





## Reading Sample

In this chapter excerpt, learn about revenue accounting items (RAIs) and how to use Transaction FARR\_RAI\_MON, also known as the RAI monitor. Follow step-by-step instructions to process RAIs, exempt and change items, resolve errors, and more.

-  **“Revenue Accounting Items”**
-  **Contents**
-  **Index**
-  **The Authors**

Sreten Milosavljević, Swayam Prabha Shankara

### Revenue Accounting and Reporting with SAP S/4HANA

467 pages | 11/2023 | \$99.95 | ISBN 978-1-4932-2434-0

 [www.sap-press.com/5700](http://www.sap-press.com/5700)

# Chapter 4

## Revenue Accounting Items

*Revenue accounting items (RAIs) are sent from the source system (SAP or non-SAP) ready to be processed by revenue accounting and reporting (RAR) to create contracts, perform contract modifications, or perform fulfillments in a contract. In this chapter, we'll explain RAIs in detail and provide more information from a user perspective regarding processing and some basic error handling.*

RAIs are integral to the RAR functionality in SAP S/4HANA, impacting everything from performance obligations (POBs) to contract management. We'll begin our discussion in this chapter with a walkthrough of RAI processing, before diving into RAI management, extensibility, and error handling. Next, we'll cover the usage of Business Rules Framework plus (BRFplus) for RAI integration with adjacent processes. To close, we'll provide instructions to create custom RAIs.

### 4.1 Processing Revenue Accounting Items

As mentioned in earlier chapters, RAIs come from external systems to the Adapter Reuse Layer (ARL) and are ready for processing. The ARL is part of RAR, which is the first stop for data sent from source systems. The ARL also performs data quality checks and uses BRFplus rules to transform data to structures that will represent contracts, POBs, and revenue schedules. Let's review the high-level design of the ARL, as shown in Figure 4.1.

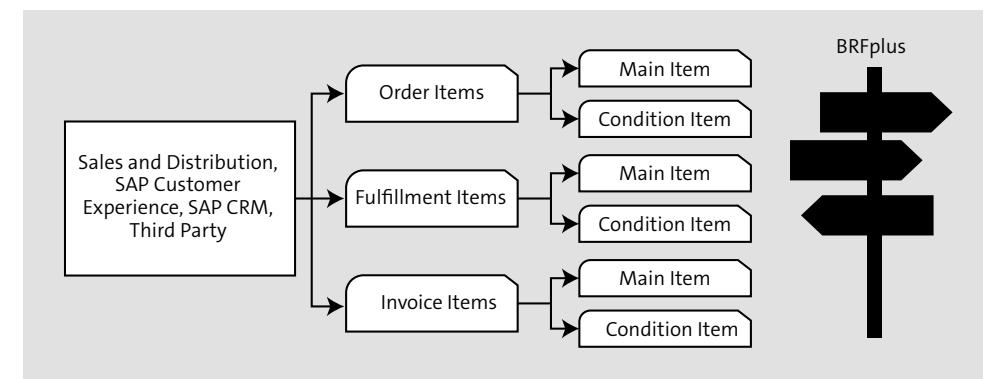


Figure 4.1 ARL Design

Once data is sent from operational systems via the standard RAR interface, it reaches the ARL. In the ARL, all data items are represented as RAIs, which are just data received and ready to be processed once it passes basic data quality checks.

All the items can be split into two categories:

- **Order items**

These items are used to perform basic contract operations: either contract creation or contract modification. If you're working with the sales and distribution interface, a sales order or sales contract will be usually represented as an order item. If you're working with external systems, it's usually data coming from customer relationship management (CRM) or order management systems. The important thing is that order items don't have a predecessor to be created.

- **Items with predecessor**

When it comes to processing fulfillments or invoices, it's essential that you know to which order item this activity refers. For example, if there is a goods issue as a fulfillment event, you need to know to which order and item this fulfillment refers.

In the following sections, we'll explain processing for both RAI categories. First, let's walk through the available methods for processing that are relevant for both categories.

#### 4.1.1 Processing Methods

All items created as RAIs have two categories: main items and condition items. Figure 4.2 shows the RAI monitor (accessed via Transaction FARR\_RAI\_MON) with a split on main and condition items.

POB(FR...	Contract(I	ItemStatus	Hist	Exist	Error	Ex	His	Ex	Send	Com...	SourceSys	SrcItmType	Source	Item ID	Subarea	RevAcc...	Header ID	ItemID	Ref	Type	Reference ID	Customer
												SD	SDOI	0099000671000010	815	SD01	99000671	10	SDO	0099000671	100088450	
												SD	SDOI	0099000672000010	580	SD01	99000672	10	SDO	0099000672	100088450	
												SD	SDOI	0411Y4167keYmm	968	SD01	877936	10				877936

Figure 4.2 Main and Condition Items in Transaction FARR\_RAI\_MON

Main items have information that is needed for identifying or creating IFRS 15 contracts. In the main items, you'll find customer, reference to document (source of RAI creation), or all dates relevant and needed for contract creation or fulfillment update, for example. Condition items contain values relevant for contracts. For order items, you'll find at least two items: condition representing transactional price and condition representing stand-alone selling price (SSP). For fulfillment items, there can be one item (if you're having proof of concept [POC] fulfillment) or none; for invoice items, you can have one or several items depending on whether discounts are represented in RAR or not.

Irrespective of the RAI type, there are several ways to process RAIs. We'll discuss them in the next sections.

#### Manual Processing

The first option is manual processing: we'll approach this activity through Transaction FARR\_RAI\_MON, which is usually the first stop for all RAR users. Once you run the transaction, a screen like Figure 4.3 will appear.

The screenshot shows the initial screen of Transaction FARR\_RAI\_MON. It features several sections for selecting criteria:

- Kind of Selection:** A dropdown menu set to "All Items Related to Order Items".
- Item Status:** A dropdown menu set to "03 Raw and Processable Items".
- Source Documents:** Fields for Component, Logical System, Type, Header ID (Orders: 800760204), Item ID (Orders), Date, and Time (00:00:00), each with a "to:" field and a selection icon.
- Master Data:** Fields for Company Code, Business Partner Number, and Customer, each with a "to:" field and a selection icon.
- Revenue Accounting Item Classes:** A field for Class with a "to:" field and a selection icon.
- Further Attributes:** Fields for Reference Type, Reference ID, Error Status, and Revenue Accounting Item ID, each with a "to:" field and a selection icon. There is also a checkbox for "Further Selections Exist".

Figure 4.3 Transaction FARR\_RAI\_MON: Initial Screen

You first need to select which items you want to see from the **Kind of Selection** dropdown:

- **Order Items**

The system will show only items that are used to create or update contract/POB data in RAR.

- **Invoice/Fulfillment**

Invoice/fulfillment are items that are made as successor items. An important thing to have in mind is that the **Header ID** field controls the document number that needs to be processed. If you select **Invoice/Fulfillment**, you need to enter the invoice number or number of the post goods issue (PGI) document.

- **All Items Related to Order Items**

Opposite to the previous option, entering the number of the order and the system will display all items related to that order, including invoices and fulfillments, if any.

- **All Items**

The system displays all items relevant to the mentioned **Header ID**.

In general, the best option to select here is **All Items Related to Order Items** because it allows you to enter just the order number and search based on that number. This option ensures that all invoices and fulfillments will be displayed together with the order.

The next option is related to **Item Status**. RAIs can change into a total of three statuses before and after being transferred to RAR:

- **01: Processable**

This means the item is ready to be processed into contracts/POBs.

- **02: Processed**

Once an item gets processed, it's stored in a separate table and changes to the **Processed** status.

- **03: Raw and Processable Items**

**Raw** is an optional status, and the system can be configured to create items in this status that aren't yet ready to be processed. So, in this case, data checks are performed before the item even reaches the **Processable** status.

Once you save a document that is then replicated as an RAI, some items are created as raw and some as processable. The system understands the processable item as the item that is completed and ready to be processed into the POB/fulfillment event/invoice. If this isn't the case, it means the item didn't pass all consistency checks and can't be processed further. However, RAR differentiates the level of issues we might have, and the following rules apply:

- If an item has some error in any key field, the item won't be created at all.
- If an item has an error, but not in key fields, it will be created as a raw RAI.
- If an item doesn't have any errors, it will be created as a processable RAI.

### Raw Status and When to Use It

Whether to use **Raw** status or go straight to **Processable** status is a common question. As usual, there is no answer to suit all customers, but the following needs to be considered: If items are initially created as raw, we're clearly adding one more step for users to execute before they can finish processing RAIs (RAI first needs to be transferred, and only after that, can it be processed in RAR). On the other hand, we're enforcing one more level of data check before the item is created. Now, the real question is whether we need this extra data consistency check. If we're using integration with sales and distribution, this most likely isn't needed because any data error would need to be corrected in the document we're trying to process. However, if we're integrating with an external system, it's highly likely that the error couldn't be corrected in the source system, so it would make sense to have **Raw** status items before they get into **Processable** status. But if there is a system doing data consistency checks before items are sent to RAR, using an additional status might not be necessary. So, before making a decision about using an extra status, all these things need to be considered.

In the **Source Documents** section, you can make all the additional selections to pinpoint the exact items requested to be displayed:

- **Component**

Represents the system to which RAR is being integrated and which will be used as the source for displaying data. Possible entries here are a result of your definition of sender components in inbound processing, as discussed in Chapter 3, Section 3.2.

- **Logical System**

Specifies the logical definition of the system that is the source for RAIs you need to process. The source item's logical system, component, type, and ID constitute the link to the source item (e.g., a sales and distribution order item) that the RAI (e.g., a sales and distribution return order) relates to.

- **Header ID (Orders)**

Reference to the original document that you try to find. This field will have different meanings depending on what was entered in **Kind of Selection**: if you select **Order Items** or **All Items Related to Order**, then the order number needs to be entered here, whereas if the **Invoice /Fulfillment** item is selected, the invoice of the PGI document number needs to be entered here.

Options in the **Master Data** section give more ways to limit data that will be displayed. For example, you can enter a **Customer** number or **Business Partner Number** to fetch the exact records you need.

Sometimes, you may need to reprocess items that are already in the error. You can select those items in the **Further Attributes** section using the **Error Status** field, as shown in Figure 4.4.

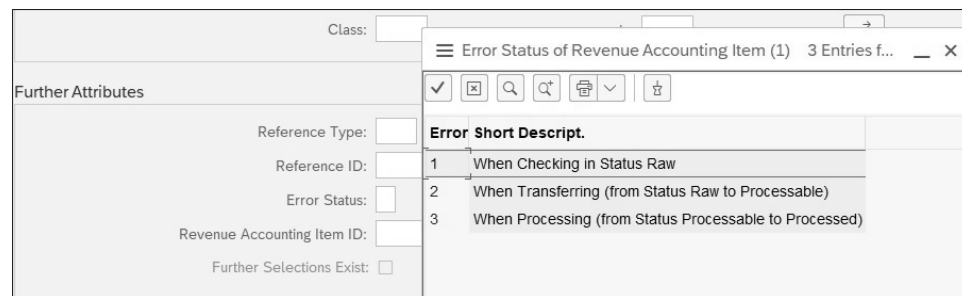


Figure 4.4 Error Status Display

As already mentioned, before being transferred to RAR, items can be in error status, which is further split into when the error actually appeared. There are in total three different statuses being triggered in different points of the RAI lifecycle, which can be used to filter RAIs further that need to be displayed. At the bottom of the screen under the **Technical Criteria** section, the technical details can be found, as shown in Figure 4.5.

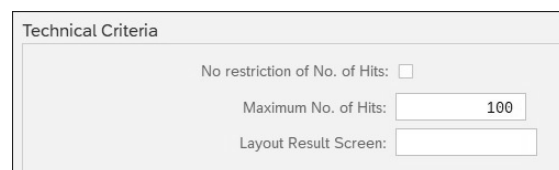


Figure 4.5 Technical Criteria for Display

Sometimes, even if you limit the selection criteria, the number of items to be displayed is too high. Because this activity can be time-consuming, by default, SAP limits the maximum number of hits to 100 main items. That can be overridden by either entering some other number in the **Maximum No. of Hits** field or simply selecting that no restriction of number of hits should be applied (**No restriction of No. of Hits**).

**Further Selections**

At the top of the initial screen of Transaction FARR\_RAI\_MON, you can see the **Further Selections** button, which you can use to access all fields in the main item structure that can be used as selection criteria. This option becomes particularly useful if you've introduced some custom fields that could be used to filter the RAI display. We'll discuss this in more detail in Section 4.3.

Once items are selected, click the **Execute** icon, and the system will display entries that fit the selections made, as shown in Figure 4.6.

In the first column, **ItemStatus**, the system will display raw, processable, or processed items depending on your selection criterion. The next important column is **Error** status. Here, the system uses a simple traffic light system to display the proper status of

the RAI: no errors equals a green light, and error equals a red light. If there is an item with an error, you can click on the item to go to the log where you'll see what caused the error while processing the RAI.

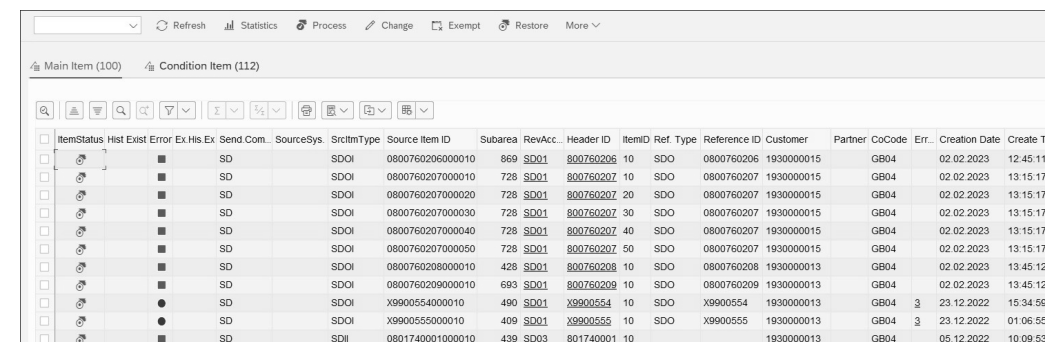


Figure 4.6 Transaction FARR\_RAI\_MON: Results

For the results just shown, you can see what error is behind the item that wasn't processed, as shown in Figure 4.7. In our example, you can see that error **C01** was raised after processing our RAI.

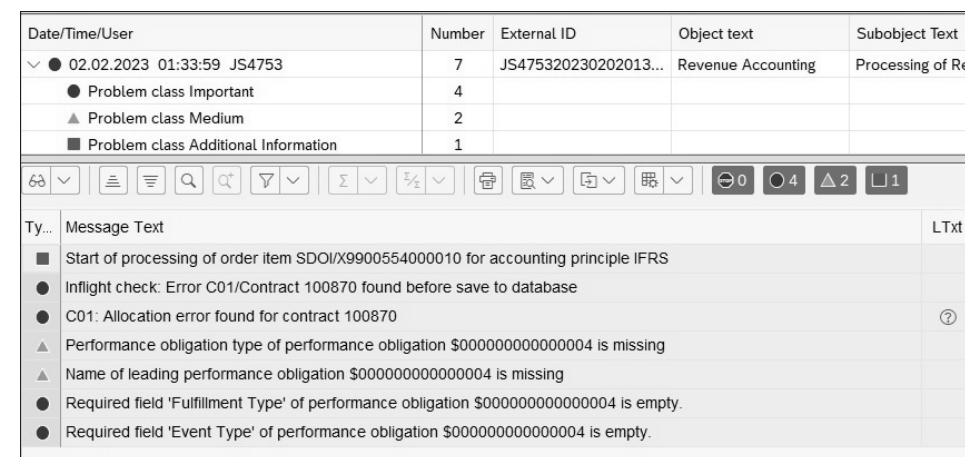


Figure 4.7 Transaction FARR\_RAI\_MON: Error

The same results can be observed if you access Transaction SLG1 for displaying system logs.

In addition, you need to understand when this error occurred, which can be checked by looking at the results table (refer to Figure 4.6) and column **Err...** on the right. Here, you can see numbers 1–3, which fit to numbers that you can enter on the initial screen and that represent the type of error as due to creation of item, change of status, or processing.

Also in the results table, you can click on the values in the **Header ID** field to go directly to the document that was used to create this RAI item. However, this functionality works only when integrating with sales and distribution.

A special explanation is needed for the **Subarea** column (refer to Figure 4.6). When we say *subareas* in RAR, it's the **KEYPP** field. The **KEYPP** field is used for the parallel processing framework (PPF). For more information, see Section 4.1.2.

The last two columns in the results table are reserved for the date (**Creation Date**) and time (**Create Time**) when the item was created. This data can't be found in the table that stores the main and condition items, but it's being decoded from the **Timestamp** field. This concept deserves additional explanation. As mentioned before, RAIs belong to the ARL, which is the integration layer between the external component and RAR. In practice, this means that an item can be created in an external component, then multiple changes can happen before the items get processed in the ARL and RAR data gets updated. All of these items will be created as separate RAIs with different timestamps, but while processing them, only the last item will be processed. This approach sometimes can create issues.

#### Example of Issue with Latest Timestamp

Let's say a user created a sales order with an item that needs to create a time-based POB, and there is a contract with a start and end date. After saving, an RAI was created, but it wasn't processed. Then, the user decided that the item needed to be rejected, so the rejection reason was used in the sales order, which was again saved. Now, in Transaction **FARR\_RAI\_MON**, there are two items, but the item with the rejection reason will have the latest timestamp and so is the only item processed. However, once the user sets a rejection reason in the sales document, this automatically triggers the population of the **Finalization Date** field with the current date. Therefore, the system tries to create a new POB, which needs to be created as terminated. This isn't possible, and it will throw an error while processing.

