SAP PRESS

SAP for Utilities

SAP
Biller Direct

POWERED BY SAP HANA

SAP S/4HANA

A Business and Technical Roadmap to Deploying SAP

WHATSAPP +255738656506

INTRODUCTION

Welcome to the fascinating world of SAP. This book helps you crack the tricks of mastering SAP HANA Customization

Biller Direct

Process Integration with Accounts Receivable Accounting

Release Credit Cards

Use

You only carry out this IMG activity if you are using the application *FSCM Biller Direct*.

You specify whether Web users can release their bills for payment by credit card.

Activities

Enter the payment method class *card payment* .

By doing so the card payment via the Web is released, and you can flag the new payment method as being for card payment.

Edit Payment Methods

Use

You execute this IMG activity only if you are using the Web application *SAP Biller Direct* .

If you permit Web users to make payments by bank collection (automatic debit) or credit card, you have to determine which payment method is to be used for the payment card.

Requirements

You have created the new payment methods and defined bank determination. You do this in the following IMG activities:

- Set Up Payment Methods per Country for Payment Transactions.
- Set Up Payment Methods per Company Code for Payment Transactions
- Set Up Bank Determination for Payment Transactions

Activities

For each company code that permits payment via the Web, enter a payment method for the classification *Credit card* and *Bank collection* (if you wish to allow both payment methods).

If you always want to allow bank collection for particular bank details, then choose the payment method classification 'Automatic debit in master record'. The payment method you define here is then entered in the company code-dependent data of the business partner (customer).

Set Account Determination for Credit Cards

Use

You only carry out this IMG activity if you are using the Web application *FSCM Biller Direct*.

With FSCM Biller Direct your Web users can release bills via the Web for payment with payment cards. If a bill is paid via the Web with a payment card, this is managed on a G/L account on which you collect all receivables that you then report to the credit card company using a settlement program.

Requirements

You have executed the IMG activity Assign G/L Account to Cash Clearing Account

Activities

Depending on the payment card type, the card type, the payment method and the currency, enter the G/L account on which you collect the receivables due from the credit card company.

This involves those accounts that you have entered in the IMG activity Assign G/L Account to Cash Clearing Account .

Assign Text ID

Use

You only execute this IMG activity if you are using the Web application *FSCM Biller Direct* .

Your Web users can use FSCM Biller Direct to create inquiries or complaints, and to give reasons for any partial payments. Your company accounting clerk can then respond to these. In accounts receivable accounting these electronic messages are managed as texts with a particular text identification for the appropriate bill. To be able to do this you define the text identification, and in this IMG activity assign the text ID to Biller Direct.

Requirements

You have defined a text ID. To do this execute the IMG activity Define Text IDs for Documents .

Activities

Enter the appropriate Text ID for your current client.

Define Entries for the Logging of the Activities of Web users

Use

You only execute this IMG activity if you are using the Web application*FSCM Biller Direct* and use the function *Logging the Activities of the Web Users*.

Requirements

- The log entries use interval 1 of the number range object EBPP_EVENT. Hence you need to have created this interval (SAP initial screen under *Tools -> ABAP Workbench -> Development -> Other Tools -> Number Ranges*).
- Log entries are personal data. In the current version of *FSCM Biller Direct* it is **not** possible for you to deactivate logging for individual Web users. Check whether you need to obtain the consent of the Web users under the data protection legislation of your country.

  In Germany, for example, the storage of personal data is only permitted under the data protection legislation providing the the user has given consent to it. In this case you have to obtain the agreement of all users before activating the log.

Activities

For each log category execute the following settings:

- Activate recording
- Determine minimum retention period of the recordings
- Determine aggregation
- Exclude company codes from the logging Deactivation is not possible for all log categories.

Use transaction **EBPP_LOG_DISP** to evaluate the log entries. You can use transaction **EBPP_LOG_ORGA** to delete old log entries.

Define Partial Payment and Currency

Use

You only carry out this IMG activity if you are using the Web application *SAP Biller Direct*.

