item hierarchy, 46
Item identification, 35
Item text, 374

K

Key aspect, 38, 40
Keyboard scanner, 230
Kitting, 49

L

Languages, 175
Lean inventory management engine (LIME), 70, 73
LE-TRA, 94
License plate, 268
List, 192
Loading, 63, 321
Location independent stock type, 124
Location index, 72
LOG_LE_INTEGRATION, 93, 100, 105–107
Logical field names, 43
Logical system, 275
Logical unit of work (LUW), 206, 384–385, 401, 405
Logistics execution, 96
Low stock check, 204

M

Mass access, 387
Mass tests, 384
Master data, 49, 117, 222, 254, 262, 306
Master data layer, 395
Master guide for SAP EWM, 93
Master menu, 169
Material document, 376
Maturation, 283
MBGMCR, 109
MDL, 395
Measurement service, 133, 138
Memory, 401, 403
Message box, 39
Message class, 269
  Z108, 343
Message control, 207
Message maintenance transaction, 284
Messagelog, 101–102
Method, 383
  ABORT, 437
  ASSIGN_CLASS, 431
  CANCEL_BUFFERED_WT, 454
  CHANGE, 334, 445
  CHANGE_DATA, 437
  CHANGE_WAVES, 435
  CHECK_CONF, 457
  CHECK_DOCUMENT, 343
  CHECK_REFERENCE, 433
  CHECK_SMAQ_LGTYPG, 452
  CONF, 458
  CONTROL_BLKBINS, 470
  CUSTOM_CAPA_CHECK, 454
  DECIDE_SMAQ_USE, 452
  DEFINE_ITEM_EXTENSION, 332
  DEL_ITM, 438
  DELETE, 441
  DELETE_EMPTY_BIN_BUFFER, 455
  DET_DEST_BIN_BULK, 450

Method (Cont.)
  DET_DEST_BIN_NORMAL, 449
  DET_DEST_BIN_PALLET, 450
  DET_DOCTYPE, 426
  DET_ERP_DLVTYPE, 426
  DET_ERP_MESSAGES, 432
  DET_ITEMTYPE, 426
  DET_NEAR_FB, 451
  DET_PRIORITIES, 447
  DETERMINE, 442, 456
  DETERMINE_BIN, 455
  DETERMINE_EMPTY_BINS, 455
  DSTGRP, 430, 462
  EXECUTE_FUNCTIONS, 471
  EXIT_SAPLV50K_007, 324
  FILL_FIELDS, 434
  FILT_SORT, 448
  FILTER_EXCEPTION_CODES, 470
  FILTER_FOLLOW_UP_ACTIONS, 471
  FILTER_IL, 463
  FILTER_SL, 464
  FORCE, 459
  HDR_PROCTY, 465
  HU_QUAN, 443
  HUTYP, 443
  INSERT, 439
  LIMIT_OVERRULE, 465
  MAPIN, 424
  MAPOUT, 432
  MIX_ALLOWED, 448
  MIX_CHECK, 448
  MODIFY_ADDMEAS_HEAD, 428
  MODIFY_ADDMEAS_ITEM, 428
  MODIFY_DATE_HEAD, 428
  MODIFY_DATE_ITEM, 428
  MODIFY_DELTERM_ITEM, 428
  MODIFY_EEW_HEAD, 428
  MODIFY_EEW_ITEM, 428
  MODIFY_HU, 428
  MODIFY_INCOTERMS_HEAD, 428
  MODIFY_PARTYLOC_HEAD, 428
  MODIFY_PARTYLOC_ITEM, 428
  MODIFY_PRCODES_ITEM, 428
  MODIFY_PRODUCT_ITEM, 428
  MODIFY_PRODUCT_ITEM_EXT, 428
  MODIFY_REFDOC_HEAD, 428
  MODIFY_REFDOC_ITEM, 428
  MODIFY_STOCK_ITEM, 428
  MODIFY_TRANSP_HEAD, 428
Method (Cont.)
  NEGATIVE, 444
  OPUNIT, 340, 444
  PACK_CHECK, 468
  PACK_DIMENSION, 467
  PACK_DSTGRP, 468
  PACKING, 469
  PMAT_CHECK, 466
  PMAT_DETERMINATION, 466
  POST, 459
  PRODUCT_ADD, 426
  PRODUCT_CREATE, 426
  PUBLISH_Q_ERROR, 425
  QUANTITY, 445
  REACHTIME_OVERWRITE, 457
  SAVE, 427
  SCRAP_ZERO, 439
  /scwm/cl_rf_bll_srvc=>init_listbox, 191
  /scwm/cl_rf_bll_srvc=>insert_listbox, 191
  /scwm/cl_rf_bll_srvc=>set_prmod, 204
  /SCWM/CL_RF_BLL_SRVC=>set_scr_tabname, 194
  SN_COMBINE, 440
  SORT, 334, 433
  SORT_SEQUENCE, 452
  STOCK_ID, 440
  STORAGE_BINTYPE_SEQ, 447
  STORAGE_SECTION_SEQ, 451
  STORAGE_TYPE_SEQ, 453
  STRATEGY, 446
  UNPACK_OUTER_HU, 460
  UPD_TABLES_DEL_ITEM, 441
  UPDATE_TABLES, 453
  VERIFY, 446
MFS telegram, 226
Middleware, 120
Mobile device, 157
Mobile printer, 364
Monitor node, 299
Monitor tree, 134
Monitoring, 127