The solution for this exact challenge is in process organization: once created, the RAI needs to be processed. Once that has happened, then the item can be processed with the rejection reason. If the process were executed in such an order, the issue wouldn't appear because the POB would be created and then terminated. In other words, the system wouldn't try to create terminated items.

Now, when the list contains all items based on your selections, you need to perform some activities. Possible options are displayed at the top of the screen, as shown in Figure 4.8.



Figure 4.8 Transaction **FARR\_RAI\_MON**: Options

Let's walk through a few of the available buttons:

#### ■ Refresh

This button's usage is the same as in other applications in SAP: if there was some change in the external application that is used to create RAIs, this change will be displayed by clicking this button.

#### ■ Statistics

When this button is selected, the system will display a breakdown of all RAIs that fit the selection made on the first screen of Transaction **FARR\_RAI\_MON**, as shown in Figure 4.9.

Status	Message Text	Counter
RAI class SD01	Main Items Status 'PROCESSABLE'	10
RAI class SD01	Condition Items Status 'PROCESSABLE'	20
RAI class SD03	Main Items Status 'PROCESSABLE'	90
RAI class SD03	Condition Items Status 'PROCESSABLE'	92
Σ	Total Main Items	100
Σ	Total Condition Items	112

Figure 4.9 Transaction **FARR\_RAI\_MON**: Statistics Display

#### ■ Process

This button processes RAIs, that is, moves them from status **Processable** to status **Processed**. Before you can click this button, you must select items in the table that need to be processed. The important thing here is that if you have multiple items that belong to the same **Header ID** and select only one of them, all will be processed if they belong to the same source document type. The reason for this is that revenue allocation occurs on the contract level, so it's essential that all items belonging to the same contract (represented by **Header ID**) get processed.

Once items are being processed, the system will issue a message about the results of the processing. These results (irrespective of whether they are errors or successes) are written in the system log. In the bottom of the message screen, you can use the **Display Log** icon to display the log, which contains useful information about what was created/updated (or for an error, information about what issue occurred), as shown in Figure 4.10. This functionality is useful when it comes to either troubleshooting or further analysis.

In the background, the system is transferring data between processable and processed tables.

Date/Time/User	Number	External ID	Object text	Subobject
02.02.2023 19:51:32 SM6918	5	SM69182023020219...	Revenue Accounting	Processing
■ Problem class Important	3			
■ Problem class Additional Information	2			

Ty...	Message Text
■	***** PROCESS *****
■	Start of processing of order item SDOI/800760206000010 for accounting principle IFRS
■	New performance obligation 108126 created for contract 100873
■	1 items processed (record type Main Item, class SD01)
■	2 items processed (record type Condition Item, class SD01)

Figure 4.10 Transaction FARR\_RAI\_MON: Processing Results

We'll discuss two more key buttons, **Change** and **Exempt**, in Section 4.2 during our coverage of RAI management.

### Automatic Processing

Often, the number of RAIs being created is substantial, and running Transaction FARR\_RAI\_MON can become a technical challenge. Another possible problem is that the process of calculating revenue needs to be performed in an orderly manner, meaning that processing RAIs needs to be executed prior to calculating results, so we get accurate calculated revenue. This is particularly relevant for companies that are often going through contract modification processes.

All of this means that processing RAIs manually might not be the most optimal solution. To perform RAI processing in an automated way, SAP delivers programs that might help in organizing the most streamlined processes of RAI transformation into contracts and POBs.

SAP delivers two transactions that can be run in the background to get contracts created/updated or any other items successfully processed: Transaction FARR\_RAI\_TRANS and Transaction FARR\_RAI\_PROC. After running these transactions sequentially, you'll use the monitor (Transaction FARR\_RAI\_MON) to complete processing and arrive at a contract.

Let's dig a little deeper. Transaction FARR\_RAI\_TRANS is used to transfer items from the **Raw** status to the **Processable** status. When the program is run, you can see the selections, as shown in Figure 4.11.

Selection data is very similar to the selection data in Transaction FARR\_RAI\_MON, and here you can limit **Selection Data** based on item class (**Rev. Acc. Itm Class**), **Sender Component**, and **Header ID**, which represent the exact items we need processed.

Figure 4.11 Transaction FARR\_RAI\_TRANS

If you have multiple item classes to be processed, it's important to know the order of items and how they are processed. Using sales and distribution as an example, the program transfers RAIs in the following order of source document types:

1. Order items without predecessor (SDOI)
2. Order items with predecessor (SDOI)
3. Fulfillment items (SDFI)
4. Planned invoice items (SDPI)
5. Invoice items (SDII)

Irrespective of which items are available, these five steps will be always executed.

Two more options (technical parameters) are important to understand. **Block Size For Mass Selection** determines the number of subareas that are used for processing. For example, an entry of **1,000** there means that all RAIs will be assigned to those 1,000 subareas. Which subarea an item is assigned to is determined based on the **Header ID** and RAI source document type (all RAIs with the same source document and **Header ID** will belong to the same subarea). For more information on subarea determination, see Section 4.1.2.

Next is the **Number of Intervals** that will run concurrently to process those RAIs. This number depends on the technical capabilities of the system; however, it should be clear that the locking mechanism is activated in that case—each item assigned to the same subarea is locked for processing.

In addition, there is also a setting named **Synchronous Call**, which causes the results of the running application to be written in the system log and can be accessed by Transaction SLG1. At the bottom of the screen, under **Settings for Application Log**, you have options for how detailed this log should be. If you don't select **Synchronous Call**, the batch monitor will be displayed after running the application.

When you're done filling in the key fields, click the **Execute** button or **Schedule** to schedule it as a job from the menu in the top-left corner.

Next, Transaction FARR\_RAI\_PROC moves items from the **Processable** status to the **Processed** status. Figure 4.12 shows how the transaction looks and which options are available.

Figure 4.12 Transaction FARR\_RAI\_PROC

As is visible from the screen, this transaction has the same options as Transaction FARR\_RAI\_TRANS. The main and most important difference is technical: while Transaction FARR\_RAI\_TRANS is used to transfer items from status 0 to 2 (**Raw** to **Processable**), Transaction FARR\_RAI\_PROC is processing items and moving them from status 2 to 4 (**Processable** to **Processed**).

So, you might be asking the following question: If you're not using **Raw** status, do you need to also schedule Transaction FARR\_RAI\_TRANS? The answer is yes. The reason

was mentioned at the beginning of this chapter: RAIs will be created if basic data checks are passed, so they will be created in **Raw** status as ready to be processed.

The process of sending data to create an RAR contract ends with Transaction FARR\_RAI\_MON, which, in this case, serves as an error correction tool: you can run it to verify how many items ended in error, what kind of errors are present, and how they can be resolved before new RAIs are created. So, to have a proper picture of the processed and remaining items in error, ideally, you should schedule Transaction FARR\_RAI\_TRANS before Transaction FARR\_RAI\_PROC and then run Transaction FARR\_RAI\_MON.

#### 4.1.2 Parallel Processing Framework

A key functionality to highlight is the use of the parallel processing framework (PPF) to create subareas, as we've touched upon in previous sections. The RAIs are grouped on specific areas and assigned the same KEYPP value, and then the packages are created. The packages are then distributed to jobs. The number of parallel jobs that can be created is again dependant on configuration and is determined by your Basis team. The creation of parallel jobs is sensitive as the packages must be grouped in a way that the locking is effective. The technical attributes of the KEYPP field are shown in Figure 4.13.

Figure 4.13 KEYPP Field Technical Description

The KEYPP field can have values from 000 to 999. In RAR, we generally must deal with huge volumes of data. The PPF has been designed to speed up the processing by dividing the data into packages, and the packages are passed on to parallel jobs or child jobs that run in parallel to save time. When defining the packages, you must consider the data locking and data grouping. To group the data into packages, the KEYPP field is used.

Lock object EFARR\_KEYPPBUKRS is used during the ENQUEUE process of locking the KEYPP fields for processing, as shown in Figure 4.14.



W	Lock parameter	Table	Field
<input checked="" type="checkbox"/>	MANDT	FARR_S_KEYPP_BUKRS_ENQ	MANDT
<input checked="" type="checkbox"/>	KEYPP	FARR_S_KEYPP_BUKRS_ENQ	KEYPP
<input checked="" type="checkbox"/>	BUKRS	FARR_S_KEYPP_BUKRS_ENQ	BUKRS
<input checked="" type="checkbox"/>	OBJPP	FARR_S_KEYPP_BUKRS_ENQ	OBJPP

Figure 4.14 KEYPP Lock Object

Most of the RAI mass processing activities can be done in parallel. The overall workload (the RAIs to be processed for the specified selection criteria) is split into intervals based on field KEYPP. There are 1,000 possible different KEYPPs (000–999). If you start the RAI processing for 100 intervals, 10 KEYPPs will be assigned per interval (10 x 100 = 1.000).

The parallel processing runs are currently supported for the following:

- RAI transfer (Transaction FARR\_RAI\_TRANS)
- RAI processing (Transaction FARR\_RAI\_PROC)

The usual number of intervals should be 3–5 times higher than the number of parallel jobs (e.g., 30–50 intervals for 10 parallel processes). The number of parallel processes may be restricted by Basis configuration.

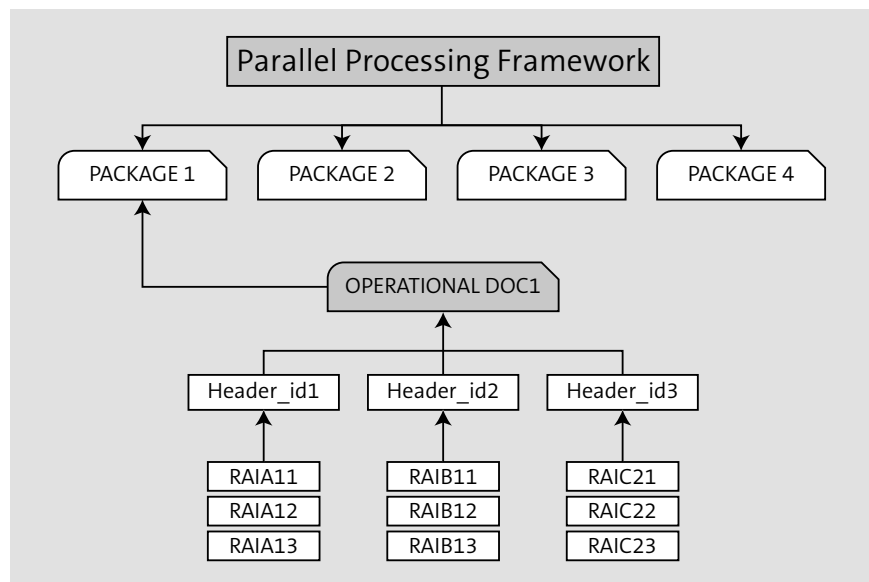


Figure 4.15 Parallel Processing Model

KEYPP will be assigned based on the same references, for example, all the header IDs belonging to a certain operational document or order will be assigned the same KEYPP, and then they will all be grouped in the same package for parallel processing. So, all the RAIs belonging to the same header ID and the header IDs belonging to the same operational document are packed in a single package, as shown in Figure 4.15. KEYPP is also heavily used in contract accounting and SAP Billing and Revenue Innovation Management.

### 4.1.3 Processing Order Items

Order items are the starting point for RAR contract creation or modification of existing contracts whenever there is an initial creation of RAR contracts or subsequent changes. Definition of the order item class is done in inbound processing (see Chapter 3, Section 3.2). Here, we'll focus on the instructions for processing order RAI classes.

In our example, we're creating one RAR contract based on a sales and distribution order. Once we populate the order with needed data (customer, condition items, duration, etc.), we save it and get its number, which we'll use in Transaction FARR\_RAI\_MON while searching for the order. It's important to mention that only complete documents will be transferred to RAR; if you get a message that a document is incomplete, but you can still save it, such a document won't be transferred.

After entering the sales order number in the **Header ID (Orders)** field and clicking **Execute**, we'll see the result shown in Figure 4.16.

Figure 4.16 Processing Order Items in Transaction FARR\_RAI\_MON

The first two columns are empty, which means that these RAIs are coming for the first time in Transaction FARR\_RAI\_MON so the contract and POB number aren't yet assigned. The **ItemStatus** column and the green square in the **Error** column tell us that the item is in **Processable** status and was created without any errors, meaning it's ready for processing. After columns of information about the data in the sales order, the date and time when the item was created are shown.

We also can see **SDPI** items in the source item type (**SrcItem**) column, which means that billing of this sales order is to be done on a regular basis, so the order has planned items representing these future billings.

Once we process the data by clicking the **Execute** button, we'll get information about what contract was created, as shown in Figure 4.17.

Date/Time/User	Number	External ID	Object text	Subject
12.02.2023 09:09:09 SM6918	19	SM69182023021209...	Revenue Accounting	Processing
■ Problem class Important	5			
■ Problem class Additional Information	14			

Ty...	Message Text
■	***** PROCESS *****
■	Start of processing of order item SDOI/90000802000010 for accounting principle IFRS
■	New performance obligation 103010 created for contract 200310

Figure 4.17 Contract Created in Transaction FARR\_RAI\_MON

Now, as the next step, we'll have our sales team make some changes. Let's assume they perform two types of changes:

1. They will change some text in the order.
2. They will change the standalone selling price (SSP).

In addition, it's important to mention that these two changes are to be performed at two separate moments in time, but no additional Transaction FARR\_RAI\_MON processing occurs in the meantime.

So, once we run Transaction FARR\_RAI\_MON, we'll see that two items are there, as shown in Figure 4.18.

POB(FR... Contract)	ItemStatus	Item Exist	Error	Ex-His	Ex	Send.Com.	SourceSys	SrcItemType	Source Item ID	Subarea	RevAcc...	Header ID	ItemID	Ref. Type	Reference ID	Customer	Partner	CoCode	Err.	Creation Date	Create Tm
103010	200310	☺	■			SD	SDOI	0090000802000010	0090000802000010	421	SD01	90000802	10	SDO	0090000802	1000884500	GB04			12.02.2023	10:18:43
103010	200310	☺	■			SD	SDOI	0090000802000010	0090000802000010	421	SD01	90000802	10	SDO	0090000802	1000884500	GB04			12.02.2023	10:20:19
103010	200310	☺	■			SDPI	SDPI	0090000802000010000000018520000...	0090000802000010000000018520000...	421	SD03	90000802	10			1000884500	GB04			12.02.2023	10:18:43

Figure 4.18 Changes in FARR\_RAI\_MON

Now, the contract and POB number are populated in the first two columns, which tells us that these RAIs represent a contract modification. It's possible to click on the **Contract** number to go straight to the contract in the respective SAP Fiori app or SAP Business Client program.

To verify that the SSP change was transferred, we can select the **Condition Item** tab, as shown in Figure 4.19.

Here, we see four conditions: each change produced two condition entries. One condition represents a pricing condition (**ZPRO**) with a **P/L Account** assignment. This condition is marked as the main condition type (**Main Cond.**). The SSP is a statistical condition type (**ZSSP**), so the **Main Cond.** checkbox remains empty.

ItemStatus	Error	Send.Com.	SrcItemType	Source Item ID	Cond.type	P/L Account	Timestamp	TC Amount	Currency	Stati...	Condi...	Main Cond.
☺		SD	SDOI	0090000802000010	ZPRO	5111000003	20.230.212.091.843	90.000,00	GBP	X	P	X
☺		SD	SDOI	0090000802000010	ZSSP		20.230.212.091.843	90.000,00	GBP	X	P	
☺		SD	SDOI	0090000802000010	ZPRO	5111000003	20.230.212.092.019	90.000,00	GBP	X	P	X
☺		SD	SDOI	0090000802000010	ZSSP		20.230.212.092.019	96.000,00	GBP	X	P	

Figure 4.19 Transaction FARR\_RAI\_MON: Conditions

By looking at these items, we see that an item with a later timestamp is an SSP with the amount 96,000.00, which means this is the item entered as last (highest timestamp).

Now we go with processing RAIs. The important thing is in the last two columns where these two items are created with different timestamps. The question is, do we need to only select the item we want to process? The answer is no because the system will automatically select only the last item and delete the rest.

There's one important point to mention regarding conditions. In order items, the order always passes at least two items: one is the main item, which represents the transactional price, and the other one is the statistical item, which represents the SSP. There can be specific situations, but this one is the most basic one. If you're considering integration with sales and distribution, a pricing procedure is in place that determines the pricing condition to be translated as the transaction price. Figure 4.20 shows what the pricing procedure assigned to a sales document looks like. All condition types appear on the lefthand side with their assigned values. On the right side, the condition value is price multiplied per quantity, which represents the value that will be passed to RAR.

I...	CnTy	Name	Amount	Crcy	per	UoM	Condition Value	Curr.	Status
☺	ZPRO	PG: List Price.		1,00 GBP		1PC		1,00 GBP	
☺	ZKUM	Cumulation cond-Stat		1,00 GBP				1,00 GBP	
		Gross Value		1,00 GBP		1PC		1,00 GBP	
		Net Value of Item		1,00 GBP		1PC		1,00 GBP	
☺	ZSSP			1,00 GBP		1PC		1,00 GBP	
		Net Value 2		1,00 GBP		1PC		1,00 GBP	
☺	ZTXD	Tax Service - Doc Lv		0,000 %				0,00 GBP	
☺	ZTXE	TaxSvc call - doc lv		0,000 %				0,00 GBP	
☺	ZWST	Output Tax		0,000 %				0,00 GBP	
		Total		1,00 GBP		1PC		1,00 GBP	

Figure 4.20 Pricing Procedure

Here, the condition type (**CnTy**) represents price **ZPRO**. Once we save document, it will create RAIs that are ready for processing, as shown in Figure 4.21.

