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.
In order to program in SAP EWM, we must first understand the individual software components that are part of the installation in the SAP backend 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.

<table>
<thead>
<tr>
<th>Document category</th>
<th>Description</th>
<th>Category</th>
<th>SAP EWM Transaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDR</td>
<td>Inbound Delivery Notification</td>
<td>Delivery document</td>
<td>/SCWM/IDN</td>
</tr>
<tr>
<td>ODR</td>
<td>Outbound Delivery Request</td>
<td>Delivery document</td>
<td>/SCWM/ODR</td>
</tr>
</tbody>
</table>

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

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.

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.

Hence business logic and data management are strictly separated, as well as modification and checking of the data managed. From the software application
delivery processing (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 2.2 BOPF UI to Object /SCDL/PDO](image)

![Figure 2.3 Construction of a Delivery Object in SAP EWM](image)
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.

<table>
<thead>
<tr>
<th>Database Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/SCDL/DB_REQH</td>
<td>Inbound Delivery Notification/Outbound Delivery Request: Header</td>
</tr>
<tr>
<td>/SCDL/DB_REQI</td>
<td>Inbound Delivery Notification/Outbound Delivery Request: Item</td>
</tr>
</tbody>
</table>

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
- /SCW1/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 \texttt{/SCDL/CL_SP} and \texttt{/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](image)

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 (\texttt{/SCDL/S_SP_A_HEAD_DATE}) and an aspect for dates data at item level (\texttt{/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 \texttt{/SCDL/IF_SP_C} and \texttt{/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 \texttt{/SCDL/CL_SP_PRD_OUT} and \texttt{/SCDL/CL_SP_MESSAGE_BOX}.

```plaintext
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.

```plaintext
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 = '804000094'.
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 = '18'.
APPEND ls_selections TO lt_selections.
```

Listing 2.2 Complex Selection Criteria
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_delterm 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 = /scdl/if_sp_c=>sc_asp_item_delterm
IMPORTING
  outrecords = lt_a_item_delterm
  rejected = lv_rejected
  return_codes = lt_return_codes ).
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`.

<table>
<thead>
<tr>
<th>Document category</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDR</td>
<td>/SCWM/CL_DLV_MANAGEMENT_DR</td>
</tr>
<tr>
<td>ODR</td>
<td>/SCWM/CL_DLV_MANAGEMENT_DR</td>
</tr>
<tr>
<td>POR</td>
<td>/SCWM/CL_DLV_MANAGEMENT_DR</td>
</tr>
<tr>
<td>EGR</td>
<td>/SCWM/CL_DLV_MANAGEMENT_DR</td>
</tr>
<tr>
<td>PDI</td>
<td>/SCWM/CL_DLV_MANAGEMENT_PRD</td>
</tr>
<tr>
<td>PDO</td>
<td>/SCWM/CL_DLV_MANAGEMENT_PRD</td>
</tr>
<tr>
<td>SPC</td>
<td>/SCWM/CL_DLV_MANAGEMENT_PRD</td>
</tr>
<tr>
<td>WMR</td>
<td>/SCWM/CL_DLV_MANAGEMENT_PRD</td>
</tr>
<tr>
<td>FDO</td>
<td>/SCWM/CL_DLV_MANAGEMENT_FD</td>
</tr>
<tr>
<td>PWR</td>
<td>/SCWM/CL_DLV_MANAGEMENT_PRD</td>
</tr>
</tbody>
</table>

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).

```plaintext
DATA:  
  lo_delivery TYPE REF TO /scwm/cl_dlv_management_prd,  
  lo_message TYPE REF TO /scwm/cl_dlv_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.
```

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

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_PROCESSING**. 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 **M1** (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.

```abap
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 = lsHierarchy-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.
```

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).

```abap
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).

```abap
IF sy-subrc = 0 AND ls_addmeas-qty NE 0.
  MESSAGE e013(zyewm) WITH
    <item>-itemno. "not yet packed completely"
ENDIF.
```

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
Architecture

2

/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,
lt_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
  lv_bname = sy-uname
  lt_create_whr = lt_create_whr
  lv_update_task = abap_false
  lv_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_UI_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 quanat 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. WTIs 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.

<table>
<thead>
<tr>
<th>Yard Management in SAP EWM</th>
</tr>
</thead>
</table>

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 WTIs 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](image_url)
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.

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

Table 2.4 Packages for 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_DD. The business logic is executed through the associated classes /SCWM/CL_SR_BOM, 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_BD, 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.