N

Node hierarchy, 134, 299
Node profile, 299
Non-SAP ERP systems, 119

O

Object instance, 33, 45
Object manager, 42
Object navigator, 168
Organizational data, 64
Outbound delivery, 31, 229
Outbound delivery order (ODO), 31, 44, 344, 394
  form view, 320, 331
  goods issue, 342
Outbound delivery request (ODR), 30, 323, 327
Outbound message processing, 96
Outbound process, 254, 265, 348

P

Pack profile, 333
Package
  /SCWM/PACKING, 233
Packaging material, 307, 333, 336
Packaging specification, 226, 228, 304, 320, 333, 418
  level, 339
Packing, 63
Pager number, 268
PAI, 163
PAI modules, 241
Parallel WT, 67
PBO, 163
PBO module, 240
Performance, 292, 384, 386–387, 405–406, 411
Periodic physical inventory, 17, 254–255, 373
Personalization profile, 159
Physical inventory
  document, 373
  low stock check, 373
  periodic, 373
  product based, 373
  putaway, 373
  stock differences, 373
  storage bin based, 373
Physical stock, 406
Pick handling unit (Pick HU), 239–240, 321, 333
Pick path, 62
Pick-by-voice, 157
Picking, 62–63
Picking label, 362, 364, 371
Picking strategy, 67
Planned HU, 250
Planned WT, 64, 66
Post processing framework (PPF), 50, 55, 127, 206, 215
  action, 53
  action definition, 208–209
  applications, 208
  delivery processing, 217
  manager, 208
  merging, 211
  processing time, 210
  schedule condition, 213
  start condition, 213
  trigger, 208, 211
Posting change, 31, 96, 407
Posting change request, 31
Preceding documents, 32
Preconfigured Warehouse, 253, 263, 333, 373, 380
  functional scope, 255
  installation, 261
  introduction, 254
  organizational structure, 256
  process flows, 257
  testing, 262
Preliminary inspection, 89
Presentation device, 165
Presentation profile, 159, 316
Process functions, 204
Processing delivery, 424
Process-oriented storage, 25, 58, 63, 259
Product inspections, 81
Product master, 395
Product master record, 396, 426
Production date, 287
Production material request, 31, 96, 100
Putaway, 58, 61, 255
Putaway control indicator, 282, 285
Putaway strategy, 69, 282