Once items get processed, an RAR contract is created with all the corresponding POBs.

ItemStatus	Hist Exist	Error	Send.Com...	SourceSys.	SrcItemTy	Source Item ID	Cond.type	P/L Account
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		SD		SDOI	0090000847000010	ZPRO	5111000003
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		SD		SDOI	0090000847000010	ZSSP	

Figure 4.21 Condition Items in Transaction FARR\_RAI\_MON

Now, let's assume that during invoicing, there is a change in the condition type. This isn't uncommon because often companies give some discounts at the moment of invoicing. Figure 4.22 shows that the condition type changed between the order and the invoice (instead of ZPRO, we get ZPRG). This situation more often occurs in the case of credit/debit memos when users want to change the value of the debit memo, and there is a specific condition type only for manual pricing.

Chly	Name	Amount	Crcy	per	UoM	Condition Value	Curr.	Status	NumCCo
<input checked="" type="checkbox"/>	ZPRG	PG:Global List Price	15,00	GBP	1PC	15,00	GBP		1
<input checked="" type="checkbox"/>	ZKUM	Cumulation cond-Stat	12,00	GBP		7,20	GBP		0
		Gross Value	15,00	GBP	1PC	15,00	GBP		1
		Net Value of Item	15,00	GBP	1PC	15,00	GBP		1
<input checked="" type="checkbox"/>	ZSSP	Net Value 2	12,00	GBP	1PC	7,20	GBP		1
		Net Value 2	7,20	GBP	1PC	7,20	GBP		1
<input checked="" type="checkbox"/>	ZTXD	Tax Service - Doc Lv	0,000	%		0,00	GBP		0
<input checked="" type="checkbox"/>	ZTXE	TaxSvc call - doc lv	0,000	%		0,00	GBP		0
<input checked="" type="checkbox"/>	ZWST	Output Tax	0,000	%		0,00	GBP		0
		Total	15,00	GBP	1PC	15,00	GBP		1
		Standard - USA /With	15,00	GBP	1PC	15,00	GBP		1

Figure 4.22 Changed Condition Type during Invoicing

Because there is a standard integration between sales and distribution and RAR, you'll get an RAI ready for processing where now we see condition type ZPRG instead of ZPRO. However, RAR won't allow such an item to be processed due to the difference in the main condition type in the order (defined while creating the contract) and the invoice items (defined during invoicing).

#### 4.1.4 Processing RAIs with a Predecessor

In terms of process chain dependency, all RAIs can be divided into those created with a predecessor and those created without a predecessor. In the previous section, we described processing RAIs without predecessors, which are used to create or change

RAR contracts. When it comes to RAIs with predecessors, there are multiple RAI types that are created depending on the fulfillment type of an RAI, with the exception of planned items (created as a result of a billing plan) and invoice items (created as a result of invoicing). In other words, fulfillment items have several different types depending on the type of fulfillment expected, while invoice items are always the result of invoicing only.

When you need to process an RAI item with predecessor, the best method is to select the option in Transaction FARR\_RAI\_MON for processing **All Items Related to Order Items**, as shown in Figure 4.23. By that, you're ensuring that the system will pick up all items related to an order irrespective of whether they are fulfillment or invoice items.

Execute	Get variant...	Program Documentation	Further Selections	Personalize	More
Kind of Selection					
Kind of Selection: <input type="text" value="All Items Related to Order Items"/>					
Item Status					
Item Status: <input type="text" value="03 Raw and Processable Items"/>					
Source Documents					
Component:	<input type="text"/>	to:	<input type="text"/>	<input type="button" value="→"/>	
Logical System:	<input type="text"/>	to:	<input type="text"/>	<input type="button" value="→"/>	
Type:	<input type="text"/>	to:	<input type="text"/>	<input type="button" value="→"/>	
Header ID (Orders):	<input type="text" value="41007138"/>	to:	<input type="text"/>	<input type="button" value="→"/>	
Item ID (Orders):	<input type="text"/>	to:	<input type="text"/>	<input type="button" value="→"/>	
Date:	<input type="text"/>	to:	<input type="text"/>	<input type="button" value="→"/>	
Time:	<input type="text" value="00:00:00"/>	to:	<input type="text" value="00:00:00"/>	<input type="button" value="→"/>	
Master Data					

Figure 4.23 Transaction FARR\_RAI\_MON with Order Number

For example, if you just enter an order number (41007138), in the result table, the system will display all items related to this order number, as shown in Figure 4.24. Here, you can see, based on the source item type (SrcItemTy), that it picked up all fulfillments and invoices created based on this order that can be processed.

Note that in the **Original Item ID** field (not shown), the system stores the order item, which is the main source (predecessor) for the follow-up item.

A few things should be kept in mind when processing follow-up items related to the original item. Sometimes, the follow-up item is created before the original item is processed. In this case, the follow-up items won't be processed before the original item is processed without an error. In such cases, you'll receive an error, as shown in Figure 4.25.

ItemStatus	Hist	Exist	Error	Ex.His.Ex	Send.Com.	SourceS...	SrcItmTy...	Source Item ID	Subarea	RevAcc...	Header ID	ItemID	Ref. Ty...
SD						SDFI		0070000118001190 20221124145422	012	SD02	70000118	1190	
SD						SDFI		0070000118001200 20221124145422	012	SD02	70000118	1200	
SD						SDFI		0070000118001210 20221124145422	012	SD02	70000118	1210	
SD						SDFI		0070000118001220 20221124145422	012	SD02	70000118	1220	
SD						SDII		9110000084000002	962	SD03	9110000084	2	
SD						SDII		9110000084000003	962	SD03	9110000084	3	
SD						SDII		9110000084000004	962	SD03	9110000084	4	
SD						SDII		6110000011001190	012	SD03	6110000011	1190	
SD						SDII		6110000011001200	012	SD03	6110000011	1200	
SD						SDII		6110000011001210	012	SD03	6110000011	1210	
SD						SDII		6110000011001220	012	SD03	6110000011	1220	
SD						SDII		9110000086001190	012	SD03	9110000086	1190	
SD						SDII		9110000086001200	012	SD03	9110000086	1200	
SD						SDII		9110000086001210	012	SD03	9110000086	1210	
SD						SDII		9110000086001220	012	SD03	9110000086	1220	

Figure 4.24 Results for Items with Predecessors

Start of processing of invoice item SDII/9110000286000001 for accounting principle IFRS
Performance obligation to be invoiced could not be determined (SDII/9110000286000001)
Start of processing of invoice item SDII/9110000285000001 for accounting principle IFRS
Performance obligation to be invoiced could not be determined (SDII/9110000285000001)
Start of processing of invoice item SDII/6110000022000020 for accounting principle IFRS
Start of processing of invoice item SDII/9110000600000004 for accounting principle IFRS
Performance obligation to be invoiced could not be determined (SDII/9110000600000004)
Start of processing of invoice item SDII/9110000600000003 for accounting principle IFRS
Performance obligation to be invoiced could not be determined (SDII/9110000600000003)

Figure 4.25 Error for Invoice Processing and POB Determination

In this case, the system is notifying you that the predecessor item for the mentioned invoice couldn't be found. There can be different reasons for this, but all are related to process design and its execution. In addition, credit/debit memos without reference can't be processed.

The second issue that often arises is that a change was made to the order and RAIs were created before the invoice was processed. This means the order item to which the invoice relates already exists as unprocessed in Transaction FARR\_RAI\_MON with an older timestamp than the one with invoice. This potentially could cause an inconsistency because the RAR engine isn't aware of the type of change in the order item. To prevent this, the ARL won't allow such an RAI to be processed. It will produce an error, as shown in Figure 4.26.

***** PROCESS *****
Invoice RAI SDII/9110000240001020, there is unprocessed previous RAI to same origdoc SDOI/41007251001020.
Invoice RAI SDII/9110000282001010, there is unprocessed previous RAI to same origdoc SDOI/41007273001010.
Start of processing of invoice item SDII/9110000539000004 for accounting principle IFRS

Figure 4.26 Invoice Error Due to an Unprocessed Order RAI

In addition, RAR doesn't support the change of condition types between invoice and order items. Often users use different condition types in a credit memo (which is also an invoice item) than they do in an order. RAR won't let such items be processed and will throw an error about condition types inconsistency between the order and invoice item. These kinds of situations should be resolved during process design.

## 4.2 Managing Revenue Accounting Items

Along with processing RAIs, there is often a need to do some other activities with RAIs before, during, or after processing. These activities can be split into either changing items after they are created in the ARL or removing them from the list for processing.

Despite RAR being a tool with high automation (meaning that data usually isn't created in RAR itself, but rather is sent from external applications), sometimes there is a need to manipulate the data before it's sent for processing. These options are enabled to ensure that once data reaches RAR, it's of the highest possible quality. But these options come with offsets as well, and you need to understand what effect using these options can have on the overall system and data integrity. In the following sections, we'll explain in detail the changing and exempting of data options that are to be processed in RAR.

### 4.2.1 Changing Items

You should now have a basic understanding of how RAR is structured: contracts and POBs are always created based on data that is sourced in some other application, which can be either SAP or non-SAP based. Before data can be stored in RAR, it needs to pass staging in the ARL.

So, what happens if some data needs a change or correction? In most cases, you need to fix the data in the source system where it appeared. There are two reasons for that:

- The aim always should be to maintain consistency between systems. In some cases, this requirement is strongly recommended (e.g., integration with sales and distribution where there are reports available to detect these kinds of issues), but in some others, it's not that strong.
- Data correction can be a tedious task. You may not always know what can and should be changed to bypass some problem and why.

However, the reality is that not all data has the same importance (e.g., changing the SSP of an item isn't the same as changing a description field), and it's not always straightforward and possible to correct data. For example, you could import data in RAR from an external CRM system where two systems have different timelines for closing. So, when you notice that some correction needs to be made, it's no longer possible to create a correction in the external system.

All of this led to the ability to manually perform changes in RAIs. To activate such an option, you need to customize which RAI fields are changeable. You can access this functionality by following menu path **Revenue Accounting • Inbound Processing • Revenue Accounting Items • Define Modifiable Fields for Revenue Accounting Items**.

You first need to define which fields aren't available for change. This information can be found in structure FARR\_S\_RAI2\_MI\_FIX for processable main items via Transaction SE11. Similarly, you can see which items aren't available for changing for condition items or items in **Raw** status. Now, once you know that the field can be changed, you can access this activity to get to the screen shown in Figure 4.27.

Maintain Changeable Fields of Rev. Accounting Item Classes				
RevAccCl	Rec. Type	Status	Field Name	FieldAttr
<input type="checkbox"/> SD01	All Record Types	▼ All Statuses	▼ QUANTITY	
	All Record Types	▼ All Statuses		
	All Record Types	▼ All Statuses		
	All Record Types	▼ All Statuses		
	All Record Types	▼ All Statuses		
	All Record Types	▼ All Statuses		
	All Record Types	▼ All Statuses		
	All Record Types	▼ All Statuses		
	All Record Types	▼ All Statuses		
	All Record Types	▼ All Statuses		
	All Record Types	▼ All Statuses		
	All Record Types	▼ All Statuses		
	All Record Types	▼ All Statuses		
	All Record Types	▼ All Statuses		
	All Record Types	▼ All Statuses		
	All Record Types	▼ All Statuses		
	All Record Types	▼ All Statuses		
	All Record Types	▼ All Statuses		
	All Record Types	▼ All Statuses		
	All Record Types	▼ All Statuses		

Figure 4.27 Changeable Fields for RAIs

The first thing you need to define is to which RAI class this change will be used in the **RevAccCl** field: you can define fields that will be changed on order items, fulfillment items, or invoice items. Then, the next step is about record types (**Rec. Type** field) and whether you want to change data in main items, condition items, or both (**All Record Types**). After you define which **Status** the data can be in to be changed (**Raw**, **Processable**, or **All Statuses**), you define which field needs to be changed in the **Field Name** column. In the last column, **FieldAttr**, you need to define whether the field will be visible to the user or only displayed without the ability to change. In this example, the **QUANTITY** field was made changeable during processing in **All Statuses**.

Once setup is complete, when running Transaction FARR\_RAI\_MON, you'll be able to use the **Change** button, as shown in Figure 4.28.

By selecting **Change** and specifying an item, a separate screen will appear in which you can change data that was previously customized as changeable.

The customization for changing items is completed. However, the recommendation is to avoid making manual changes to RAIs. The best way to keep the system consistent is to correct data in the source system in which it was created in the first place. However, if you feel that manual intervention over RAIs is unavoidable, extra caution should be paid when selecting the data that can be changed.

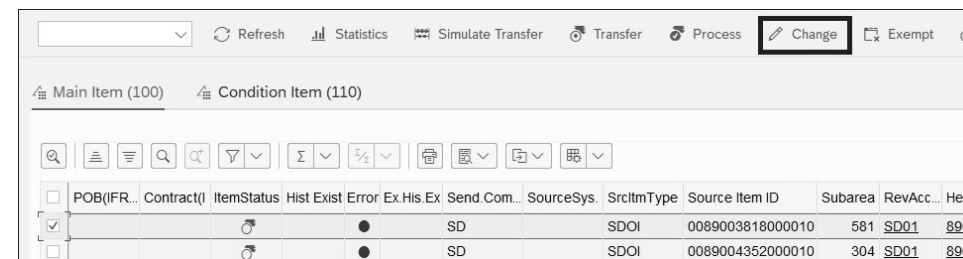


Figure 4.28 Change Option in Transaction FARR\_RAI\_MON

### 4.2.2 Exempting Items

Exemption of items is a process when, for some reason, an item needs to be excluded from processing in Transaction FARR\_RAI\_MON. Usually this refers to items being corrupted in terms of data quality, which can't be fixed in the source system. If not exempted, these items stay in Transaction FARR\_RAI\_MON (and in the processable table), consuming table space and making item processing items.

In addition, by leaving corrupted items in the processable table, you can keep complete contracts from processing. For example, if you have item 20 from a sales order with an error, the whole order won't be processed unless the item is exempted. For classic inbound processing, the table structure for exempted items looks like Figure 4.29.

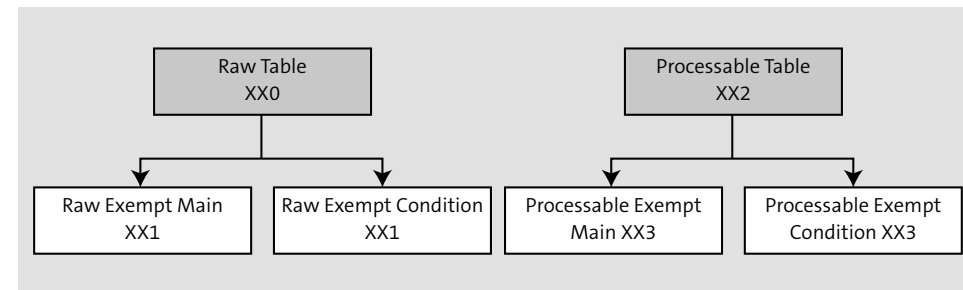


Figure 4.29 Exempted Items Table Structure

Exempted items are kept in separate tables both for main and condition items. In other words, the table structure for exempted RAIs mimics the table structure for raw and processable items. The naming convention is that tables for exempted items have a 1 or 3 after the name of the RAI class. For example, for integration with sales and distribution, the name of the RAI class for order items is SD01, so tables generated for them are as follows:

- Table /1RA/SD012MI for main items
- Table /1RA/SD012C0 for main items

For these RAI classes, the corresponding tables for exempted items are generated:

- Table /1RA/SD013MI for exempted main items
- Table /1RA/SD013CO for exempted condition items

If you're using raw data in processing and have three RAI classes (order, fulfillment, and invoice items), then in total you'll have 12 tables related to exempted items: two tables for main and condition items, multiplied by one set for raw and processable, and multiplied by the number of RAI classes.

Tables for exempted items are almost the same as those for processable or raw items. The difference is in the fields giving information about when the exemption occurred and by whom. In the **Exemption History** shown in Figure 4.30, it's clear which user performed the exemption and when. This information is useful for audit requirements.

Exemption History			EXCHIST
Exemption Date (UTC)	15.02.2023	20230215	EXCDATE
Time of Exemption (UTC)	12:36:41	123641	EXCTIME
Exemption Reason	EX	EX	EXCREASON
User for Exemption	SM6918	SM6918	EXCUSNAM

Figure 4.30 Exemption Items Fields

For optimized inbound processing (OIP), the table structure is significantly changed. Instead of dynamic generation of tables and APIs, in OIP, the system works with static, predefined tables. In addition, the process of exemption is replaced by postponing.

### Caution

A shift in tables structures is one of the reasons customers need to be extra careful before deciding which option to adopt when either implementing RAR for the first time or deciding about upgrading to a newer version of RAR.