You specify the following for the Web users:

- Always allow partial payments for payment instructions using the Internet
- Allow partial settlement of credits for payment instructions using the Internet If you allow partial payments, this means you can also allow partial settlements of credits.
- Currency for bills and credits

  You should only define one currency here if bills can be posted in different currencies on one account. The currency defined here is automatically the display and payment currency. If you do not define a currency here, the display and payment currency is the currency of the most recent bill.
- Determine posting logic for partial payments

  You specify whether, following the payment run, a bill paid using SAP Biller Direct is cleared under residual item creation, or whether the bill remains open with a partial payment.
- Determine minimum payment amount for partial payments

  You can specify a minimum payment amount, a percentage rate or both. The upper limit applies. If you want to use either amounts or specified percentages only, then leave the value in the other field blank.
- Allow discount for partial payments

  You specify whether a discount is allowed for partial payments. You have the following variants:
- For residual item creation, a discount is allowed with the first partial payment. Note that you can specify a minimum payment amount using the tolerances.
- For partial payment creation, the discount is only allowed with the final payment.
- Explicitly allow a proportional discount.

Define Payment Block for Release Process

Use

When you activate this function, all of the new bills posted from component *Sales and Distribution (SD)* are provided with a payment block. Bills blocked in this way are displayed in the Web application *SAP Biller Direct* on tab page *Bills to be Approved*, and *not* on the tab page *Open Bills*. If a user who has the appropriate authorization approves one of these bills in the Web application, the payment block is removed from the bill (from the open items), and the bill is displayed on the tab page *Open Bills*. You will usually find two users are involved in this process:

- The first user is authorized to release bills
- The second user is authorized to pay bills

Requirements

- If you want to create your own payment block, you have defined this in the following IMG activity: Define Payment Block Reasons.
- To make it possible to set payment blocks when posting (SD) bills, the payment block has to be set in a substitution when posting a document. Carry out the following IMG activity: Business Transaction Events.
  a) Choose *Settings -> Products -> ...of a Customer*.

  b) Make a suitable entry in the customer name space without specifying an RFC destination.

  c) Set the entry to active.
  Example: ZAPR with text (release workflow for Biller Direct and active indicator)

  d) Choose *Settings -> Process Modules -> ...of a Customer*.

  e) Enter function module APAR_BD_BTE_112_IMPL for process 112 and the product you specified.
  Example: 112, APAR_BD_BTE_112_IMPL, ZAPR Do not make any entries in the fields *Ctr* und *Appl.*.

  f) Save your entries.

Activities

Specify a payment block that is entered in the open items when you post a new bill.

Define Reference User

Use

In this IMG activity you assign a description of the authorization to existing reference users of SAP Biller Direct.

Requirements

You have created the reference user with the corresponding authorizations.

Example

Reference user NAME1 is authorized to display and pay bills with *SAP Biller Direct*. In this case enter the following:

Reference User	Description Reference User

NAME1	Authorization to display and pay

Business Add-Ins (BAdIs)

BAdI: Determination of Bills

Use

This Business Add-In (BAdI) is used in theSAP Biller Direct *(FIN-FSCM-BD-AR)* component.

It is used to determine bill and payment data that is displayed in the *SAP Biller Direct* Web application.

You can implement the following functions for displaying transactions in the Web application:

- Display a text for a bill in the bill overview
- Display special G/L transactions
- Prevent transactions, such as the clearing of payments on account

The BAdI is called as soon as a user logs on to *SAP Biller Direct* and displays his transactions (such as open bills and paid bills) in the Web application.

Standard settings

The Business Add-In is active in the standard system. The default coding is executed automatically.

Default implementations have been defined in the standard system for the individual methods. If you only want to overwrite one method, call the default coding for the other methods.

Activities

If you want to display different texts for your bills and payments in the *SAP Biller Direct* Web application, or you want to display special G/L transactions in addition to the bills, you have to implement this BAdI.

After calling up the IMG activity, a dialog box appears, in which you can enter a name for the implementation.

If you have already made other implementations for this BAdI, another dialog box appears, in which the existing implementations are displayed. In this case, choose *Create*, and proceed as follows:

1. In the dialog box, enter a name for the BAdI implementation in the *Implementation* field, and choose *Create*.
   The screen for creating BAdI implementations is now displayed.
2. Enter a short text for the implementation in the *Short text for implementation* field.
3. From the tab index, choose *Interface*.
   The *Name of implemented class* field is already filled on the tab page, as a class name was automatically assigned to the implementation when you named it.
4. Save your entries, and assign the implementation to a development class.
5. Place the cursor on the method, and double-click to enter method processing.
6. Enter the code for the implementation between the statements `method <Interface name> ~ <Name of method>` and `endmethod`.
7. Save and implement your code. Return to the *Edit Implementation* screen.
8. Save the entries on the *Edit Implementation* screen.