Q

QIE_COMMUNICATION, 112
QIE_RFC_CONF_CANCEL_EXT_INSP, 113
QIE_RFC_CONF_EXT_INSP, 113
QIE_RFC_NOTIFY_RES_EXT_INSP, 114
QIE_RFC_STATUS_INFO_EXT_INSP, 114
QM interface, 92, 111
QPLEXT_COMM_TEC, 113
QPLEXT_RFC_INSP_LOT_CANCEL, 114
QPLEXT_RFC_INSP_LOT_CHANGE, 114
QPLEXT_RFC_INSP_LOT_CREATE, 114
Quality control, 61, 86
Quality inspection, 61, 88, 233, 255
Quality inspection document, 226
Quality inspection engine (QIE), 50, 80, 85, 88, 111, 209
Quality management (QM), 29, 80, 83, 259
Quant, 74
Quant separation, 49
Quantity classification, 304, 308, 339
Quantity offsetting, 45
Quantity role, 44–45, 47
Quarantine period, 283
Queue, 282
Queue deregistration, 124
Queue name, 401
Queue prefixes, 401
Queued remote function call (qRFC), 112, 121, 220, 401

R

Radio frequency, 60, 179
Radio frequency framework (RFF), 127, 174, 184, 191, 251, 418
Read options, 279
Reason for movement, 374, 376
Receiving, 258
Reference, 374
Reference document, 35, 76
Release upgrade, 384
Remote-enabled module, 279
Repack HU, 235–236
Repacking, 265
Report
  /SCWM/RCALL_PACK_IBDL, 233
  /SCWM/RCALL_PACK_OBDL, 233
  /SCWM/RHU_PACKING, 233
  /SCWM/RPACKENTRY, 232
  /SCWM/RPOPACKENTRY, 233
  /SCWM/RQINSPENTRY, 233
  /SCWM/RSPREADENTRY, 232
  /SCWM/RVASENTRY, 233
  ZRCALL_PACK_IBDL, 248
Resource management, 255
RF cookbook, 158
RF device, 200
RF environment, 157, 159
RF menus, 158
RF putaway, 301
RF template screens, 167
RF transaction, 89, 157, 170, 173
RFC connection, 275
ROLLBACK, 385
Rollback, 49
Route, 321
Route determination, 50
Routing, 83

S

SAP Basis, 206
SAP Chart Designer, 148
SAP Community Network (SCN), 21, 208
SAP ECC, 424
SAP EWM, 21, 253
SAP EWM PID, 375
SAP GUI, 180, 191, 233
SAP Help Portal, 264, 303
SAP ITSmobile, 158
SAP List Viewer (ALV), 227
SAP NetWeaver, 206
SAP Solution Manager, 254, 256, 262
/SAPAPO/CIF_BATCH_INBOUND, 117
/SAPAPO/CIF_CLAF_INB, 117
/SAPAPO/CIF_LOC_INBOUND, 116
/SAPAPO/CIF_PROD_INBOUND, 116
Scanner area, 231
Schedule condition, 208, 215, 217