Before the process of exemption can be used, you need to perform certain customizing tasks in RAI management while setting up inbound processing. Follow menu path **FARR\_IMG • Inbound Processing • Revenue Accounting Item Management**. When this area for customizing appears, fill in the following fields (see Figure 4.31):

- **ExempRsn**  
First, you need to define the exemption reason with this two-digit code. If there is a specific logic for exempting items, you might have multiple reasons; otherwise, one is enough.
- **Exemption Reason**  
You can enter text explaining the reason for exemption.
- **StatusGrp**  
You can define for which items you're creating exempted items: just for processable, just for raw, or for all.

### ExempType

Exemption type tells the system whether there will be an option to restore an item that was exempted once. This again depends on the process. If the reason for exemption is data issues, you can either fix the data in the source system or recreate it, in which case, it might be a good option not to restore the item that was exempted. Alternatively, the process might be that data will be corrected while in the exempt table. In that case, defining a restore reason is a good option.

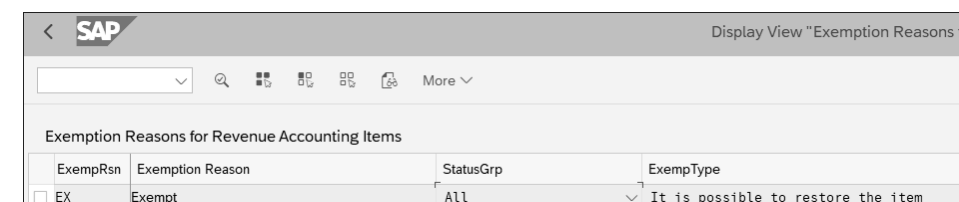


Figure 4.31 Defining the Exemption Reason

The option to define restoration reasons can be found just below the menu path for exemption (**Define Restoration Reasons for RAI Items**), and setup is the same as for exempt reasons except that **ExempType** doesn't need to be defined.

Once you set an exemption reason, it's ready to be used. Run Transaction **FARR\_RAI\_MON**, and enter the header ID that needs to be processed. In the list shown in Figure 4.32, you can see that the **Exempt** button is available.

Figure 4.32 Exemption in Transaction FARR\_RAI\_MON

Click the **Exempt** button to see exemption reasons in the **Exempt Processable Item** popup, as shown in Figure 4.33. Select your exemption reason (**EX Exempt**, in our example), and click the green checkmark.

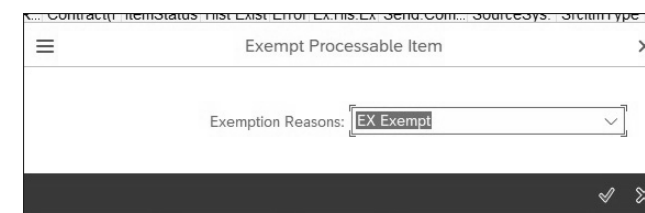


Figure 4.33 Selecting the Exempt Reason

In Transaction FARR\_RAI\_MON, you can see which items are exempt by choosing **07 Exempted Items** in the **Item Status** dropdown, as shown in Figure 4.34.

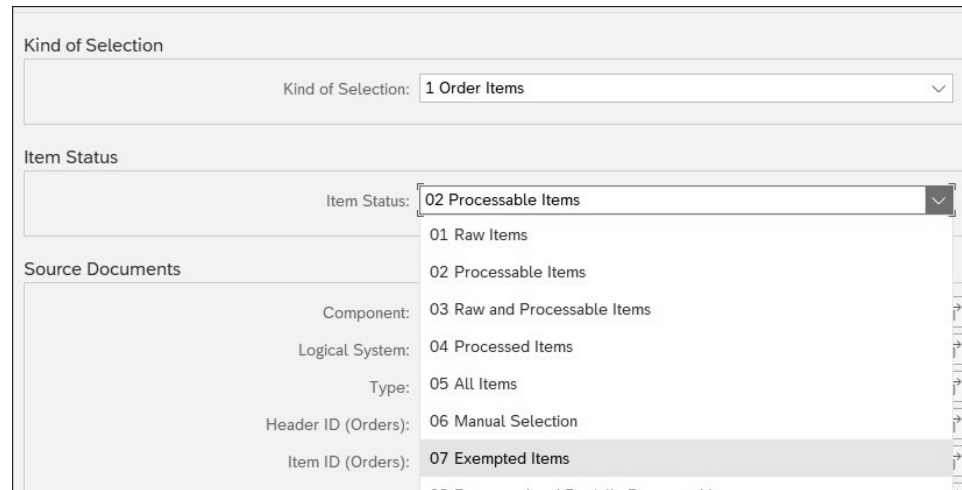


Figure 4.34 Exempted Items in Transaction FARR\_RAI\_MON

In the list shown in Figure 4.35, you can see all the items being exempted. In addition, the option for restoring an item is available (the **Restore** button), which can be used to transfer items back to the processable table.

POB(FRS)	Contract	ItemStatus	Hist Exist	Error	Ex.His.Ex	Send.Comp	SourceSys	SrcItem	Type	Source Item ID	Subarea	RevAccCl	Header ID	ItemID	ExempDate	Reason	ExempUse
		SD					SDOI	0089004364000010	341	SD01	89004364	10	18.02.2023	EX	SM6918		
		SD					SDOI	0089004364000010	341	SD01	89004364	10	18.02.2023	EX	SM6918		
		SD					SDOI	0089004364000010	341	SD01	89004364	10	18.02.2023	EX	SM6918		
		SD					SDOI	0089004364000010	341	SD01	89004364	10	18.02.2023	EX	SM6918		
104007	200407	SD					SDOI	0090000820000010	009	SD01	90000820	10	15.02.2023	EX	SM6918		
104007	200407	SD					SDOI	0090000820000010	009	SD01	90000820	10	15.02.2023	EX	SM6918		
104007	200407	SD					SDOI	0090000820000010	009	SD01	90000820	10	15.02.2023	EX	SM6918		
104012	200412	SD					SDOI	0090000825000010	563	SD01	90000825	10	15.02.2023	EX	SM6918		

Figure 4.35 Exempted Items List in Transaction FARR\_RAI\_MON

### 4.3 Extending Transaction FARR\_RAI\_MON

The RAI monitor is the main tool used for data transfer between the ARL and RAR database tables. The standard program contains most of the options to search and filter items that need to be processed or analyzed further. However, you can extend some of the standard options without the need for additional programming.

In the upper part of the screen in Transaction FARR\_RAI\_MON, you'll find the option to further enhance selection parameters (**Further Selections**), as shown in Figure 4.36.

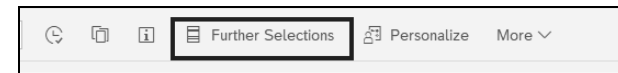


Figure 4.36 Further Selections in Transaction FARR\_RAI\_MON

By selecting this option, you can see all the fields that are part of structures assigned to standard RAI structures, as shown in Figure 4.37. All fields here can be used as selection parameters when you want to see only data that is relevant for processing. On the left-hand side are all the fields in the structure for the main items. In our example, we've selected **Channel partner**, as you can see under the **Dynamic selections** screen on the right-hand side.

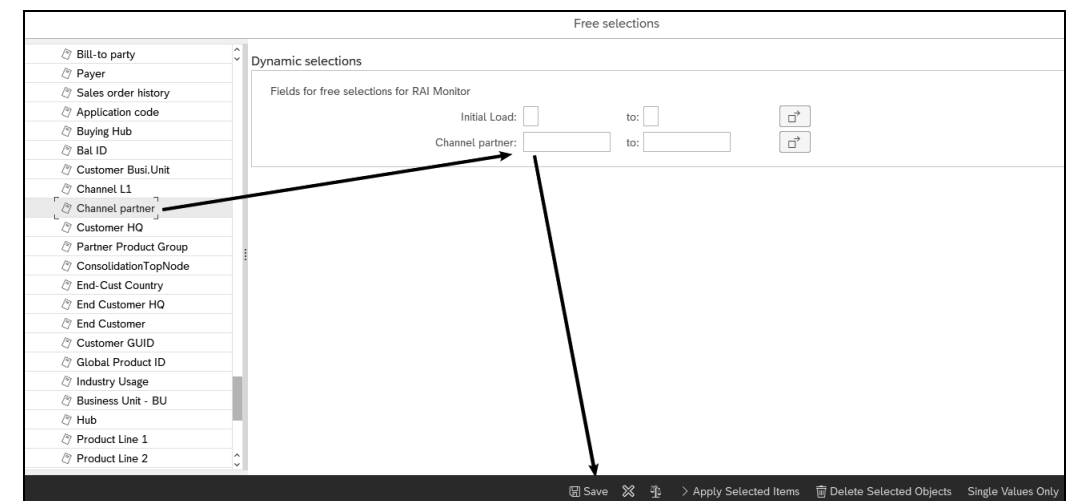


Figure 4.37 Data Selection for Further Options

Once additional data has been selected, click the **Save** button, and the **Further Selections Exist** checkbox will be selected in the **Further Attributes** section (refer to Figure 4.3).

#### Use Case for Additional Selection

It's not uncommon that one RAR contract is created as a combination of different documents that also can have different sources. For example, one RAR contract is created as a combination of sales contract, sales order, and service order. Here, to make the necessary links between them, we often use separate, custom fields. So, to ensure that in one shot all documents are being taken into consideration, a good idea is to extend the search in Transaction FARR\_RAI\_MON with that special custom field.

## 4.4 Error Resolution While Creating Revenue Accounting Items

As mentioned in previous chapters, data quality is one of the prerequisites for RAR to perform. When thinking about how to ensure that data is able to be processed by RAR, it's worth mentioning that RAR is much more integrated with external systems than plain finance. In other words, the old paradigm that operations and finance need to integrate at the moment of account determination isn't the case with RAR.

Due to the nature of IFRS 15, RAR represents processes in sales operations that are copied in finance. So that is why implementation of RAR requires a whole different level of understanding for finance consultants about SAP Billing and Revenue Innovation Management and sales and distribution. The same applies to sales and distribution consultants: it's not enough just to have a high-level understanding of accounting and controlling (if any). Now, people involved in the sales process need to understand from the very beginning the financial implications of decisions made very early in the process.

All this being said, if these recommendations are followed, the data coming to RAR as RAIs should be of sufficient quality. However, it's safe to assume that even with the most rigid rules, some errors will appear in the RAI monitor requiring both *proactive* and *reactive* measures. We'll discuss both proactive and reactive measures next, and then explain a few error handling techniques.

### 4.4.1 Proactive and Reactive Measures

Proactive measures are checks put in place so the item will be verified before it comes to RAR. Figure 4.38 represents one way of preventing errors from happening.

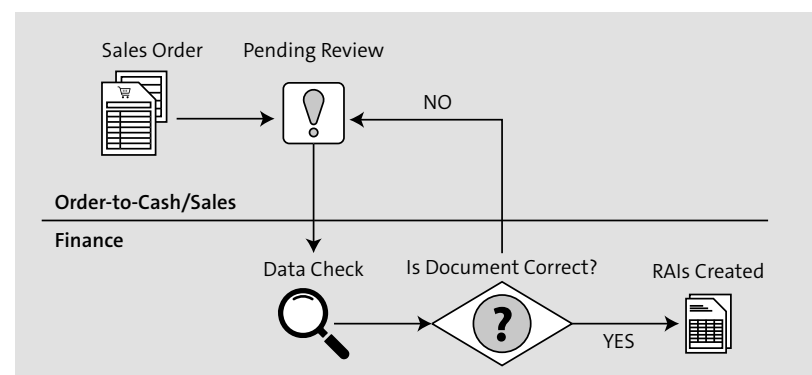


Figure 4.38 Statuses While Processing RAIs

The process starts in the sales department by creating the sales document. This document can be a sales order, contract, or even service order. Once data has been saved, it will have an assigned status that review is needed (**Pending Review** status), which is the finance team's responsibility. The finance team will check data that is needed for RAI creation such as the following:

- SSP existence/correctness
- Contract duration
- Item type correctness
- Some custom rules that companies might have implemented in their processes

As a result of the finance team check, the document may be sent back to the sales team to complete it/correct it if errors are found. The other result may be that the finance team determines that document is in proper quality and decides to set the status to **Approved**, which would trigger creation of RAIs.

Of course, this process can be extended in many ways. For example, a company might opt for making some kind of workflow: there might be more statuses representing readiness for processing (document might be initially entered and not yet be processed until created), there might be routing between different departments based on statuses, and so on. But the main point remains—let only correct data come to RAR.

The main benefit of such a process is that the data coming as an RAI is the quality needed for successful processing and contract creation. This part can't be emphasized enough—as mentioned, the logic on which revenue recognition works according to IFRS 15 is completely different than before. Here, revenue recognition happens based on triggers, and sometimes these triggers are automatic and something you have no influence on (e.g., time-based POBs). So, error fixing is much harder or even impossible if an issue is noticed after the error recognition process has already kicked in. So, the main goal of the setting statuses process is preventing such issues from happening.

There are some drawbacks to the process. In Figure 4.38, you can see that the finance team is expected to be familiar with applications that aren't strictly from the finance domain, such as checking the sales document and change log, document flow, and so on. The second point isn't a limitation as such, but needs consideration: Such a process is very suitable for organizations that have a revenue recognition process as part of shared service processes. In that case, it would be enough to train one group of people who could perform such controls and checks. Usually, IFRS 15 is a global policy for reporting, so making the process organized in the form of a shared service center would make sense.

Now, let's discuss reactive measures. Not all errors can be caught before they reach the ARL: Even in environments where most of errors would be caught before they reach the ARL, some will come through. When they come to the table for transfer or processing (depending on whether raw items are included or not), those items will appear in Transaction FARR\_RAI\_MON for processing. When you open the transaction, all items with an error will have a red dot in the **Error** column, as show in Figure 4.39.

To see what the error type is, click an item and see what message is. As shown in Figure 4.40, the system will display all information needed for you to recognize the root cause and correct the problem.



ItemStatus	Hist	Exist	Error	Ex	Hls	Ex	Send Com...	SourceSys	ScrItmType	Source Item ID	Subarea	RevAcc.	Header ID	ItemID	Ref. Type	Reference ID	Customer	Partner	CoCo.	Err.	Creation Date	Create Tim
								SD	SDOI	0800760259000010	526	SD01	800760259	10	SDO	0800760259	1200000344		NL03	3	16.02.2023	22:21:30
								SD	SDOI	0800760276000010	436	SD01	800760276	10	SDO	0800760276	1000055298		ID02	3	22.02.2023	10:14:57
								SD	SDOI	0800760276000020	436	SD01	800760276	20	SDO	0800760276	1000055298		ID02	3	22.02.2023	07:27:54
								SD	SDOI	0800760293000010	617	SD01	800760293	10	SDO	0800760293	1930000001		AU02	3	25.02.2023	10:11:36
								SD	SDOI	0800760296000010	930	SD01	800760296	10	SDO	0800760296	1200000383		NL03	3	22.02.2023	09:31:11
								SD	SDOI	0800760296000020	930	SD01	800760296	20	SDO	0800760296	1200000383		NL03	3	22.02.2023	09:31:11
								SD	SDOI	0800760297000010	271	SD01	800760297	10	SDO	0800760297	1200000383		NL03	3	22.02.2023	09:49:06
								SD	SDOI	0800760297000020	271	SD01	800760297	20	SDO	0800760297	1200000383		NL03	3	22.02.2023	09:49:06

Figure 4.39 Errors in Transaction FARR\_RAI\_MON

Ty...	Message Text	LText
	Start of processing of order item SDOI/800760259000010 for accounting principle IFRS	
	Inflight check: Error C01/Contract 102008 found before save to database	
	C01: Allocation error found for contract 102008	?
	Start date of time-based performance obligation \$000000000000001 is empty	?
	Either a duration or an end date must be specified for perf. obligat. \$000000000000001	

Figure 4.40 Details of Error in Transaction FARR\_RAI\_MON

You can see what errors were thrown before data could be saved to the database. The first error you can see is that SAP has thrown an inflight check error. These checks are proactive checks that SAP makes before data is saved in tables in order not to cause inconsistencies. In other words, inflight check errors will appear so that data inconsistency doesn't appear at all. Data validation checks will appear before data is written permanently to database. Both of these checks will be covered in detail next.

In total, there are 26 types of errors that can appear, and all of them are a threat to data integrity. All errors are split into two categories:

- C errors represent inflight checks.
- E errors represent data consistency checks that are performed before data is written in tables.

### C and E Errors

The list of errors that can be issued isn't final and can be updated by SAP to include more of them. This means that more issues are being discovered and classified, and this issue list can grow with time.

#### 4.4.2 Inflight Errors

*Inflight errors* are introduced as standard functionality in RAR to give more information about the root cause of erroneous items that might appear as a result of processing. The inflight check functionality is shown in Figure 4.41.

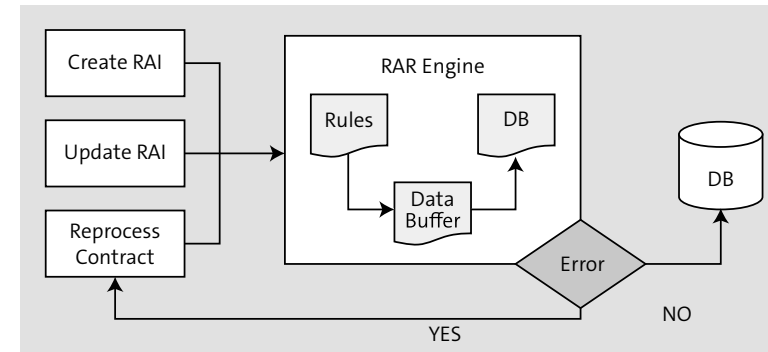


Figure 4.41 Inflight Error Logic

As mentioned, this error check sits in the RAR engine itself and is triggered each time there is some change coming from any user activity, which is either creating new RAIs, updating existing data, or reprocessing contracts, which is an activity that runs directly on a contract when it's already created. When such an activity is performed, data is kept in a buffer and not saved to the database yet, and all checks are performed over that buffer. The database commit is executed only after all errors are cleared.