Note: You can also create an implementation, and then activate it at a later time. In such a case, end the processing stage at this point.

9. Choose *Activate*

   The code you stored in the method will be run when the application program is executed.

Example

To display default coding, choose *Goto -> Default Coding*.

See also:

Methods

Transfer of data when logging on to SAP Biller Direct

Display of special G/L transactions in SAP Biller Direct

Display of texts and attributes for transactions in SAP Biller Direct

Setting attributes for bill payment instruction in SAP Biller Direct

Change error messages from credit card clearing houses

BAdI: Determination of Bill Details

Use

This Business Add-In (BAdI) is used in the *SAP Biller Direct (FIN-FSCM-BD)* component.

You can use this BAdI to determine bill details from accounts receivable accounting that you want to display in *SAP Biller Direct*.

In addition to the request to accounts receivable to determine the bill details, you can also support a request from the Web application direct to another Web server, for example. See Documentation for BAdI Determination of Bills (APAR_EBPP_GET_DATA).

SAP delivers the corresponding BAdI implementation for the billing components (such as EBPP_INVDETAIL_SD) for displaying billing documents from the *Sales and Distribution (SD)* component. This contains a sample implementation for system access to an optical archive for determining data for the bill using the SAP ArchiveLink interface. If you need more information, contact your SAP consultant.

Once you have run the BAdI *Determination of Bills* (APAR_EBPP_GET_DATA), the system creates a bill overview in which a bill is displayed in the Web application on one or more lines. The same happens for the display of open items, line items, and the balance confirmation. When the Web user chooses to display bill details, a request is made to accounts receivable accounting to determine the detail data.

When creating the bill overview in the BAdI *Determining Bill Details* (EBPP_INVOICEDETAIL), the billing application transfers the information for each bill regarding which formats you can use for displaying the bill details. When bill details are requested, the request is made in one or more of the formats laid down there (parameter I_INVOICE_FORMATS).

The system runs through all active implementations of the BAdI *Determination of Bill Details* (EBPP_INVOICEDETAIL). Using the customer account transferred, the bill number (BOR key), language, and format requests as a basis, the BAdI implementations check their responsibility and supply the bill details in the desired format as a result of calling the BAdI.

Standard settings

The Business Add-In can be used repeatedly. All active implementations are called and executed.

Activities

After you call the IMG activity, the system displays a dialog box where you enter a name for the implementation.

If implementations of this Business Add-In have already been created, the system displays them in a dialog box. You then choose one of them by choosing ***Create***, and continue as follows:

1. In the dialog box, enter a name for the implementation of the Add-In and choose ***Create***. The system displays the initial screen for creating Business Add-In implementations.
2. On this screen, enter a short description for you implementation in the ***Implementation Short Text*** field.
3. If you choose the ***Interface*** tab, you will notice that the system has filled in the ***Name of the Implementing Class*** field automatically, by assigning a class name based on the name of your implementation.
4. Save your entries and assign the Add-In to a package.
5. To edit a method, double-click its name.
6. Enter your implementation code between the `method <Interface Name>~<Name of Method>.` and `endmethod.` statements.
7. Save and activate your code. Navigate back to the ***Change Implementation*** screen.

   Note: You can also create an implementation for an Add-In and not activate it until later. If you want to do this, do not carry out the following step:
8. Choose ***Activate***.

   When the application program is executed, the system carries out the code in the method you wrote.

Example

See also:

Method

Determine Detail Data for the Bill

BAdI: Assignment of User and Customer

Use

This Business Add-In (BAdI) is used in the *SAP Biller Direct (FIN-FSCM-BD)* component.

Once you have logged on, you can use this BAdI to enhance or change the assignment of the user to customers. Prior to calling the BAdI, the system determines which customers have been assigned to the user in user management. The BAdI is then called in table T_OBJKEYS with this quantity.

Activities

After you call the IMG activity, the system displays a dialog box where you enter a name for the implementation.

If implementations of this Business Add-In have already been created, the system displays them in a dialog box. You then choose one of them by choosing ***Create***, and continue as follows:

1. In the dialog box, enter a name for the implementation of the Add-In and choose ***Create***. The system displays the initial screen for creating Business Add-In implementations.

2. On this screen, enter a short description for you implementation in the ***Implementation Short Text*** field.

3. If you choose the ***Interface*** tab, you will notice that the system has filled in the ***Name of the Implementing Class*** field automatically, by assigning a class name based on the name of your implementation.