<table>
<thead>
<tr>
<th>Object</th>
<th>Table</th>
<th>Activity</th>
<th>Additional table</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vehicle</td>
<td>/SCWM/VEHICLE</td>
<td>/SCWM/VEH_SR_ACT</td>
<td>/SCWM/VEH_STATUS /SCWM/VEH_IDENT</td>
</tr>
<tr>
<td>Door</td>
<td>/SCWM/TDOOR</td>
<td>/SCWM/DOOR_SR_ACT</td>
<td>/SCWM/DOOR_SR_ACT</td>
</tr>
</tbody>
</table>

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_SR_ACT 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.
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 Categories in SAP EWM

<table>
<thead>
<tr>
<th>Warehouse process category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Putaway</td>
</tr>
<tr>
<td>2</td>
<td>Picking</td>
</tr>
<tr>
<td>3</td>
<td>Internal movement</td>
</tr>
<tr>
<td>4</td>
<td>Inventory</td>
</tr>
<tr>
<td>5</td>
<td>Goods receipt</td>
</tr>
<tr>
<td>6</td>
<td>Goods issue</td>
</tr>
<tr>
<td>7</td>
<td>Posting change</td>
</tr>
</tbody>
</table>

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.

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 upload 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.

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 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.

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.

<table>
<thead>
<tr>
<th>Creation Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consolidation group</td>
<td>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.</td>
</tr>
<tr>
<td>Pick path</td>
<td>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.</td>
</tr>
<tr>
<td>Pick pack pass – system</td>
<td>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.</td>
</tr>
<tr>
<td>Pick pack pass – user</td>
<td>In this scenario, the WTs are also bundled on consolidation group per activity area, but all warehouse orders are active.</td>
</tr>
</tbody>
</table>

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>
<thead>
<tr>
<th>Table</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>/SCWM/ORDIM_O</td>
<td>Planned WTs with status open or waiting</td>
</tr>
<tr>
<td>/SCWM/ORDIM_OS</td>
<td>Serial number for open WTs</td>
</tr>
<tr>
<td>/SCWM/ORDIM_L</td>
<td>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</td>
</tr>
<tr>
<td>/SCWM/ORDIM_LS</td>
<td>Serial numbers for log-table</td>
</tr>
<tr>
<td>/SCWM/ORDIM_C</td>
<td>Completed WTs</td>
</tr>
<tr>
<td>/SCWM/ORDIM_CS</td>
<td>Serial numbers for completed WTs</td>
</tr>
<tr>
<td>/SCWM/ORDIM_H</td>
<td>Stock information for confirmed HU WTs with mixed stock</td>
</tr>
<tr>
<td>/SCWM/ORDIM_HS</td>
<td>Serial numbers on mixed HU WT referencing /SCWM/ORDIM_H</td>
</tr>
<tr>
<td>/SCWM/ORDIM_E</td>
<td>Exception codes for completed WTs</td>
</tr>
</tbody>
</table>

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.

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

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).

<table>
<thead>
<tr>
<th>Status</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open</td>
<td>WT waiting for processing</td>
</tr>
<tr>
<td>Waiting</td>
<td>WT that reserves source stock and capacity on destination but cannot be processed</td>
</tr>
<tr>
<td>Confirmed</td>
<td>WT was processed successfully</td>
</tr>
<tr>
<td>Canceled</td>
<td>Planning was canceled</td>
</tr>
</tbody>
</table>

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.

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.

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/HU_FUNC. 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 must be done. Classification of capacity check and capacity update can be useful for work center and staging bins.

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](image)

### 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

### 3.1 Warehouse Management Monitor
- Basics and Technical Structure ........................................... 127
- Enhancement Options .......................................................... 136

### 3.2 Easy Graphics Framework and Measurement Services
- Foundations ........................................................................... 138
- Custom Development: Basic Measurement Service .......... 141
- Adjust Chart Template .......................................................... 148
- Custom Development: Easy Graphics Framework
  - Object Service Provider ....................................................... 149

### 3.3 Radio Frequency Framework
- Basics .................................................................................... 158
- Radio Frequency Framework and the Web Dynpro ABAP Transaction .................................................. 163
- Create a New Display Profile ................................................ 165
- Create a New Radio Frequency Menu ............................... 169
- Specification and Design of a New Radio Frequency Transaction .................................................. 170
- Realization of the New Radio Frequency Transaction in the System .................................................. 173
- Realization of a Verification Profile ..................................... 187
- Value Helps in Radio Frequency with Function Key F8 .... 191
- Realization of Lists ............................................................... 192
- Exception Handling in Radio Frequency ......................... 198
- Process Functions in Background Mode in Radio Frequency Transactions ........................................... 204

### 3.4 Post Processing Framework
- What is the Post Processing Framework .......................... 206
- Extended Warehouse Management and Post Processing Framework .................................................. 208
- Enhancement Options of the Post Processing Framework .............................................................. 215

### 3.5 Easy Enhancement Workbench
- Introduction to Easy Enhancement Workbench ............... 220
- Extended Warehouse Management and Easy Enhancement Workbench .................................................. 222