Not all activities will trigger inflight error checks, however. If invoice items are processed that don't lead to price reallocation or running ABC programs, these activities won't trigger inflight error checks. This was done on purpose because the idea of error checking is to avoid data inconsistency without stopping business-related activities such as the month-end closing process.

Inflight error checks can be further extended. SAP created a BAdI called Extended Checks before Saving to Database where you can opt to implement your own logic by using subclass CL\_FARR\_DATA\_EXTENDED\_CHECK.

Once an inflight error occurs, in most situations, the best option is to fix the problem where it occurred (in the source system) and trigger recreation of RAIs again. Another option is to fix the data in the RAI table if proper customization is made. However, this option should be used only as a last resort because it would mean there is a difference between data in the source system and RAR.

But let's look at our example error shown previously in Figure 4.40. In the first line, the system reported its C01 error for a contract that already exists. So, the RAI is modifying the existing contract, which is triggering reallocation.

In short, the C01 error means that if a change coming from an RAI sent by the source system is saved in the database, it will create an inconsistency in the balance of the allocation effect. For all POBs in a contract, the total allocation effect must be zero. For example, a calculation that would throw an error as described previously can be represented as shown in Table 4.1.

POB_ID	Transactional Price	Allocated Price	Allocation Effect
POB A	1,000.00	1,200.00	+200
POB B	2,000.00	1,900.00	-100
<b>Total</b>	<b>3,000.00</b>	<b>3,100.00</b>	<b>+100</b>

Table 4.1 Allocation Effect

In our case, we have a new RAI that can't be used to create a POB, but if the transaction price were created, it would create an inconsistency with the allocated price.

In other words, looking simply at the CO1 error, it won't always give you a clear place where the issue occurred and what to do to solve it. It will show more about what kind of problem could be caused if the error isn't solved.

In the third line shown previously in Figure 4.40, you can see all the necessary information about where and why the error actually occurred. The system is saying that the start date of item \$00000000001 is missing. First, now you're clear what caused an error: the POB affected is actually a time-based POB missing its start date, which is preventing the POB from being created. The second thing you can see from the message is that the system is trying to create a new POB. So, if you're integrating with sales and distribution in this case, you can look at a new sales document item, which is newly added in the sales document where the start/end date is missing and resolve it there. Once that is done, a new POB with a later timestamp is created that you can process, and this error would be resolved.

#### 4.4.3 Data Validation Checks

Unlike the inflight errors functionality, data validation checks are issued after the database update is executed to validate data once the data reaches tables in RAR. There are currently 21 error categories that can be issued once data is validated, and, unlike inflight errors, data validation checks start with prefix E.

The purpose of using data validation checks is different from inflight errors. We mentioned that inflight error checks don't prevent the business part of contract management, even if errors are detected. So even if some C errors are raised, contracts will still be posted and revenue will be recognized if any. The purpose of E errors, on the other hand, is to protect you from submitting potentially erroneous results to revenue reporting. Therefore, contracts with E errors won't be processed further.

The root cause of errors coming from data validation checks can vary from errors coming from ARL data processing, wrong BRFPplus rules, or even mistakes in different BADl implementations. However, there are situations where data validation checks can't be of help. These are different errors caused by user actions (e.g., wrong revenue suspension), no end-to-end reconciliation (between sender component and contracts), compounding POBs issues, and so on.

Data validation checks are performed in two steps:

#### 1. Consistency check

This activity is performed by running Transaction FARR\_CONTR\_CHECK, where the system runs all 21 checks and writes results to table FARR\_D\_CONS.

#### 2. Consistency monitor

This activity reads entries in table FARR\_D\_CONS and displays results to the user.

When you execute Transaction FARR\_CONTR\_CHECK, the screen shown in Figure 4.42 is displayed.

Figure 4.42 Transaction FARR\_CONTR\_CHECK

Here, you need to specify **Accounting Principle**, **Company Code**, **Dialog Mode** (i.e., running the application in foreground or background), and **Problem class** (i.e., level of log details). In addition, it's a good idea to schedule and run this program in regular intervals. In those cases, you should pay attention to which company code is included in the run because only that company will be refreshed.

Once the program is run by clicking the **Execute** icon, data is entered in table FARR\_D\_CONS and read with Transaction FARR\_CONTR\_MON, as shown in Figure 4.43.

Besides the standard fields of **Accounting Principle**, **Company Code**, and **Revenue Accounting Contract** (i.e., range of contracts) that are optional, the most important setting is in the **Processing** section. If you select **Read data from error table**, you're reading entries previously created by Transaction FARR\_CONTR\_CHECK. If you select **Read data online**, the system reads the contract management tables and executes checks online. This option makes running Transaction FARR\_CONTR\_CHECK obsolete. However, it should be taken into consideration that executing checks online takes significant time and can cause memory dumps even in the case of a moderate number of contracts. The recommended method is using Transaction FARR\_CONTR\_CHECK to fill

the error table with contracts that have errors, and then run the monitor to display them. After the errors are corrected, the results are written to the database.

Figure 4.43 Transaction FARR\_CONTR\_MON

When you're done, click the **Execute** button. The result of Transaction FARR\_CONTR\_MON is shown in Figure 4.44 and can be split into three sections.

Exception	Contra	POB	CoCode	Acc. Princ.	Post. Liab	Post. Level	Post Liab	Leadg/LnkId	LC Calcul.	Leading	Higher-Lvl	Excl. Alloc.	BOM Root	Comp.	FulfillType	Event Type	Val R
○	200064	101176	GB04	IFRS	Contract Liability/Contract Asset	Post at Performance Obligation Level								D	E	Cl	X
○	200064	101177	GB04	IFRS	Contract Liability/Contract Asset	Post at Performance Obligation Level								D	E	Cl	X
○	200066	101179	GB04	IFRS	Contract Liability/Contract Asset	Post at Performance Obligation Level								D	E	Cl	X
○	200069	101184	GB04	IFRS	Contract Liability/Contract Asset	Post at Performance Obligation Level								D	E	Cl	X
○	200074	101193	CH04	IFRS	Contract Liability/Contract Asset	Post at Performance Obligation Level					X			D	E	MA	X
○	200074	101194	CH04	IFRS	Contract Liability/Contract Asset	Post at Performance Obligation Level					X			D	E	MA	X
○	200075	101195	GB04	IFRS	Contract Liability/Contract Asset	Post at Performance Obligation Level								D	E	Cl	X
○	200076	101196	GB04	IFRS	Contract Liability/Contract Asset	Post at Performance Obligation Level								D	E	Cl	X
●	200077	101197	GB04	IFRS	Contract Liability/Contract Asset	Post at Performance Obligation Level								D	E	Cl	X
●	200077	101198	GB04	IFRS	Contract Liability/Contract Asset	Post at Performance Obligation Level								D	E	Cl	X
○	200078	101199	GB04	IFRS	Contract Liability/Contract Asset	Post at Performance Obligation Level								D	O	MA	
○	200079	101200	GB04	IFRS	Contract Liability/Contract Asset	Post at Performance Obligation Level								D	O	MA	X
○	200080	101201	GB04	IFRS	Contract Liability/Contract Asset	Post at Performance Obligation Level								D	E	Cl	X
○	200080	101202	GB04	IFRS	Contract Liability/Contract Asset	Post at Performance Obligation Level								D	E	Cl	X
○	200081	101203	GB04	IFRS	Contract Liability/Contract Asset	Post at Performance Obligation Level								D	E	MA	X
○	200082	101204	GB04	IFRS	Contract Liability/Contract Asset	Post at Performance Obligation Level								D	E	MA	X

Figure 4.44 Error Monitor Results

In the first part, you see information about the POB that was checked and the liability posting level. In addition, on the right side, you can see fulfillment information with compounding information, if any. When you scroll to the right, you'll see which checks were performed successfully and which have errors, as shown in Figure 4.45.

Figure 4.45 Errors Found While Processing Consistency Check

**Note**

You can find PDF versions of Figure 4.44 and Figure 4.45 available for download at [sap-press.com/5700](http://sap-press.com/5700) under the **Product supplements** section.

In this case, POB number **200077** has failed data validation check **E11**. If you read the documentation about data validation errors, you'll find that E11 represents errors related to calculation of contract liability if the option of calculation on the contract but posting on the POB level is used. So, to determine the reason for the error, you need to compare values in the columns for planned liability versus posted liability. It's possible that running program B was skipped or there's some other problem to be investigated further. But depending on the impact on reporting, the decision might simply be to exclude the contract from processing or to fix a problem in the current month.

**4.4.4 Resolving Errors without Transaction FARR\_RAI\_MON**

In certain cases, organizations have thousands of RAIs being created on a daily or weekly basis, and often in such situations, Transaction FARR\_RAI\_MON can't be used as the main tool for error reporting/issues resolution. So, what should you do in such cases?

The first step is to check how many items have an error in the processable table. Run Transaction SE16N over the processable order RAI table (table /IRA/OSD012MI), for example (most problems will be with order items). Here, you need to filter the **Error** column, as shown in Figure 4.46.

In the popup that appears after you click **Error**, select value **3** because it represents processing data moving from the **Processable** to **Processed** status. Options **1** and **2** are related to checks performed in which the item is either saved in the **Raw** status or moved to the **Processable** status, respectively. So, these statuses will be used only when the **Raw** status is set. The system will provide the total number of data items in error. You can further limit the date to get a list that only shows items that failed while processing at a specific time.

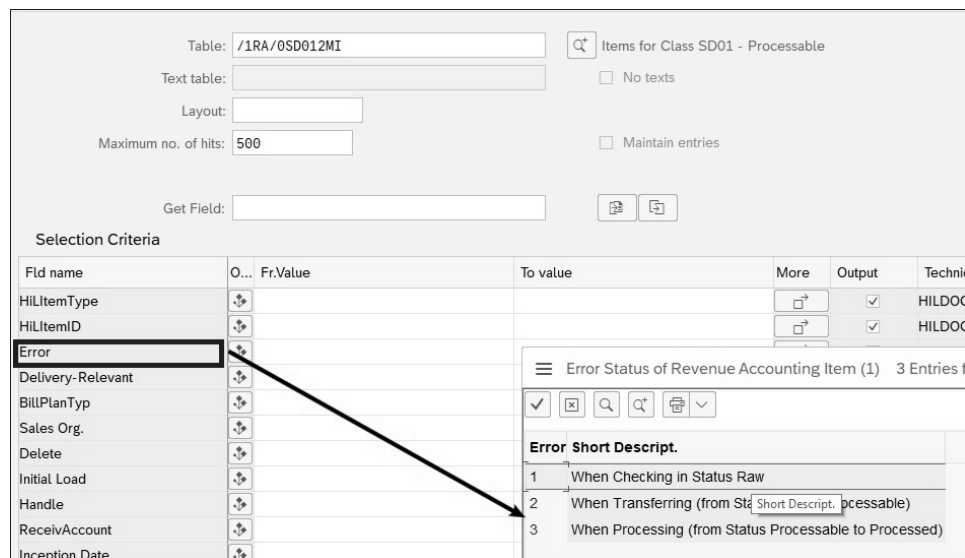


Figure 4.46 Errors in Processable Table

Now, you have a list of items that are in error, which caused them not to be processed. The next step is to look at the log. Each processing of RAIs is saved in the log with a different level of information, and you can check that information further. Run Transaction SLG1, as shown in Figure 4.47.

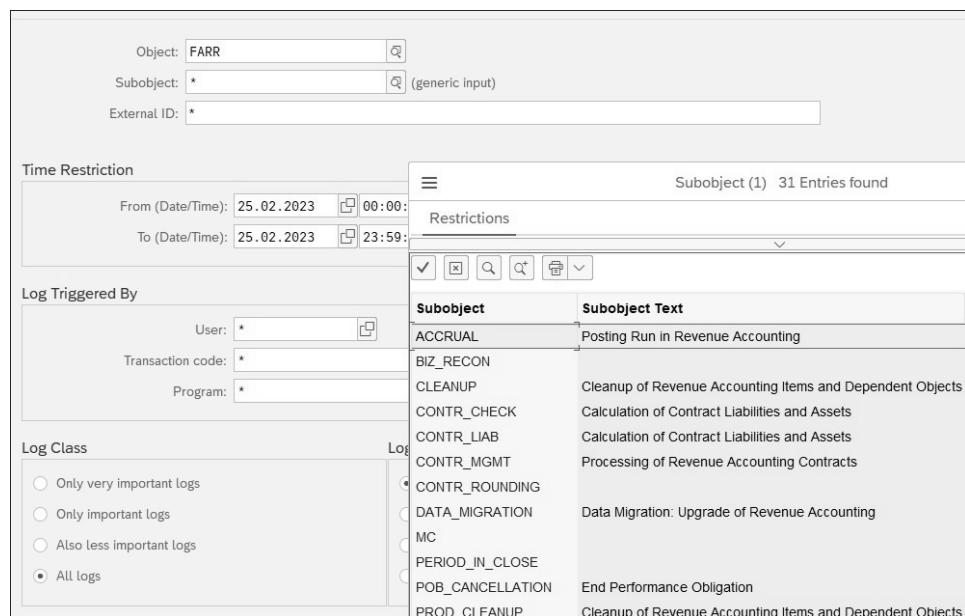


Figure 4.47 Transaction SLG1

In the **Object** field, you'll enter "FARR", which causes a list of possible entries to appear in the **Subobject** column. In this case, it can be either **CONTR\_MGMT** if you're searching for errors during processing of contracts, or **RAI\_PROCESS** (not shown) if you're searching for errors during RAI processing. Once you execute the transaction, you'll arrive at the screen shown in Figure 4.48.

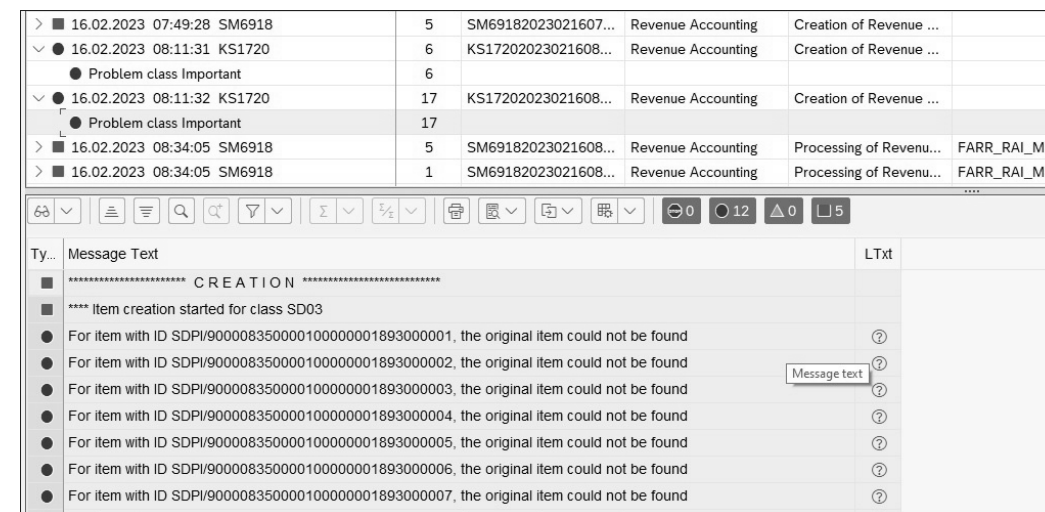


Figure 4.48 List of Errors in Transaction SLG1

You can see that the information in Transaction SLG1 is very similar to the information in Transaction FARR\_RAI\_MON.

The conclusion on error management related to RAIs is that unverified data coming to the ARL should be avoided as much as possible in the first place. Different proactive error management approaches can be taken: either by implementing logic in applications that create orders or by implementing layers that will be used to clean data before it reaches the ARL.

Once data is in the ARL, standard mechanisms are triggered to protect data integrity in RAR itself. You can extend these checks with available methods, or you can create a custom error management module before data is passed further.

All of this emphasizes the key message: data quality is crucial for RAR to fulfill its main purpose as an engine to calculate IFRS 15 revenue.

### 4.5 Business Rules Framework Plus

Business Rules Framework plus (BRFplus) is an SAP tool that is used to reduce complexity when it comes to defining business rules. Rules that are generated by BRFplus can be easily incorporated in different applications used in SAP.

Think, for example, about validations or substitutions in which you define rules once that you can later reuse across different applications. Compare that with the old methods where you needed to define rules based on an application area, and the number of areas determined the number of rules.

BRFplus isn't related to SAP HANA nor is it that new, but with newer releases of SAP S/4HANA, it's getting more and more attention. The overall BRFplus architecture is shown in Figure 4.49.