4. Save your entries and assign the Add-In to a package.

5. To edit a method, double-click its name.
6. Enter your implementation code between the `method <Interface Name>~<Name of Method>.` and `endmethod.` statements.

7. Save and activate your code. Navigate back to the ***Change Implementation*** screen.

   Note: You can also create an implementation for an Add-In and not activate it until later. If you want to do this, do not carry out the following step:

8. Choose ***Activate***.

   When the application program is executed, the system carries out the code in the method you wrote.

Example

A Web user of *SAP Biller Direct* wants to view his own account (the customer assigned to him) and any other customers who are related to his customer.

Assume that a customer has been assigned to the Web user representing the head office of a buying group. When you call method GET_USAPPLREF, the table of object relationships already contains this customer. Within the method you can now determine all customers who have defined this customer as a buying group. When you add these customers to table T_OBJKEYS (with the appropriate company code), all customers are offered for selection to the head office user.

See also:

Methods

Determine Object References

Determine Texts for Object References

BAdI: Processing of User Requests

Use

This Business Add-In (BAdI) is used in component *SAP Biller Direct (FIN-FSCM-BD-AR)*.

In the Web application *SAP Biller Direct*, it is possible for a Web user to request a user online. Providing *SAP Biller Direct* has been appropriately configured, the system creates a general message via an Internet Service Request (ISR) with the data that the accounting clerk then checks and releases. When checking the user data, the accounting clerk assigns a customer and a default reference user to the user.

With this BAdI, you can make the processing of user requests easier for the accounting clerk in the following ways:

- You can assign the default reference user to a user automatically when you create the user.
- You can restrict the number of customers that is displayed in the activities list of a general message under *Check User Data*.
- You can change the confirmation of registration from e-mail to a printed version.

Standard settings

This business add-in is not active in the standard system.

Activities

After you call the IMG activity, the system displays a dialog box where you enter a name for the implementation.

If implementations of this Business Add-In have already been created, the system displays them in a dialog box. You then choose one of them by choosing ***Create***, and continue as follows:

1. In the dialog box, enter a name for the implementation of the Add-In and choose ***Create***. The system displays the initial screen for creating Business Add-In implementations.
2. On this screen, enter a short description for you implementation in the ***Implementation Short Text*** field.
3. If you choose the ***Interface*** tab, you will notice that the system has filled in the ***Name of the Implementing Class*** field automatically, by assigning a class name based on the name of your implementation.
4. Save your entries and assign the Add-In to a package.
5. To edit a method, double-click its name.
6. Enter your implementation code between the `method <Interface Name>~<Name of Method>.` and `endmethod.` statements.
7. Save and activate your code. Navigate back to the ***Change Implementation*** screen.
   Note: You can also create an implementation for an Add-In and not activate it until later. If you want to do this, do not carry out the following step:
8. Choose ***Activate***.
   When the application program is executed, the system carries out the code in the method you wrote.

BAdI: Notification Address

With this Business Add-In you can change some of the parameters of notifications per e-mail or SMS when a bill or invoice is received. This covers: the e-mail address of the sender and the settings for the bill recipient to be sent the notification (medium, time, choice of document header). This Add-In makes available information on the document header of the bill and the details of the message.

BAdI: Notification using SMS or E-mail

Use

You use this Business Add-in to create a text for a notification that you send either by e-mail or SMS. To do this you need to implement method CREATE_TEXT. The following import parameters are available:

- I_KNA1 Customer master (with the structure KNA1)
- I_VBDKR Document header view for billing (with the structure VBDKR)
- I_NAST Message status (with the structure NAST)
- I_LINK The URL that you defined in table T1URL for displaying the bill
- I_SFORM The Smart Form that you entered in the message type used should you want to send the notification as a PDF attachment
- I_SMS Displays whether an e-mail (= space) or an SMS (= X) is to be sent You must fill the following export parameters:
- E_TEXT Table containing the text to be output
- E_DOC Document attributes with structure SODOCCHGI1. Field OBJ_DESCR contains the reference line of the e-mail sent.
- E_DOC_TYPE Set this parameter to RAW, if you want to send an SMS or e-mail with unformatted text. Set the parameter to HTM if you want to send an e-mail in HTML format.
- RETURN Set this parameter to once the Business Add-In has run without errors.