---

## 4 Extending Processes in the Preconfigured Warehouse W001

### 4.1 Introduction to Preconfigured Warehouse for Extended Warehouse Management ........................................... 254
- Functional Scope of the Preconfigured Warehouse ............ 255
- Overview of Custom Developments within the Processes ........................................................................ 260
- Installation of Preconfigured Warehouse ......................... 261
- Testing Preconfigured Warehouse with Solution Manager ........................................................................ 262

### 4.2 Inbound Process with Repacking for Putaway (106) ........................................................................ 265
- Process Description 106 ....................................................... 266
- Custom Development 106A: Permit Activation of the Transportation Unit Only after Entering the License Plate and Pager ........................................................................ 268
- Custom Development 106B: Delay Inbound Delivery with Missing Batch .................................................. 274
- Custom Development 106C: Putaway Depending on Quarantine Period .................................................. 282
- Custom Development 106D: Enhancing the Warehouse Monitor .......................................................... 287

### 4.3 Inbound Process without Packing Information (107) ........................................................................ 301
- Process Description 107 ....................................................... 302
- Custom Development 107A: Automatic Handling Unit Creation without Packaging Specifications .......... 303
- 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_HANDLER ......................................................... 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_CORECONS ....................... 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
Contents

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

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

## B

<table>
<thead>
<tr>
<th>Term</th>
<th>Page(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Background job</td>
<td>410</td>
</tr>
<tr>
<td>BAAd, 234, 237, 245, 308, 351, 354, 382</td>
<td></td>
</tr>
<tr>
<td>/SCDLTS_DATA_COPY, 327, 427</td>
<td></td>
</tr>
<tr>
<td>/SCDLTS_DATA_COPY_0, 427</td>
<td></td>
</tr>
<tr>
<td>/SCWM/CL_EX_ERP_GOODSMVT_EXT, 376</td>
<td></td>
</tr>
<tr>
<td>/SCWM/EX_CORE_CO_CHECK_CONF, 457</td>
<td></td>
</tr>
<tr>
<td>/SCWM/EX_CORE_CO_HU_SAVE, 458</td>
<td></td>
</tr>
<tr>
<td>/SCWM/EX_CORE_CO_IMPORT, 458</td>
<td></td>
</tr>
</tbody>
</table>
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
term, 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
/SCL/TS_EXT, 427
/SCWM/ES_CORE_CO, 457
/SCWM/ES_CORE_CONS, 430
/SCWM/ES_CORE_CRE, 437
/SCWM/ES_CORE_PTS, 447
/SCWM/ES_CORE_RMS, 441
/SCWM/ES_CORE_WRT_RT, 456
/SCWM/ES_DLV_DET, 429
/SCWM/ES_DLV_EGR2PDI, 430
/SCWM/ES_DLV_GM, 434
/SCWM/ES_DLV_TOWHR, 329
/SCWM/ES_DLV_UI_SCREEN, 332
/SCWM/ES_ERP_ERROR_HANDLING, 425
/SCWM/ES_ERPGOODSMVT, 376
/SCWM/ES_ERP_INT_CONF, 425
/SCWM/ES_ERP_MAPIN, 326, 424
/SCWM/ES_ERP_MAPOUT, 431
/SCWM/ES_ERP_PBIOP, 427
/SCWM/ES_ERP_PROD, 426
/SCWM/ES_EXP_EXC, 469
/SCWM/ES_HU_BASICS, 335
/SCWM/ES_KSR_0, 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

F
Field control, 292
Final delivery, 424
Flexibility of software, 15, 22, 127
Flow chart, 164
Flow-on documents, 428
Follow-up processing, 91
Follow-on documents, 428
Freight order, 226
Function builder, 279
Function code, 161, 177
Function group, 161
/SCWM/RF_INQUIRY, 193
/SCWM/RF_PACKING, 240
ZBKS_WAVE_KPI, 142
ZRF_SORT, 174
ZUL_PACKING, 239

Function module (Cont.)
/SCWM/RF_PICK_BIN_CHECK, 187
/SCWM/RF_PICK_MATID_CHECK, 190
/SCWM/RSRC_USER_DEF_SET, 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/WAVEWHMI, 228, 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
Implementation Guide (IMG), 243, 251