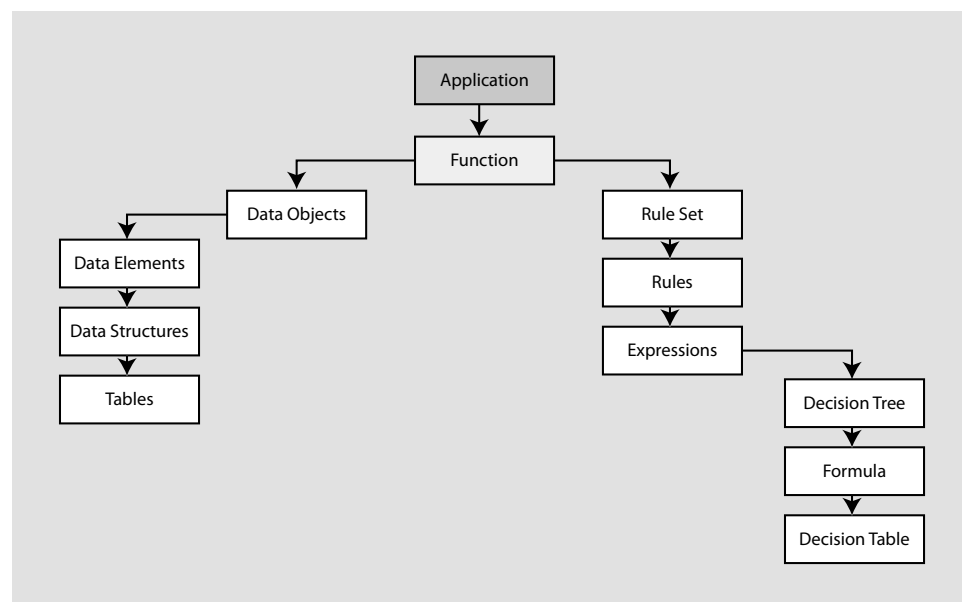


Figure 4.49 BRFplus Architecture

Let's walk through each part of the BRFplus architecture:

#### ■ Application

The application is a simple container in BRFplus for other BRFplus objects. It's classified highest in the hierarchy, and you can create as many applications as needed. There are three types of applications:

- System applications
- Master data applications
- Customizing applications

The main difference is in transports: system and customizing applications are always transportable, whereas master data applications are created locally.

#### ■ Function

This is the rule interface and behaves as a link between application code and BRFplus code. Function has a context and a result. Context is an import parameter when an application is being called, and Result is the return of its execution.

#### ■ Rule set

Rule set is nothing but a collection of rules that will be executed for a specific business case. It's an entry point for tasks to be executed. The rule object is a technical representation of a simple business rule to be applied on a specific object. Rules have to be assigned to the rule set; they can't be executed as standalone rules.

#### ■ Expressions

Expressions make up the computational power of BRFplus where each contains a logical formula that needs to be executed. There is a predefined number of expression types, and BRFplus is enhanced each time with a new set of expression types.

#### ■ Decision tables

Decision tables belong to a catalog within each BRFplus application. It's a crucial part because it holds all rules based on which the derivations of target values are determined.

With this architecture in mind, we'll explore BRFplus in the context of RAR in the following sections. We'll start with the available applications and structures, and then explain how to set up BRFplus and relevant extensions.

### 4.5.1 Applications and Structures for RAR Integration

SAP delivers different BRFplus template applications that are integrated with RAR. Applications delivered are to be used for separate functions:

- Integration with sender components
- Account determination
- Determination of POB status

To access delivered applications, run Transaction BRFPPLUS. You'll arrive at the screen shown in Figure 4.50.

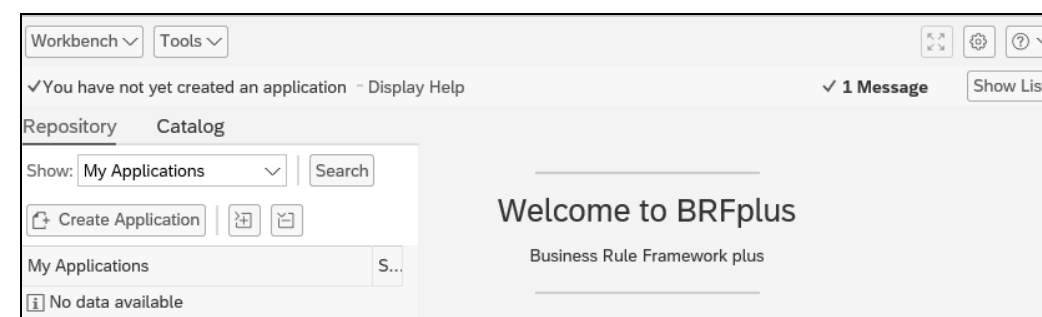


Figure 4.50 Transaction BRFPPLUS

To find the application that you need, you can perform a search on the left part of the screen. Here, you can see all the information about the application selected. Click the **Search** button in the **Repository** section.

### BRFplus Expert Mode

It's a good idea when working on development of BRFplus application to be in expert mode. By default, the transaction is running in **Simple** mode, which can be changed. In the top right corner, select the **Personalize** button (not shown), and a further list of options will appear. On the left part of the window, a selection named **User** mode will appear. Select **Expert** from the dropdown list. Once this is done, you'll see technical names instead of descriptions and a few more features that are helpful when it comes to the development of applications.

The **Search** screen shown in Figure 4.51 will appear. To find the RAR applications required, enter "FARR\*" in the **Application** area, which will display all standard applications for RAR. Click **Ok**.

Figure 4.51 BRFplus Application Selection

The first step in the process is to select which applications are needed. To work with the ARL, the first needed application is one used for processing data and creating RAR contracts and POBs. Here, you'll select applications based on the type of integration used:

- FARR\_AP\_CA\_PROCESS\_TEMPLATE for integration with SAP Billing and Revenue Innovation Management
- FARR\_AP\_SD\_PROCESS\_TEMPLATE for integration with sales and distribution
- FARR\_AP\_CRM\_PROCESS\_TEMPLATE for integration with SAP CRM
- FARR\_AP\_PROCESS\_TEMPLATE for integration with external non-SAP components

Once you select an application that is proper for integration in a specific case, it's highly recommended to copy it to the new application in a specific namespace. In addition, be sure to also select the **Contained Objects** option when copying the application. This option will move all corresponding objects to the target application too, so no object will be missed.

### Important Note

Special attention should be paid to changes and transporting of changes in BRFplus. When creating an application, you need to assign a transport request to which changes will be saved. All BRFplus applications for RAR are of type system, which means they can't be maintained in production directly, but they can be maintained on the quality assurance system. Situations where applications are maintained in separate systems should be avoided. Because each application has its own unique ID, maintenance in different systems would cause transport to become impossible. The aim should be to maintain applications in one system and move changes by a regular transport route.

Once you've copied the necessary application, you can maintain it, which means you're maintaining the decision tables. Decision tables contain a set of rules or values that represent the configuration according to which decisions are made.

The next step is to configure these rules in the decision table. On the left-hand side, under the **Expression** menu, you can find all the decision tables that need maintenance. All columns available can be split into two sections: (1) input columns that represent input parameters for decisions and (2) output columns that represent results. Input columns are marked with gray, and output columns are marked in green.

First, you need to maintain the decision tables needed for integration with sender components. There is an order in which decision tables functions are executed, and their descriptions follow:

- **DT\_PROCESS\_COMPOUND**  
Applied for no bill of materials (BOM) items and contains a set of rules for how compounding will be performed.
- **DT\_PROCESS\_BOM**  
Contains members of BOM to be managed as distinct or nondistinct POBs.
- **DT\_PROCESS\_POB**  
Main table that contains rules for how POB is determined, including type, fulfillment type, deferral method, start and end dates, and so on.
- **DT\_PROCESS\_POB\_ADD**  
Contains links to implicit POBs.
- **DT\_PROCESS\_SSP**  
Used for SSP determination. SSPs can be determined in two ways, either sent by sender component or determined in BRFplus. If needed, the SSP can be entered in this decision table together with tolerance limits and calculation type.
- **DT\_PROCESS\_DEFERAL**  
Used for special condition types such as right of return (ROR) to determine deferrals.
- **DT\_PROCESS\_HEADER**  
Used to determine contract header attributes such as description and contract category.

### Processing SSPs in BRFPplus

As stated, a standard decision table in BRFPplus is table DT\_PROCESS\_SSP, which is used for determining the SSP for the POB. However, there are limitations on how this feature should be applied. A huge amount of data in BRFPplus can seriously hamper system performance, and there is a limit of around 10,000 entries, which should be respected. Regarding SSPs, there are other features, such as validity period, which often are needed by the customer (available in condition records maintenance) and which could easily reach that limit even for a moderate number of materials. So, using BRFPplus as a main source for SSP maintenance should be used only as a last resort and only when following the limitations such an approach can impose.

Maintenance of decision tables is simple and intuitive: you need to enter criteria based on which target values are being determined. For POB determination, for example, you'll enter material and/or document type and line-item category as input parameters, and you'll enter the POB type being determined as the target.

Once decision tables for POB determination are maintained, the next application that requires maintenance is FARR\_ACC\_DETERMINE\_TEMPLATE. The approach is similar, and this template application needs to be copied to the customer namespace where changes will be performed.

The following decision tables are available for setup:

- **FARR\_ACCT\_DETERMINE\_DT\_CORR**  
Used for revenue adjustment for allocated revenue with posting category RV in table FARR\_D\_POSTING when running program A.
- **FARR\_ACCT\_DETERMINE\_DT\_CORR\_A**  
Used the same way, just for linked POBs.
- **FARR\_ACCT\_DETERMINE\_DT\_CT\_AST**  
Used for postings of contract assets with category CA in table FARR\_D\_POSTING while running program B.
- **FARR\_ACCT\_DETERMINE\_DT\_CT\_LIB**  
Used for postings of contract liability with category CL in table FARR\_D\_POSTING while running program B.
- **FARR\_ACCT\_DETERMINE\_DT\_DCOGS**  
Used for posting cost deferrals in scenarios where cost recognition is used with category CJ in table FARR\_D\_POSTING.
- **FARR\_ACCT\_DETERMINE\_DF\_REV**  
Used for posting deferred revenue with posting category DR in table FARR\_D\_POSTING.
- **FARR\_ACCT\_DETERMINE\_DT\_RADJ**  
Used for posting receivable adjustments with category RA in table FARR\_D\_POSTING after running program A.

- **FARR\_ACCT\_DETERMINE\_DT\_RC\_CST**  
Used for recognized costs in a cost deferral scenario with category CO in table FARR\_D\_POSTING.
- **FARR\_ACCT\_DETERMINE\_DT\_RC\_CST**  
Used for recognized revenue with category RA in table FARR\_D\_POSTING.
- **FARR\_ACCT\_DETERMINE\_DT\_ROR**  
Used for ROR posting. It's represented with a separate condition type and not a posting category.
- **FARR\_ACCT\_DETERMINE\_DT\_UB\_REC**  
Used for unbilled receivables with category UR in table FARR\_D\_POSTING.

Besides these, if optimized contract management (OCM) is used, you can also customize table FARR\_ACCT\_DETERMINE\_DT\_ASST\_IM for impairment postings by posting category AI in table FARR\_D\_POSTING. Impairment is the result of termination: it represents a balance of either the contract asset or contract liability that is being moved later to profit and loss (P&L).

To maintain BRFPplus tables, you click the + button to add new entries. The same approach is needed for all decision tables: input parameters are marked in gray, and target (or export) parameters are marked in green. You need to select from the list of possible entries, which will depend on the data element assigned to the table. Once entries are added, the table needs to be saved and later activated.

Not all tables require maintenance for all business scenarios. For example, if only one accounting principle is used with the contract assets/contract liabilities (CA/CL) calculation (see Chapter 5, Section 5.1.3), then maintenance of unbilled receivables and deferred revenue tables isn't necessary. Similarly, if there is no cost recognition process, then the table for cost deferral can be left empty.

Once all tables have been maintained, rules are configured (see Chapter 5, Section 5.2.2 for more information on creating rules for POBs). To complete integration with RAR, application assignments need to be performed, which we'll discuss next.

#### 4.5.2 BRFPplus Setup in RAR

Once tables have been maintained, you need to make assignments to RAR by using Transaction FARR\_IMG and going to **Revenue Accounting • Inbound Processing • Revenue Accounting Item Management • Assign BRFPplus Applications to Revenue Accounting Item Classes**.

Once the transaction is run, you need to assign the application that was customized in the previous step to appropriate RAI classes. This step is only necessary for order item classes.

The next step is assignment of the other two applications: one for POBs and one for postings/account determination. The application for POB statuses is optional, but for postings and account determination, it's mandatory. However, these steps can be found in Chapter 5. You execute the process by accessing Transaction FARR\_IMG and going to **Revenue Accounting Contracts • Assign BRF+ Application to Revenue Accounting Processes**. As shown in Figure 4.52, you need to select which application is needed for which process. There are two processes available that can be selected depending on your needs: **AD Account Determination** and **PS Performance Obligation Status**.

BRF+ Rule Configuration	
BRF+ Related Process	BRF Application
<input type="checkbox"/> AD Account Determination	▼ FARR_ACC_DETERMINE_PG1
<input type="checkbox"/> PS Performance Obligation Status	▼ FARR_POB_STATUS_PG1

Figure 4.52 Posting Application Assignment

Now, to change some entries in decision tables, the most common method is via Microsoft Excel. BRFplus comes with an embedded Microsoft Excel download/upload functionality so you can easily download all entries from the decision table, quickly make updates, and then upload again. The template fits the structure of the decision table, making it simple to share among different people and upload back to BRFplus without any additional adjustments. Every decision table in the application has this option available.

You also have additional options that make maintenance of BRFplus decision tables easier. In the menu at the top of the screen, you have the option of defining which BRFplus decision table can be maintained using the simplified BRFplus user interface (UI).

The next step is to assign specific decision tables to be maintained by the simplified UI. Navigate to **ID for Decision Table UI** in the dialog structure, as shown in Figure 4.53. Once you select this option, there will be a link between tables used for customizing and the BRFplus application they belong to.

Dialog Structure	
ID for Decision Table UI	
Decision Table ID Used for Customizing	BRF Application
<input type="checkbox"/> PROCESS_BOM	FARR_AP_SD_PROCESS_PG1
<input type="checkbox"/> PROCESS_COMPOUND	FARR_AP_SD_PROCESS_PG1
<input type="checkbox"/> PROCESS_DEFERRAL	FARR_AP_SD_PROCESS_PG1
<input type="checkbox"/> PROCESS_HEADER	FARR_AP_SD_PROCESS_PG1
<input type="checkbox"/> PROCESS_POB	FARR_AP_SD_PROCESS_PG1
<input type="checkbox"/> PROCESS_POB_ADD	FARR_AP_SD_PROCESS_PG1
<input type="checkbox"/> PROCESS_SSP	FARR_AP_SD_PROCESS_PG1

Figure 4.53 Assignment of BRFplus Applications to the Simplified GUI

Once you select the decision table in Figure 4.53, it can be maintained with the simplified GUI, as shown in Figure 4.54. The main (right-hand) part of the simplified GUI

shows which application is selected and which decision tables are created in the application. Once you select the needed table decision table, maintenance will begin.

Application Name	Decision Table	Description
<input type="radio"/> FARR_AP_SD_PROCESS_PG1	DT_PROCESS_BOM	
<input type="radio"/> FARR_AP_SD_PROCESS_PG1	DT_PROCESS_COMPOUND	Process Compound
<input type="radio"/> FARR_AP_SD_PROCESS_PG1	DT_PROCESS_DEFERRAL	
<input type="radio"/> FARR_AP_SD_PROCESS_PG1	DT_PROCESS_HEADER	
<input type="radio"/> FARR_AP_SD_PROCESS_PG1	DT_PROCESS_POB	DT_PROCESS_POB
<input type="radio"/> FARR_AP_SD_PROCESS_PG1	DT_PROCESS_POB_ADD	Process additional POB
<input type="radio"/> FARR_AP_SD_PROCESS_PG1	DT_PROCESS_SSP	Process SSP

Figure 4.54 Simplified GUI

The simplified UI option enables you to quickly change data required for POB determination of, for example, the SSP. However, you should be very careful regarding maintenance. As previously mentioned, maintenance of RAR BRFplus data is an activity that should be taken with extra caution. Settings made here can have multiple impacts, and the recommendation is to always respect the transport route for changing anything in decision tables.

## 4.6 Creating Custom Revenue Accounting Items

We've already discussed the creation of RAIs in previous sections. The RAIs that are created all adhere to the configurations and the rules defined. There are several instances when you need to validate, manipulate, delete, enhance, or perform a lot of other things on RAIs. The custom fields that you add need to be filled or validated and that can be achieved here. So, customizing the RAIs and controlling the further processing of RAIs is the main discussion of this section.

The RAIs that are created in RAR are dependent on the information that is sent from the source system. The source systems are called the sender component, as discussed in Chapter 2, Section 2.3.1. The data that is sent is converted to RAIs. It's very important for the data to be right to create correct RAIs.

With data, there are always issues, so keep the following in mind:

- The data can't be assumed to be right always, so you need to validate it.
- The data can't always be in the expected format, so you need to check the data format.
- The number of characters that is expected for a field may be exceeded, so you need to check that.