Example

A straightforward example is delivered, and you can display this by choosing *Goto -> Example Coding -> Display*.

SAP Biller Direct for Bill Recipients

You only carry out the following IMG activities if you are using SAP Biller Direct as a bill recipient.

Define Invoice File Archiving

Use

You save bill files from SAP Biller Direct on your PC, and subsequently transfer these to your optical archive. Here you assign a storage system and document type to the file extension (such as PDF, XML). In addition you can also set whether the bill file is deleted by the PC once it has been archived.

The only file types that are archived are those that you have defined. If, for example, you have saved XML and PDF files on your PC, but only want to archive the PDF files, define just one entry here with file type PDF.

Requirements

- You are using an optical archive.
- You have made the settings to SAP ArchiveLink

Example

You want to archive a PDF file in the storage system with the *Identification* **A2** and the *Document type* **ZEBPPPDF**.

BAdI: Check of Bill Files

Use

You can use this BAdI to check and if necessary reject the bill files that have been uploaded in the bill receipt process of *SAP Biller Direct* (program REBPP_REC_INV, transaction EBPP_RI). You can also change the files, and transfer additional files to further processing.

Activities

Implement the method check_files. You need to call up the method for each bill once. All files that relate to this bill are transferred (for example, the XML file and the PDF file). Each line corresponds to a file in table T_FILE_INFO. Table T_FILES contains the data for all files. It is possible for you to change these tables. In this case set C_X_FILES_CHANGED to 'X'.

If you do not want to upload the bill files, set the return parameter C_X_NOT_OK to 'X'.

Integration with SAP Dispute Management

Use

If you make use of the integration of SAP Biller Directwith SAP Dispute Management carry out the following activities in Customizing in the accounts receivable accounting system under *SAP Dispute Management Process Integration*:

- Assign Case Types for Dispute Cases from SAP Biller Direct
- Define Reasons for SAP Biller Direct

If you want to use the option in SAP Biller Direct of appending attachments to dispute cases, then choose *Define Document Category for Attachments to Dispute Cases* in Customizing under *Financial Supply Chain Management -> Dispute Case Processing -> Process Integration*.

Billing Consolidation

General Settings

Define Basic Settings

Use

In this IMG activity, you define general basic settings for the *SAP BCONS Connector*.

- You can specify a work directory.
- You enter the port number of the file system where XML files are to be saved.
- You can choose how signed files are to be saved (with or without a signature).
- You can choose whether logging is to be activated.

Requirements

- You have activated EDX.
- You have created a port in the IMG activity Define Ports in IDoc Processing.

Activities

- In the *Work Directory* field, enter a directory on the *SAP ERP* back-end system for which you have read and write authorization.
- In the *Port for XML File* field, enter the port number that you specified in the IMG activity *Define Ports in IDoc Processing*.
- Choose a signature type from the *Signed File in Arch.* selection menu.
- Select the *Logging Active* checkbox if you want to activate logging. This is useful in the event of errors only.
- Select the *Unsigned PDF* checkbox if you want the system to archive an unsigned PDF in addition to a signed PDF.

Example

In the *Work Directory* field, enter **</usr/sap/E4A/SYS/global/access/>**.

In the *Port for XML File* field, enter **<IDOCXML>**.

In the *Sig.Typ* selection menu, choose the **PM** entry.

Define Business Partners

Use

In this IMG activity, you define business partners as senders or recipients. *SAP BCONS Connector* adds this data to messages. For each business partner, you can enter IDs and message types and specify whether he or she would like to receive message confirmations.

Activities

- In the *Company Code* selection menu, choose the company code for which the IDs are to apply.
- Choose an entry in the *Bus. Partner Type* selection menu (**KU** or **LI)**.
- Enter an asterisk (*****) in the *Customer / Vendor* input field.
- In the *Outgoing Message Tp* selection menu, choose **ALL All Message Types** as the type of outbound message.
- In the *ID at Service Provider* field, enter the PID that the business partner received from his service provider.
- In the *Service Provider ID* field, enter the service provider's PID.
- Select the *Rec. Mess. Confirm.* checkbox to send or receive message confirmations.
- Select the *Send PDF* checkbox so that the system also sends PDF files.
- Select the *Send ISF* checkbox so that the system sends ISF files.

Define Inbound Object Types

Use

In this IMG activity, you define inbound messages so that the SAP system knows how they are to be stored in the park table EDX_PARK and how they are to be archived.