SCM master data, 222–223
Screen BAdI, 230
Screen painter, 168, 289
/SCWM/ACTLOG, 390
/SCWM/CL_ACC_ERP, 107
/SCWM/CL_AVAIL_CHECK_ERP, 107
/SCWM/CL_DEF_IM_ERP_INT_CONF, 275
/SCWM/CL_DISPA_W2IM_GOODSMVT, 109
/SCWM/CL_MAPIN, 97
/SCWM/CL_MAPOUT, 102
/SCWM/CL_MAPOUT_ID_CONF_DEC, 104
/SCWM/CL_MAPOUT_ID_REJECT, 104
/SCWM/CL_MAPOUT_ID_REPLACE, 103
/SCWM/CL_MAPOUT_ID_REPLICA, 102
/SCWM/CL_MAPOUT_ID_RESPONSE, 103
/SCWM/CL_MAPOUT_ID_SPLIT_DEC, 103
/SCWM/CL_MAPOUT_OD_CONF_DEC, 106
/SCWM/CL_MAPOUT_OD_REJECT_CRM, 106
/SCWM/CL_MAPOUT_OD_SAVEREPLICA, 105
/SCWM/CL_MAPOUT_OD_SPLIT_DEC, 105
/SCWM/CL_TM, 384
/SCWM/ERP_EGR_DELETE, 104
/SCWM/ERP_INTEGRATION, 92, 108
/SCWM/ERP_STOCKCHECK, 109
/SCWM/ERP_STOCKID_MAPPING, 119
/SCWM/ERP_TM_INTEGRATION, 109
/SCWM/ERPDETERMINATION, 94
/SCWM/ERPVALIDATION, 94
/SCWM/ES_CORE_PTS, 284
/SCWM/ES_DLV_UI_SCREEN, 227
/SCWM/ES_ERP_INT_CONF, 275
/SCWM/ES_ERP_MAPIN, 97
/SCWM/ES_GR, 227
/SCWM/ES_PS_UI, 228
/SCWM/ES_SR_READ_SAVE, 270
/SCWM/ES_VAS_UI, 228
/SCWM/EX_CORE_PTS_TYPSQ, 284
/SCWM/EX_ERP_INT_CONF, 275
/SCWM/EX_SR_SAVE, 270
/SCWM/GET_STOCKID_MAP_INSTANCE, 119
/SCWM/IDOC_INPUT_SHPMNT, 110
/SCWM/IDOC_OUTPUT_GOODSMVT_CR, 109
/SCWM/IDOC_OUTPUT_IBDLV_CONFDC, 104
/SCWM/IDOC_OUTPUT_OBDLV_CONFDC, 106
/SCWM/IDOC_OUTPUT_OBDLV_SPLTDC, 105
/SCWM/IDOC_OUTPUT_SHIPPL, 111
/SCWM/IDOC_OUTPUT_SHPMNT, 111
/SCWM/INB_DELIVERY_REPLACE, 98
/SCWM/INB_DLV_SAVEREPLICA, 98, 104–105
/SCWM/INB_PO, 99
/SCWM/IPU, 418
/SCWM/OBDLV_CHNG_QUAN_MUL, 100
/SCWM/OUTB_DLV_CANCELLATION, 100
/SCWM/OUTB_DLV_CHANGE, 100
/SCWM/OUTB_DLV_SAVEREPLICA, 99
/SCWM/PRODUCTION_WHR_MAINTAIN, 100
/SCWM/SR_INVOICE_CREATE, 107
/SCWM/VALUATION_SET, 117
/SCWM/WM_BATCH_MAINT, 117
/SCWM/WME, 390
Selection criteria, 295
Selection screen, 291
Selection variant, 132, 137
Service provider, 394
Service-oriented architecture (SOA), 33
Service-provider (SP), 37
Shipment, 226
Shipment interface, 92, 109
Shipping, 260
Shipping and receiving (S&R), 49, 54, 208, 362
Shipping cockpit (SCO), 229
SHP_IBDLV_CONFIRM_DECENTRAL, 104
SHP_OBDLV_CONFIRM_DECENTRAL, 106
SHP_OBDLV_SPLIT_DECENTRAL01, 106
SHPMNT05, 109–110
SMOD_V50B0001, 97
SMQ2, 282
SMQR, 281
Sort order, 24
Sort sequences, 23
Source data, 66
Source location, 65
/SPE_INB_DELIVERY_RESPONSE, 98
/SPE/AAC_DETERMINATION, 107
/SPE/BATCH_SAVE_REPLICA, 117
Specification, 170
/SPE/CL_DLV_DISPATCH, 97
/SPE/CREATE_NEW_BILLING_CALL, 107
/SPE/GOODSMVT_CREATE, 108–109
/SPE/INB_CALL_TRX_VL60, 105
/SPE/INB_DELIVERY_CONFIRM_DEC, 104
/SPE/INB_DELIVERY_REPLACE, 103
/SPE/INB_DELIVERY_RESPONSE, 103
/SPE/INB_DELIVERY_SAVEREPLICA, 102
/SPE/INB_DELIVERY_SPLIT, 103
/SPE/INB_EGR_CREATE_POSA, 105
/SPE/INB_EGR_CREATE_PROD, 104
/SPE/INSP_MAINTAIN_MULTIPLE, 115
/SPE/MATERIAL_STOCK_READ, 109
/SPE/OUTB_DELIVERY_SAVEREPLICA, 105
Splitting the WT, 64
SSCC barcode, 188
Staging, 63
Staging area, 321
Start condition, 208, 218
Step flow, 165
Stock, 74, 405
  attributes, 64, 299, 407
  fields, 386
  identification, 75, 180
  index table, 76
  management, 259–260
  mapping, 124
  separation, 76
  transfer, 31
  type, 119, 407
  type role, 124
Stock identification (SI), 362, 403
Storage control, 24
Storage requirements, 22
Storage section indicator, 282
Storage type search sequence, 283, 285
Structure
  ET_EXTENSION1, 324
  ET_EXTENSION2, 324
  /SCDL/INCL_EEW_DLV_ITEM_STR, 323, 331
  /SCDL/INCL_EEW_DR_ITEM_STR, 323, 325
  /SCWM/INCL_EEW_S_ORDIM, 328
  /SCWM/INCL_EEW_S_WT_CREA, 328
  /SCWM/S_ORDIM_ATT, 329
Success message, 181
Supply chain management (SAP SCM), 220, 225
Supply chain unit, 335
Support package upgrade, 384
Synchronous processing, 206