# Contents

Preface .....	13
<b>1 Introduction to Revenue Recognition</b> .....	<b>17</b>
<b>1.1 What Is Revenue Recognition?</b> .....	<b>18</b>
<b>1.2 Five-Step Model of IFRS 15</b> .....	<b>19</b>
1.2.1 Step 1: Identify the Contract .....	20
1.2.2 Step 2: Identify Performance Obligations .....	23
1.2.3 Step 3: Determine the Transactional Price .....	25
1.2.4 Step 4: Allocate the Transactional Price .....	26
1.2.5 Step 5: Recognize Revenue .....	29
<b>1.3 Industry Impact</b> .....	<b>32</b>
1.3.1 Telecommunications .....	33
1.3.2 Life Sciences .....	39
1.3.3 Manufacturing .....	44
<b>1.4 Revenue Recognition and SAP</b> .....	<b>46</b>
1.4.1 Revenue Accounting and Reporting .....	47
1.4.2 Event-Based Revenue Recognition .....	52
<b>1.5 Summary</b> .....	<b>52</b>
<b>2 Designing Your Revenue Recognition Landscape</b> .....	<b>55</b>
<b>2.1 The SAP Landscape and Revenue Recognition</b> .....	<b>55</b>
2.1.1 SAP System Landscape .....	55
2.1.2 RAR in the SAP Landscape .....	58
<b>2.2 Integration with Different SAP Components</b> .....	<b>61</b>
2.2.1 Sales and Distribution .....	61
2.2.2 Customer Relationship Management .....	64
2.2.3 Billing .....	65
2.2.4 Non-SAP Systems .....	67
<b>2.3 Revenue Recognition Data Model</b> .....	<b>70</b>
2.3.1 Sender Components .....	72
2.3.2 Structures .....	74

2.3.3	Tables .....	84
2.3.4	Relationships .....	99
2.3.5	Extensions and Transports .....	103
<b>2.4</b>	<b>Choosing Your Revenue Recognition Tool .....</b>	<b>110</b>
2.4.1	RAR with SAP S/4HANA .....	111
2.4.2	RAR with SAP ERP .....	112
<b>2.5</b>	<b>Summary .....</b>	<b>115</b>
<b>3</b>	<b>Configuring Inbound Processing (Classic and Optimized) .....</b>	<b>117</b>
<hr/>		
<b>3.1</b>	<b>Setting Up Revenue Accounting Integration .....</b>	<b>118</b>
<b>3.2</b>	<b>Inbound Processing .....</b>	<b>122</b>
3.2.1	RAI Classes in Classic Inbound Processing .....	123
3.2.2	RAI Classes in Optimized Inbound Processing .....	131
<b>3.3</b>	<b>Extending RAI Classes .....</b>	<b>138</b>
3.3.1	Defining Extensions .....	138
3.3.2	Populating Extensions .....	140
<b>3.4</b>	<b>Summary .....</b>	<b>140</b>
<b>4</b>	<b>Revenue Accounting Items .....</b>	<b>141</b>
<hr/>		
<b>4.1</b>	<b>Processing Revenue Accounting Items .....</b>	<b>141</b>
4.1.1	Processing Methods .....	142
4.1.2	Parallel Processing Framework .....	153
4.1.3	Processing Order Items .....	155
4.1.4	Processing RAIs with a Predecessor .....	158
<b>4.2</b>	<b>Managing Revenue Accounting Items .....</b>	<b>161</b>
4.2.1	Changing Items .....	161
4.2.2	Exempting Items .....	163
<b>4.3</b>	<b>Extending Transaction FARR_RAI_MON .....</b>	<b>166</b>
<b>4.4</b>	<b>Error Resolution While Creating Revenue Accounting Items .....</b>	<b>168</b>
4.4.1	Proactive and Reactive Measures .....	168
4.4.2	Inflight Errors .....	170
4.4.3	Data Validation Checks .....	172
4.4.4	Resolving Errors without Transaction FARR_RAI_MON .....	175

<b>4.5</b>	<b>Business Rules Framework Plus .....</b>	<b>177</b>
4.5.1	Applications and Structures for RAR Integration .....	179
4.5.2	BRFplus Setup in RAR .....	183
<b>4.6</b>	<b>Creating Custom Revenue Accounting Items .....</b>	<b>185</b>
4.6.1	Custom RAI Structure .....	186
4.6.2	Custom RAI Content .....	187
4.6.3	Singleton Classes .....	200
4.6.4	Dynamic Processing Flow Controlled from Table .....	206
<b>4.7</b>	<b>Summary .....</b>	<b>211</b>
<b>5</b>	<b>Contract Management .....</b>	<b>213</b>
<hr/>		
<b>5.1</b>	<b>Setting Up Contract Management .....</b>	<b>213</b>
5.1.1	Optimized versus Classic Contract Management .....	213
5.1.2	Accounting Principles .....	216
5.1.3	Presentation Methods .....	218
5.1.4	Calculation Methods for Local Currency .....	225
5.1.5	Contract Modifications .....	228
5.1.6	Cost Recognition .....	228
5.1.7	Contract Assets and Liabilities .....	228
5.1.8	Assign Company Codes, Number Ranges, and Contract Categories .....	232
5.1.9	Condition Types .....	234
5.1.10	Posting Periods .....	235
<b>5.2</b>	<b>Setting Up Performance Obligations .....</b>	<b>236</b>
5.2.1	Create Performance Obligation Types .....	238
5.2.2	Determine POB Types in BRFplus .....	242
<b>5.3</b>	<b>Managing Performance Obligations and Event Types .....</b>	<b>252</b>
5.3.1	Over Time Revenue Recognition .....	253
5.3.2	Point-in-Time Fulfillments .....	263
<b>5.4</b>	<b>Modifying Contracts .....</b>	<b>277</b>
5.4.1	Prospective or Retrospective .....	278
5.4.2	Contract Combination .....	284
5.4.3	Terminating Contracts .....	292
5.4.4	Contract Freeze .....	295
<b>5.5</b>	<b>Handling Price Allocations .....</b>	<b>299</b>
<b>5.6</b>	<b>Summary .....</b>	<b>300</b>

<b>6</b>	<b>Revenue Posting and Reporting</b>	303
<b>6.1</b>	<b>Basics of Reporting and Calculations</b>	303
6.1.1	Posting Logic	303
6.1.2	Customization for Posting of Revenue	307
6.1.3	Table FARR_D_POSTING and Revenue Categories	316
<b>6.2</b>	<b>Posting with ABC</b>	327
6.2.1	Transfer Revenue	330
6.2.2	Calculate Contract Liabilities and Contract Assets	331
6.2.3	Revenue Posting Run	334
<b>6.3</b>	<b>Integrating with Profitability Analysis</b>	338
<b>6.4</b>	<b>Reporting</b>	341
6.4.1	SAP Fiori Applications	341
6.4.2	New Styles of Reporting and CDS Views	353
<b>6.5</b>	<b>Summary</b>	363
<b>7</b>	<b>Administration and Troubleshooting</b>	365
<b>7.1</b>	<b>Parallel Processing Framework and Performance Issues</b>	366
7.1.1	What Is Parallel Processing?	366
7.1.2	Application Server Instances	368
7.1.3	Job Server Group	369
7.1.4	Custom Parallel Processing Framework	373
7.1.5	Modularization	379
7.1.6	Packaging	382
<b>7.2</b>	<b>Reclassification</b>	383
7.2.1	Creation and Structure of ZFARR_D Tables	384
7.2.2	Populating the Custom Posting Table	388
7.2.3	Extending the Standard Posting Program	390
<b>7.3</b>	<b>Data Cleanup</b>	392
7.3.1	Data Checks	393
7.3.2	Customizing	403
7.3.3	Business Add-In FARR_BADI_RAI2	405
7.3.4	Message Capturing by Simulating	406
<b>7.4</b>	<b>Helpful Tips and Tricks</b>	414
7.4.1	Navigator	414
7.4.2	Additional Information	418
<b>7.5</b>	<b>Summary</b>	423

<b>8</b>	<b>Migrating to Optimized Contract Management</b>	425
<b>8.1</b>	<b>Business Case for Migration</b>	426
<b>8.2</b>	<b>Preparation Activities for Migration</b>	427
8.2.1	Contract Management Activation	428
8.2.2	BRFplus Verification	428
8.2.3	Business Add-In Modifications	429
8.2.4	Process Changes	432
<b>8.3</b>	<b>Performing the Migration</b>	434
8.3.1	Migration Activities	434
8.3.2	Migration Errors	437
<b>8.4</b>	<b>Post-Migration Cleanup</b>	440
<b>8.5</b>	<b>Summary</b>	441
<b>9</b>	<b>Event-Based Revenue Recognition</b>	443
<b>9.1</b>	<b>Solution Background</b>	443
<b>9.2</b>	<b>Sell-from-Stock Scenario</b>	447
<b>9.3</b>	<b>Sales-Oriented Scenario for Customer Projects</b>	448
<b>9.4</b>	<b>Revenue Recognition Keys</b>	452
<b>9.5</b>	<b>Summary</b>	452
<b>Appendices</b>		455
A	Important Business Add-Ins	455
B	The Authors	459
Index		461

# Index

## A

ABAP Dictionary .....	106
ABAP-Managed Database Procedures (AMDPs) .....	359
ABC programs .....	50, 327
Acceptance date .....	277
Account assignment .....	226, 312
Accounting principle .....	216, 316, 434
<i>company codes</i> .....	232
<i>number ranges</i> .....	233
<i>parallel</i> .....	216
Accounting Standards Codification (ASC) 606 .....	19
Actual exchange rate method .....	225
Adapter Reuse Layer (ARL) .....	47, 71, 117, 141
Administration .....	365
Allocation .....	26, 239, 299, 429
<i>effect</i> .....	235
<i>engine</i> .....	299
Allow method .....	211
Append structure .....	81
<i>sets</i> .....	84
App Finder .....	342
Application .....	178, 180, 243
<i>assign</i> .....	183
Application log .....	333
Application server instance .....	368

## B

Batch server group .....	369
Billable amount .....	225
Billing .....	65
<i>indicator</i> .....	449
Bill of materials (BOM) .....	250
Business add-in (BAI) .....	455
<i>FARR_BADI_CONTRACT_COMBINATION</i> ...	291
<i>FARR_BADI_DEFERRAL_METHOD_V2</i> ....	430
<i>FARR_BADI_PRICE_ALLOCATION</i> .....	430
<i>FARR_BADI_RAIO</i> .....	190, 220
<i>FARR_BADI_RAII2</i> .....	193, 220, 405
<i>FARR_BADI_RAII4</i> .....	196
<i>FARR_EXTENDED_CHECK</i> .....	394
<i>FARR_POSTING_ENHANCEMENT</i> .....	311
<i>FARRIC_BADI_ORDER</i> .....	121
<i>IF_FARR_BADI_DETNI_IP_VERSION</i> .....	133
<i>OCM</i> .....	429
<i>signature and implementation</i> .....	190

## Business Rules Framework plus

(BRFplus) .....	49, 177
<i>account assignment</i> .....	312
<i>architecture</i> .....	178
<i>expert mode</i> .....	180
<i>OCM</i> .....	428
<i>POB types</i> .....	242
<i>RAR integration</i> .....	179
<i>setup</i> .....	183
<i>transport</i> .....	248
Business support services (BSS) .....	35

## C

Calculation method .....	225
Call-off order .....	277
Capital sales contract .....	41
Child job .....	376, 378
Classic contract management (CCM) ...	51, 111, 213
<i>contract modifications</i> .....	431
<i>migrate to OCM</i> .....	425
<i>reports</i> .....	351
Classic inbound processing .....	123
CleanUp program .....	403
Client .....	56
Cloud .....	57
Company code check .....	197
Compound structure .....	237
Condition item .....	71, 156, 266
Condition type .....	157, 234, 266, 319
<i>FX differences</i> .....	226
Conservatism .....	225
Consideration payable to customer .....	26
Consistency check .....	129, 173
Consistency monitor .....	173
Contract .....	20, 100
<i>balance report</i> .....	345
<i>contract ID</i> .....	102
<i>currency</i> .....	225
<i>freeze</i> .....	295
<i>identify</i> .....	21
<i>shift</i> .....	308
<i>suspensions</i> .....	430
<i>terminate</i> .....	51, 292, 427
<i>type</i> .....	450

- Contract assets/contract liabilities
    - (CA/CL) ..... 218, 228, 306, 325
    - calculation* ..... 222, 230, 331
    - CL netting* ..... 231
    - contract level* ..... 232
    - POB level* ..... 229–230
  - Contract Balance Reclassification app ..... 353
  - Contract-by-contract basis ..... 21
  - Contract combination ..... 44, 284
    - automated* ..... 289
    - perform* ..... 287
    - quick combine* ..... 288
  - Contract management ..... 213
    - activate OCM* ..... 428
    - select type* ..... 214
  - Contract modification ..... 228, 277, 286
    - prospetive versus retrospective* ..... 278
  - Contract Search app ..... 258, 267
  - Control transfer ..... 30
  - Convergent invoicing ..... 66
  - Core data services (CDS) view ..... 358
  - Costing-based profitability analysis ..... 338
  - Cost object controlling ..... 275
  - Cost of goods sold (COGS) ..... 338
  - Cost-plus method ..... 27
  - Cost recognition ..... 228, 239
  - Credit posting ..... 323
  - Customer ..... 21
  - Customer acceptance clause ..... 263
  - Customer field ..... 80, 139, 186, 189, 310
    - select* ..... 83
  - Customer invoice fulfillment ..... 270
  - Customer projects ..... 448
  - Customer relationship management (CRM) ..... 34, 64
- D**
- Data cleanup ..... 392
    - customizing* ..... 403
  - Data hub ..... 69
  - Data model ..... 70
  - Data validation ..... 393, 399
    - check* ..... 170, 172
    - monitor* ..... 400
  - Day-based contract modification ..... 51
  - Debit/credit indicator ..... 310
  - Debit posting ..... 323
  - Decision table ..... 179, 181, 246
    - change* ..... 184
    - revenue posting* ..... 313
    - settings* ..... 249
    - simplified GUI* ..... 184
  - Deferral category ..... 260
  - Deferral method ..... 254
    - custom* ..... 256
    - freeze periods* ..... 296
  - Deferral table ..... 259
  - Direct posting ..... 112
  - Disaggregation of revenue ..... 344
  - Disclosure ..... 341, 344
  - Discount ..... 29
  - Distinction ..... 23
  - Due date ..... 220
  - Dynamic processing flow ..... 206
- E**
- Easy Enhancement Workbench (EEW) ..... 80
  - Enforceability ..... 22
  - Enhancement spot ..... 455
  - Enrich method ..... 187, 201, 221, 405
  - Error category ..... 397, 401
  - Error resolution ..... 168
    - categories* ..... 397, 401
    - error table* ..... 407
    - without Transaction FARR\_RAI\_MON* ..... 175
    - with Transaction FARR\_CONTR\_CHECK* ... 173
  - Event-based POB ..... 252, 264
  - Event-based revenue recognition (EBRR) .... 52, 443, 446
    - comparison* ..... 444
    - revenue recognition keys* ..... 452
  - Event-Based Revenue Recognition – Sales Order app ..... 450
  - Event type ..... 252, 264, 276
  - Exchange rate method ..... 227
  - Exempted item ..... 163, 405, 420
  - Exemption type ..... 165
  - Expression ..... 179, 181, 246, 314
  - Extension ..... 103
    - define* ..... 138
    - FARR\_RAI\_MON* ..... 166
    - populate* ..... 140
  - Extension include ..... 80, 139
  - Extract, transform, load (ETL) ..... 35, 68
- F**
- Financial accounting ..... 337
    - reconciliation* ..... 347
  - Financial document ..... 335
    - analysis* ..... 347
  - Fixed exchange rate method ..... 225
  - Fixed-price billing ..... 451
  - Flow control table ..... 208

- Foreign currency ..... 226
  - Fulfillment event ..... 119
  - Fulfillment item ..... 78, 142
  - Fulfillment type ..... 261
  - Function ..... 178
  - Function module ..... 366, 374
- G**
- General ledger account ..... 349
  - General ledger mapping ..... 387
  - Globally unique identifier (GUID) ..... 320
  - Goods delivery ..... 268
  - Goods issue fulfillment ..... 264
- H**
- Header ID ..... 100, 145, 374
  - Hierarchy ..... 100
- I**
- Impairment effect ..... 292
  - Implementation class ..... 221
  - Inbound processing ..... 117, 122
    - usage parameters* ..... 133
  - Incoterm ..... 31, 45
  - Index ..... 96
  - Inflight check ..... 170, 393
  - Input costs method ..... 44
  - Input method ..... 31
  - Instrument ..... 40
  - Integration ..... 61
    - billing* ..... 65
    - CRM* ..... 64
    - non-SAP systems* ..... 67
    - profitability analysis* ..... 338
    - results analysis* ..... 273
    - sales and distribution* ..... 61
  - Interface component ..... 75, 123
    - activate* ..... 128
    - list* ..... 122
  - Interface generation ..... 129
  - International Financial Reporting Standards (IFRS) ..... 17
    - Five-step model* ..... 19
    - IFRS 15* ..... 19, 46, 58, 305
    - industry impact* ..... 32
  - Invoice ..... 323
  - Invoice item ..... 78, 142
  - Invoice reversal ..... 304, 324
  - Item category ..... 120
- J**
- Job monitor ..... 331
  - Job server group ..... 369
    - create* ..... 370
- K**
- KEYPP field ..... 153
- L**
- Lease contract ..... 41
  - Leasing ..... 43
  - Life sciences ..... 39
  - Logical system ..... 72, 145
  - Loyalty points ..... 39
- M**
- Main item ..... 71, 142
  - Main program
    - child job* ..... 378
    - parent program* ..... 376
  - Manage Revenue Contracts app ..... 346
  - Manual fulfillment ..... 277
  - Manufacturing ..... 44
  - Margin analysis ..... 52, 338
  - Market observable price ..... 28
  - Mass process ..... 91
  - Mass transfer ..... 88
  - Matching principle ..... 447
  - Message capturing ..... 406
  - Message class ..... 393
  - Microsoft Excel ..... 247
  - Migration ..... 425
    - activities* ..... 434
    - business case* ..... 426
    - cleanup* ..... 440
    - errors* ..... 437
    - perform* ..... 434
    - preparation* ..... 427
  - Modularization ..... 379
  - Monitor Revenue Contract app ..... 346
  - Month-end closing ..... 328
- N**
- Navigator ..... 414
    - set up* ..... 416
  - Noncash consideration ..... 25
  - Number range ..... 233