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
Inbound message processing, 96
change, 103
notification, 30, 274
replication, 102
Inbound message processing, 96
Inbound process, 254, 264
Inbound message processing, 96
change, 103
notification, 30, 274
replication, 102
Inbound message processing, 96
Inbound process, 254, 264
Inbound message processing, 96
change, 103
notification, 30, 274
replication, 102
Inbound message processing, 96
Inbound process, 254, 264
Inbound message processing, 96
change, 103
notification, 30, 274
replication, 102
Inbound message processing, 96
Inbound process, 254, 264
Inbound message processing, 96
change, 103
notification, 30, 274
replication, 102
Inbound message processing, 96
Inbound process, 254, 264
Inbound message processing, 96
change, 103
notification, 30, 274
replication, 102
Inbound message processing, 96
Inbound process, 254, 264
Inbound message processing, 96
change, 103
notification, 30, 274
replication, 102
Inbound message processing, 96
Inbound process, 254, 264
Inbound message processing, 96
change, 103
notification, 30, 274
replication, 102
Inbound message processing, 96
Inbound process, 254, 264
Inbound message processing, 96
change, 103
notification, 30, 274
replication, 102
Inbound message processing, 96
Inbound process, 254, 264
Inbound message processing, 96
change, 103
notification, 30, 274
replication, 102
Inbound message processing, 96
Inbound process, 254, 264
Inbound message processing, 96
change, 103
notification, 30, 274
replication, 102
Inbound message processing, 96
Inbound process, 254, 264

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 texts, 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
MBGMR, 109
MDL, 395
Measurement service, 133, 138
Memory, 401, 403
Message box, 39
Message class, 269
Message text, 343
Message control, 207
Message maintenance transaction, 284
Messagelog, 101–102
Method, 383
ABORT, 457
ASSIGN_CLASS, 431
CANCEL BUFFERED_WT, 454
CHANGE, 334, 445
CHANGE_DATA, 437
CHANGE_WAVIES, 435
CHECK CONF, 457
CHECK DOCUMENT, 434
CHECKREFERENCE, 433
CHECK SMAQ LTGTPG, 452
CONF, 458
CONTROL BLK BINS, 470
CUSTOM CAPA CHECK, 454
DECIDE SMAQ USE, 452
DEFINE ITEM EXTENSION, 332
DEL_ITEM, 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
DETERMINE,<br>403, 462
EXECUTE FUNCTIONS, 471
EXIT SAPLV50K, 324
FILL_FIELDS, 434
FILL_SORT, 448
FILTER EXCEPTION_CODES, 470
FILTER FOLLOW UP ACTIONS, 471
FILTER IL, 463
FILTER SL, 464
FORCE, 459
HDR PROCOT, 465
HU QUAN, 443
HU Type, 443
INSERT, 439
LIMIT OVERRIDE, 465
MAPIN, 424
MAPPOT, 432
MIX ALLOWED, 448
MIX CHECK, 448
MODIFY ADDMEAS ITEM, 428
MODIFY ADDMEAS_ITEM, 428
MODIFY DATE HEAD, 428
MODIFY DATE ITEM, 428
MODIFY DELTERM ITEM, 428
MODIFY EW HEAD, 428
MODIFY EW ITEM, 428
MODIFY HU, 428
MODIFY INCOTERMS HEAD, 428
MODIFY PARTYLOC HEAD, 428
MODIFY PARTYLOC ITEM, 428
MODIFY PROCIDES ITEM, 428
MODIFY PRODUCT ITEM, 428
MODIFY PRODUCT ITEM EXT, 428
MODIFY REFDOC HEAD, 428
MODIFY REFDOC ITEM, 428
MODIFY STOCK ITEM, 428
MODIFY TRANSF HEAD, 428
Method (Cont.)
NEGATIVE, 444
OPUNIT, 340, 444
PACK CHECK, 468
PACK DIMENSION, 468
PACK DGTRG, 468
PACKING, 469
PMAT CHECK, 466
PMAT DETERMINATION, 466
POST, 459
PRODUCT ADD, 426
PRODUCT CREATE, 426
PUBLISH_Q ERROR, 425
QUERY, 445
REACHTIME OVERWRITE, 457
SAVE, 427
SECURITY, 439
<br /]>SCWM/CL_RF_BLL_SRV=>init_listbox, 191
[SCWM/CL_RF_BLL_SRV=>insert_listbox, 191<br /]>SCWM/CL_RF_BLL_SRV=>set_premod, 204<br /]>SCWM/CL_RF_BLL_SRV=>set_scr_tablename, 194<br /]>SN COMBINE, 440<br /]>SORT, 334, 433<br /]>SORT SEQUENCE, 452<br /]>STOCK ID, 440<br /]>STORAGE BINTYPE SEQ, 447<br /]>STORAGE SELECTION, 451<br /]>STORAGE_TYPE SEQ, 453<br /]>STRATEGY, 446<br /]>UNPACK OUTER HU, 460<br /]>UPD TABLES DEL ITEM, 441<br /]>UPD TABLES, 453<br /]>VERIFY, 446
<br /]>MFs telegram, 226
Middleware, 120
Mobile device, 157
Mobile printer, 364
Monitor node, 299
Monitor tree, 134
Monitoring, 127

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
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.

© 2016 by Rheinwerk Publishing, Inc. This reading sample may be distributed free of charge. In no way must the file be altered, or individual pages be removed. The use for any commercial purpose other than promoting the book is strictly prohibited.