T

Table
  /SCWM/ORDIM_O, 330
Tailored measurement service (TMS), 140
Template screens, 168
Time zone, 387
TPSSHT01, 111
Transaction
  MB03, 381
  /SCWM/CAP, 233
  /SCWM/DCONS, 232
  /SCWM/DIFF_ANALYZER, 381
  /SCWM/EGF, 147
  /SCWM/EGF_CHART, 148
  /SCWM/EGF_COCKPIT, 155
  /SCWM/EGF_OBJECT, 147, 154
  /SCWM/MON, 233
  /SCWM/PACK, 232
  /SCWM/PACKSPEC, 333
  /SCWM/PI_DOC_CREATE, 381
  /SCWM/PRDI, 233
  /SCWM/PRDO, 233, 332
  /SCWM/QINSP, 233, 239
  /SCWM/RFUI, 157
  /SCWM/SEBA, 333
  /SCWM/TLR_WIZARD, 145
  /SCWM/VASEXEC, 233
  /SCWM/WM_ADJUST, 377
  SE11, 151, 179, 325
  SE19, 245, 324, 334
  SE24, 149
  SE37, 202
  SE38, 248
  SE51, 165
  SE55, 151
  SE80, 142, 168, 179, 193, 199
  SE91, 344
  SM12, 180
  SM30, 151
  SOLAR_PROJECT_ADMIN, 262
  SOLAR01, 263
  SOLAR02, 263

Transaction manager, 384
Transactional RFC, 220
Transfer service, 32
Transition service, 327
Transportation unit (TU), 50–51, 226, 268, 414, 416
Tree control, 170, 231, 238
Truck, 321

U

Unit of measure (UOM), 67, 304
alternative, 338
base, 339
operative, 320, 338–339
Unloading, 59
with storage process, 59
without storage process, 59
User interface (UI), 37, 157

V

Validation objects, 187–188
Validation profile, 160, 187–188
Value-added service (VAS), 49, 61, 63, 233
Variant maintenance, 132
VAS order, 226, 228
Vehicle, 226
Vendor batch, 287
Verification field, 187
Verification profile, 187
Virtual stock, 72