## O

Operational report ..... 341, 345  
 Operations support systems (OSS) ..... 35  
 Optimized contract management (OCM) ..... 50, 111, 213  
   *BAdIs* ..... 429  
   *contract modifications* ..... 432  
   *migration* ..... 425  
   *process changes* ..... 432  
   *reports* ..... 351  
 Optimized inbound processing (OIP) ..... 51, 70, 111, 131  
   *activate* ..... 133  
   *tables* ..... 132  
 Order assignment ..... 449  
 Order item ..... 78, 142  
   *process* ..... 155  
 Original item ..... 159  
 Output cost method ..... 45  
 Output method ..... 31  
 Over time revenue recognition ..... 29, 44, 253  
   *calculation* ..... 257

**P**

Package size ..... 374  
 Packaging ..... 382  
 Parallel accounting principle ..... 217  
 Parallel buffering ..... 310  
 Parallel processing ..... 366  
 Parallel processing framework (PPF) ..... 112, 153, 366  
   *custom* ..... 373  
   *function module* ..... 374  
   *packaging* ..... 382  
   *program execution* ..... 378  
   *selection screen* ..... 373  
   *standard* ..... 366  
 Parent program ..... 376  
   *processing* ..... 377  
 Percentage of completion (POC) ..... 31, 44, 252  
   *event* ..... 272  
 Perform ..... 381  
 Performance ..... 318, 366, 382  
 Performance obligation (POB) ..... 23, 35, 236  
   *allocation data* ..... 239  
   *BRFplus* ..... 242  
   *compound* ..... 237  
   *create time-based POB* ..... 254  
   *define types* ..... 238  
   *fulfillment data* ..... 239

Performance obligation (POB) (Cont.)  
   *fulfillment type* ..... 252  
   *manage* ..... 252  
   *POB ID* ..... 101  
   *posting categories* ..... 318  
   *remaining* ..... 345  
   *table FARR\_D\_DEFITEM* ..... 260  
   *time based* ..... 49  
 Periodic service billing ..... 451  
 Perpetual contract ..... 41  
 Point-in-time revenue recognition ..... 29, 263  
 Portfolio approach ..... 21  
 Posting ..... 303, 383  
   *ABC programs* ..... 327  
   *account assignment* ..... 312  
   *categories* ..... 318, 323  
   *custom* ..... 383  
   *extend program* ..... 390  
   *GUID* ..... 320  
   *keys* ..... 308, 316, 336  
   *optimization* ..... 309  
   *run* ..... 334  
   *settings* ..... 307  
 Posting period ..... 235  
   *status* ..... 236  
 Postpaid contract ..... 37  
 Postponed item ..... 137, 164  
 Predecessor item ..... 136, 142  
   *process* ..... 158  
 Prepaid contract ..... 37  
 Presentation method ..... 218  
   *calculations* ..... 222  
 Price allocation ..... 299  
 Proactive measure ..... 168  
 Probability ..... 23  
 Processable item ..... 90, 145  
   *custom* ..... 193  
   *exempt* ..... 421  
 Processed item ..... 92  
   *custom* ..... 196  
 Processing block ..... 380  
 Processing method ..... 142  
 Profitability analysis ..... 338  
 Program enhancement ..... 124  
 Project System ..... 448  
 Proof of delivery ..... 119  
   *fulfillment* ..... 271  
 Prospective change ..... 279

## R

Raw item ..... 87, 145  
   *cleanup* ..... 403  
   *custom* ..... 190  
   *exempt* ..... 420  
 Reactive measure ..... 169  
 Reagent ..... 40, 229  
 Receivable adjustment ..... 314  
 Reclassification ..... 383  
 Reconciliation ..... 122, 337, 347  
   *reports* ..... 341, 347  
 Reconciliation key ..... 259, 316, 336, 384  
   *custom* ..... 386  
   *number* ..... 323  
 Reference key ..... 336  
 Relationship ..... 99  
 Remote function call (RFC) ..... 49  
 Reporting ..... 303, 341  
   *AMDP* ..... 359  
   *CCM versus OCM* ..... 351  
   *CDS views* ..... 358  
   *disclosures* ..... 344  
   *operational reports* ..... 345  
   *staging tables* ..... 354  
 Residual approach ..... 28, 241  
 Restoration reason ..... 165  
 Results analysis ..... 272, 443  
   *calculations* ..... 275  
   *comparison* ..... 444  
   *keys* ..... 276  
   *methods* ..... 274  
 Retrospective change ..... 282  
 Return order item ..... 136  
 Revenue ..... 18  
   *categories* ..... 316  
 Revenue accounting and reporting (RAR) ..... 17, 47, 55, 111, 445  
   *add-on* ..... 59  
   *architecture* ..... 48  
   *closing* ..... 328  
   *comparison* ..... 444  
   *data model* ..... 70  
   *data sources* ..... 71  
   *integration* ..... 61, 118  
   *landscape* ..... 58  
   *mark as relevant* ..... 120  
   *posting* ..... 303  
   *reconciliation* ..... 337  
   *sidecar approach* ..... 59  
   *tables* ..... 93  
   *versions* ..... 47, 426

Revenue accounting item (RAI) ..... 49, 71, 141  
   *automatic processing* ..... 150  
   *batch processing* ..... 51  
   *change* ..... 161  
   *custom* ..... 185  
   *customize content* ..... 187  
   *error resolution* ..... 168  
   *error status* ..... 145  
   *exempt* ..... 163, 420  
   *extend monitor* ..... 166  
   *manage* ..... 161  
   *manual processing* ..... 143  
   *mass creation* ..... 418  
   *mass processing* ..... 154  
   *modifiable fields* ..... 105, 162  
   *monitor* ..... 103  
   *postponed* ..... 137  
   *predecessors* ..... 158  
   *process* ..... 92, 135, 141, 149  
   *restore* ..... 421  
   *status* ..... 71, 76, 87, 135, 144  
   *tables* ..... 86  
   *timestamps* ..... 148  
   *transfer* ..... 89, 105, 420  
 Revenue accounting item (RAI) class ..... 77–78  
   *activate* ..... 84  
   *activate interfaces* ..... 128  
   *configuration* ..... 126  
   *custom fields* ..... 80  
   *extend* ..... 138  
   *generate interfaces* ..... 85  
   *maintain* ..... 79  
   *OIP* ..... 131  
   *type* ..... 77, 127  
 Revenue amount ..... 261, 305  
 Revenue Contract Search app ..... 346  
 Revenue Explanation app ..... 346  
 Revenue forecast ..... 259  
 Revenue posting ..... 303, 383  
   *ABC programs* ..... 327  
   *account assignment* ..... 312  
   *categories* ..... 318  
   *custom* ..... 383  
   *customize* ..... 307  
   *extend program* ..... 390  
   *keys* ..... 316  
   *run* ..... 328, 334  
   *settings* ..... 307  
 Revenue recognition ..... 29, 46  
   *calculation* ..... 305  
   *choosing your tool* ..... 110  
   *data model* ..... 70

Revenue recognition (Cont.)	
<i>design your landscape</i>	55
<i>reporting</i>	303, 341
Revenue schedule	258
Revenue Schedule app	349
Revenue transfer	305
Revenue Transfer program	230
Rule set	179
Run ID	234

## S

Sales and distribution	61, 118
Sales document type	120
SAP Billing and Revenue Innovation	
Management	65, 132, 355
SAP Business Client	343
SAP Convergent Charging	66
SAP Convergent Mediation by DigitalRoute	66
SAP Customer Financial Management	66
SAP Customer Relationship Management	
(SAP CRM)	64
SAP ERP	46, 112
SAP Fiori app	341
<i>CCM versus OCM</i>	351
SAP Fiori launchpad	341
SAP HANA	360
SAP Revenue Accounting and Reporting	46
SAP S/4HANA	47, 111
<i>releases</i>	426
SAP S/4HANA Cloud, public edition	446
SAP S/4HANA Service	64
SAP Subscription Billing	66
Selection screen	373, 408
<i>main program</i>	376
Sell-from-stock scenario	447
Sender component	72
<i>assign source item type</i>	73
<i>enable integration</i>	118
Sequence	74
Server group	369
Significant financing component	25
Simulation mode	407
Singleton class	200
<i>attributes</i>	203
<i>create</i>	202
<i>methods</i>	204
Source item type	73
Staging table	354
Standalone selling price (SSP)	26, 42, 251, 266
<i>BRFplus</i>	182
<i>change</i>	156

Standalone selling price (SSP) (Cont.)	
<i>determine</i>	27
<i>tolerance</i>	242
Standard class	187
Start date type	256
Static class	202
Structure	74
Subscription order management	65
Suspension period	430
Switch method	209
System landscape	55
<i>RAR</i>	58
<i>regional split</i>	57

## T

Table	84
<i>/IRA</i>	51, 87
<i>add index</i>	96
<i>BRFplus</i>	181
<i>conversion</i>	107
<i>custom</i>	106
<i>display structure</i>	98
<i>DT_PROCESS_BOM</i>	250
<i>DT_PROCESS_COMPOUND</i>	251
<i>DT_PROCESS_POB_ADD</i>	251
<i>DT_PROCESS_SSP</i>	251
<i>FARR_ACCT_DETERMINE_DT_ASST_IM</i>	294
<i>FARR_D_CONS</i>	173, 399
<i>FARR_D_CONTRACT</i>	440
<i>FARR_D_DEFITEM</i>	259, 263, 322
<i>FARR_D_FULFILLMNT</i>	269
<i>FARR_D_MAPPING</i>	217
<i>FARR_D_POB</i>	51, 251
<i>FARR_D_POSTING</i>	223, 303, 314, 316, 320, 322, 327, 383
<i>FARR_D_RECON_KEY</i>	317, 321, 336, 384
<i>FARR_RAI_MON</i>	269
<i>OIP</i>	132
<i>populate custom</i>	388
<i>staging</i>	354
<i>standard</i>	93
<i>temporary</i>	109
<i>ZFARR_D</i>	384
<i>ZTRANSFER_MSG</i>	413
Table maintenance generator	208
Telecommunications	33, 329
Termination fee	293
Time and material billing	451
Time-based POB	252–253, 321
<i>settings</i>	254
Time value of money	38

Transaction	
<i>BRFPLUS</i>	179, 244
<i>FARR_CCM_OCM_MIG_CON</i>	434
<i>FARR_CONTR_CHECK</i>	173, 399, 440
<i>FARR_CONTR_MON</i>	173, 400
<i>FARR_CONTRACT_LIABILITY</i>	328, 331
<i>FARR_D_POSTING</i>	324
<i>FARR_IMG</i>	72, 76, 82, 123, 183, 214, 216, 307
<i>FARR_LIABILITY_CALC</i>	112
<i>FARR_PERIOD_IN_CLOSING</i>	236
<i>FARR_RAI_CONF</i>	127, 138
<i>FARR_RAI_MON</i>	51, 87, 103, 135, 142, 153, 166, 266, 321
<i>FARR_RAI_PROC</i>	91, 152
<i>FARR_RAI_TRANS</i>	88, 150, 407, 420
<i>FARR_REPR_PPRAI</i>	137
<i>FARR_REV_TRANSFER</i>	112, 328, 330
<i>FARR_REVENUE_POSTINGS</i>	328, 334
<i>FBO3</i>	336
<i>OB22</i>	226
<i>OKG3</i>	274
<i>SE11</i>	81, 162, 208, 355, 387, 407
<i>SE14</i>	109
<i>SE16</i>	223
<i>SE16N</i>	175
<i>SE17</i>	109
<i>SE18</i>	121, 133, 189, 220, 289, 291, 406
<i>SE24</i>	202, 206, 418, 423
<i>SE37</i>	421
<i>SE38</i>	355, 408, 416
<i>SE91</i>	393
<i>SHDB</i>	112
<i>SHDB PFW</i>	112
<i>SLG1</i>	147, 176, 333, 335, 407
<i>SM30</i>	208, 276

Transaction (Cont.)	
<i>SM36</i>	371
<i>SM37</i>	331, 355
<i>SM61</i>	369
<i>SNRO</i>	310
<i>SOAMANAGER</i>	297
<i>SPRO</i>	119
<i>VF44</i>	61, 112
<i>VLPOD</i>	272
<i>ZRAR_NAVIGATE</i>	415
Transactional price	25, 42, 266
<i>allocation</i>	26
Transfer Revenue program	228, 305, 324, 328, 330
Transport	103
<i>request</i>	106
Troubleshooting	365
Type	74

## U

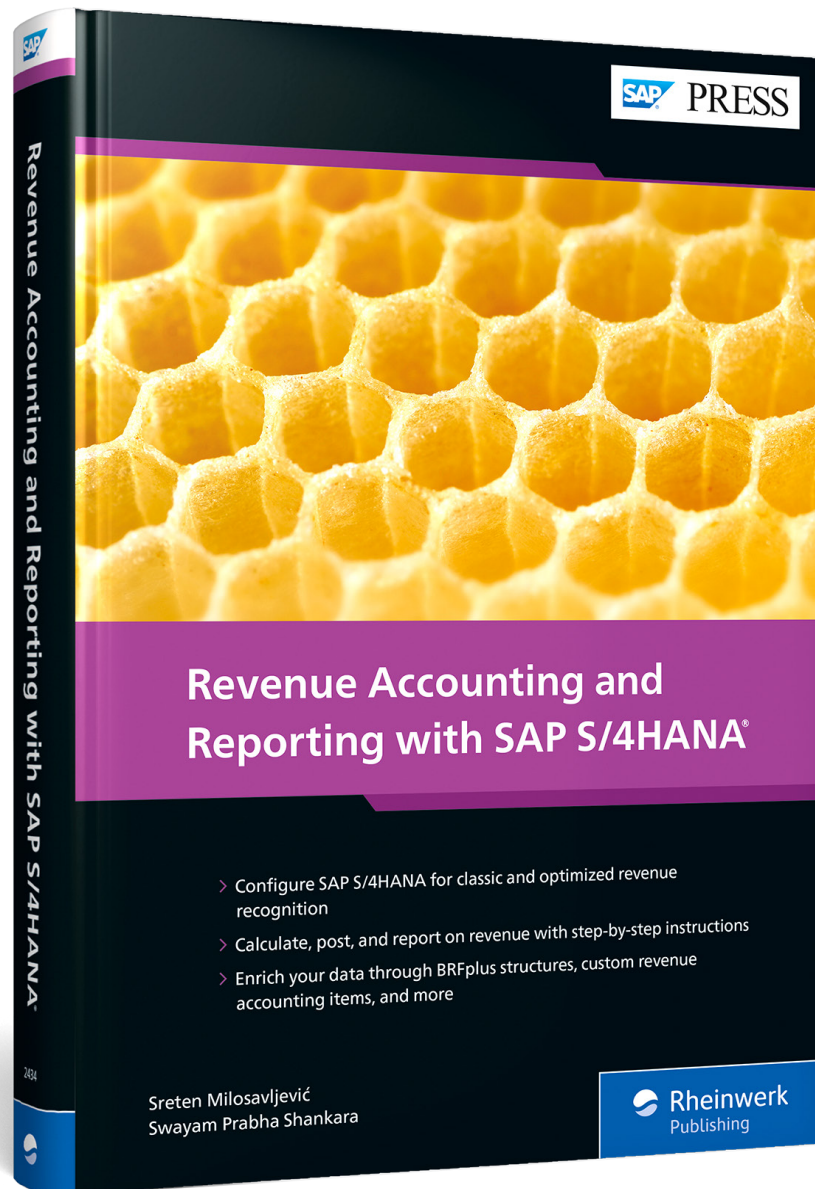
Unbilled receivable/deferred revenue	
(iR/DR)	219
<i>calculation</i>	223
Unconditionality	220
Upload rule	130

## V

Variable consideration	23, 25, 38
------------------------	------------

## W

Weighted average rate	227
Work breakdown structure (WBS) element	448



Sreten Milosavljević, Swayam Prabha Shankara

## Revenue Accounting and Reporting with SAP S/4HANA

467 pages | 11/2023 | \$99.95 | ISBN 978-1-4932-2434-0

 [www.sap-press.com/5700](http://www.sap-press.com/5700)



**Sreten Milosavljević** has worked as a senior SAP finance consultant for more than 20 years. Thanks to the perfect storm of International Financial Reporting Standards (IFRS) 9, IFRS 15, and IFRS 16, since 2016, his primary focus has been on working with revenue accounting and reporting and contract lifecycle management. During this time, he gained experience with most of the industries that are impacted by the implementation of either IFRS 15 or Accounting Standards Codification (ASC) 606, including telecommunications, healthcare, manufacturing, and hightech. Sreten is currently involved in solving complex topics for a variety of customers, especially regarding integration points between different systems.



**Swayam Prabha Shankara** has been a technical consultant for more than 16 years. Over this period, she has worked with different technologies to apply the latest approaches. She has been working with revenue accounting and reporting in the SAP space since 2017, beginning with a project for a major telecommunications customer that had complex issues related to data volume and performance. She has continued working on different revenue accounting and reporting projects, updating her skills for SAP HANA, core data services (CDS) views, and system design and development.

*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.*