W

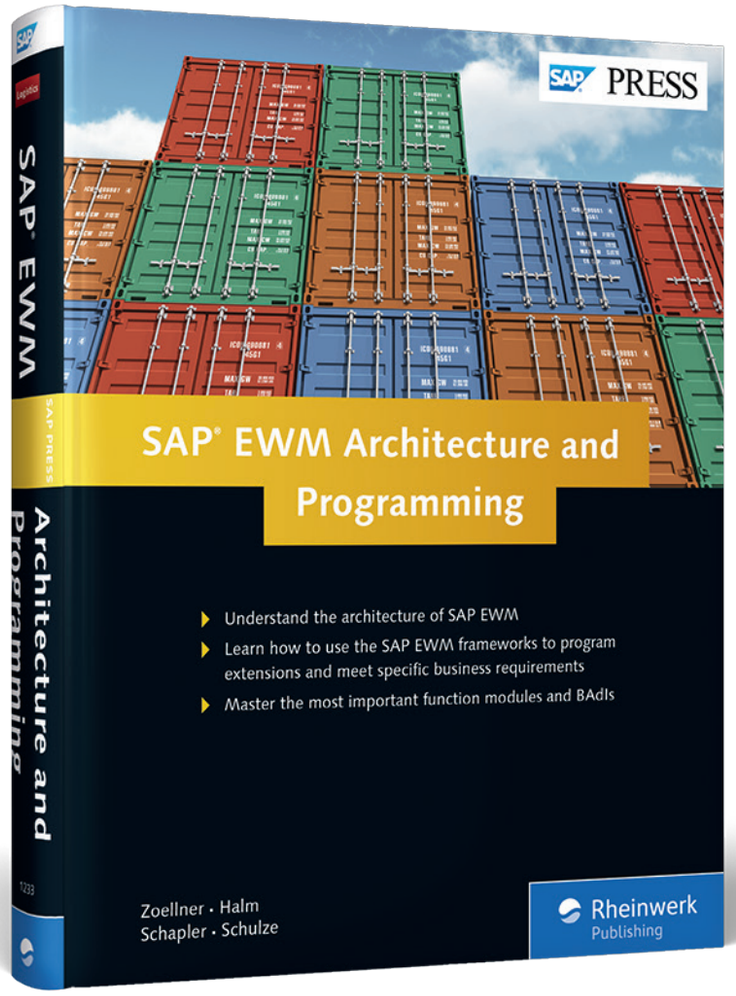
Warehouse activities, 32
Warehouse cockpit, 138, 148
Warehouse logistics, 29, 50, 259
Warehouse management monitor, 127, 251, 272, 287
category nodes, 129
enhancements, 138
hotspots, 131
methods presentation, 131
Warehouse management monitor (Cont.)
monitor methods, 131
node hierarchy, 130
node hierarchy tree, 129
profile nodes, 129
variant node, 132
Warehouse monitor, 338, 382
Warehouse number, 71, 74, 76, 387
Warehouse order (WO), 57
Warehouse order creation rule, 333
Warehouse process category, 57
Warehouse process type, 62, 87
Warehouse product, 396
Warehouse request, 30, 32, 226, 228, 392, 423
change, 393
read, 391
Warehouse request management, 258–259
Warehouse request processing, 30
Warehouse task (WT), 25, 50, 57, 226, 400
Wave, 345
Wave determination, 434
Wave functionality, 62
Wave management, 410
Web Dynpro, 229
Web Dynpro ABAP, 163
Wildcard logic, 133
Wizard, 158
WM Execution, 283
WO printing, 364, 367
Work center, 127, 230, 243, 248, 251
Workflow conditions, 212
WT processing, 47

Y

Yard, 51
Yard management, 52, 54

Z

ZSORT, 172, 174, 183, 188, 190, 192, 200, 371
ZSWAP, 350, 360



Peter Zoellner, Robert Halm, Daniela Schapler, Karen Schulze

SAP EWM Architecture and Programming

494 Pages, 2016, $79.95/€79.95
ISBN 978-1-4932-1233-0

www.sap-press.com/3860

Peter Zoellner joined SAP Consulting in 2001 and has worked for both domestic and international customers from numerous industries, specializing in logistics execution and SAP EWM since its first introduction to the market. He holds a degree in economic studies (Diplom Kaufmann) from the Catholic University of Eichstätt, Germany. Peter is married with children, and lives in the Kreuzberg district of Berlin.

Robert Halm is the head of Portfolio Management at prismat, where he focuses on mobile applications and SAP HANA solutions, especially for warehouse processes. Prior to this appointment in 2014, he served the company as a Senior Consultant and a Project Manager, and has been successfully managing a consulting team since 2011.

Daniela Schapler is a Solution Architect at SAP SE. She joined SAP SE in 1996 as a developer in the area of logistics execution, following her study of physics at the Universität Tübingen. In her role as a Solution Architect, she was involved in the design of SAP EWM from the very beginning. She was instrumental in the development of handling unit management, and has supervised several customer projects.

Karen Schulze has worked as a developer and consultant in the area of SAP EWM for more than 14 years. She is currently a Senior Consultant at abat, a company focusing on automotive industries and logistics. After finishing her studies in 1998 with a degree in mathematics and computer science from the University of Kaiserslautern, Karen started her professional career as a developer at SAP SE for warehouse management, joining the EWM development team.

We hope you have enjoyed this reading sample. You may recommend or pass it on to others, but only in its entirety, including all pages. This reading sample and all its parts are protected by copyright law. All usage and exploitation rights are reserved by the author and the publisher.