Requirements

- You know the document types and formats used by your service provider.
- You know whether and how the inbound messages are signed.

Activities

- Choose the type of inbound document from the *Object Type* selection menu.
- In the *SP Message Name* field, enter the name of the document type configured by your service provider.
- In the *SP Document Format* field, enter the name configured by your service provider.
- Select the *SP Document Signed* checkbox to specify whether the inbound document is signed.
- If the message to be defined is a signature, choose the document type to which it belongs from the *Signature Assignment* selection menu.
- Choose the document type in the *SAP BCONS Connector* from the *Incoming Message Tp* selection menu.
- In the *Archive Object* field, enter the document type in the R/3 archive.
- In the *Doc. Format* field, enter the document format in the R/3 archive.

Example

You choose **1 Purchase Order (XML)** as the *object type*.

You enter **Order** as the *SP Message Name*.

You enter **XML-IDOC-OUT** as the *SP document format*.

You select the *SP Document Signed* checkbox.

You choose **No Document** as the *Signature Assignment*.

You choose **ORD Purchase Order** as the *Incoming Message Tp*.

You enter **EDXORDXMLS** as the *Archive Object* .

Enter **XMLS** as the *Doc. Format*.

Define Outbound Object Types

Use

In this IMG activity, you define outbound messages sent from *SAP BCONS Connector to a service provider (SP).*

Requirements

You know the document types used by the SP.

Activities

- Choose an EDX document type from the *Outgoing Message TP* selection menu.
- Enter the corresponding SP document type in the *SP Message Name* input field.
- Enter the EDX object type in the *Object Type* input field.
- Enter the EDX archive document type in the *Arch. Link Doc. Type* input field.
- Enter the EDX archive object type in the *Archive Link Obj. Typ* input field.

Example

- Choose **INV Invoice** as the *Outgoing Message TP*.
- Enter **Invoice** as the *SP Message Name*.
- Enter **VBRK** as the *Object Type*.
- Enter **SDOINVOICE** as the *Arch. Link Doc. Type*.
- Enter **VBRK** as the *Archive Link Obj. Typ*.

Define Transformations for Incoming Messages

Use

In this Customizing activity, you can specify an XSL transformation for incoming messages, if the format of the XML messages does not correspond to that of the XML IDoc.

This means various XML formats can be provided that can be converted into the correct XML format (the XML IDoc) before being entered in table *EDX: Inbound Messages (Messages Ready for Processing)* (EDX_PARK) and further processing in the connector.

The message format (purchase order, purchase order change, order confirmation, shipping notification, invoice) must be XML for this. The transformation is based on XSLT in version 1.

Requirements

1. You have created a customer-specific transformation in the customer namespace (Y or Z namespace), and this is entered in the repository.
2. The transformation is configured as appropriate.

Activities

3. Select the corresponding transformation in the table.
4. You can change the entries in the following fields:
   - *ID of Service Provider*: Description of the service provider:
   - *ID of Sender*: Description of the original sender of the message (partner ID).
   - *Message Description*: Description within the DocumentSet message for the incoming message from the service provider.
     Example:
     The service provider determines the message description *Invoice*, and is specified as a type for the document internally.
   - *Document Format*: Source format for the structured XML message.
     This is the description of the document format within the DocumentSet message before the transformation takes place.
   - *Transformation*: Customer-specific description of the XSL transformation. Enter the description of the XSL transformation here that you have already created (see 'Prerequisites').

Define Logical Port Names

Use

In this Customizing activity, you define logical port names for a Web service interface.

Instead of using middleware (such as SAP PI), you can also provide DocumentSet messages direct from the SAP back-end system to the service provider. You can also retrieve new, pending DocumentSet messages at the service provider direct.

There is a Web service interface available for this in the Billing Consolidation Connector solution.

Requirements

- You do not wish to use middleware (SAP PI) as a message router.
- The DocumentSet messages sent by the service provider for this customer are signed. The IDOc and the PDF that are in the signed DocumentSet message are signed, and the signature is verified at the same time. Consequently, a VerifyLog is also found in this DocumentSet message.

- No verification of signed DocumentSet messages is possible in the SAP back-end system.
- Verification is undertaken by the service provider. This means signed and verified DocumentSet messages are sent by the service provider.
- You register the SSL certificates in the SAP back-end system that you use to communicate with the service provider.
- You have created a corresponding port in the SOA Manager.

Activities

You have to construct communication secured by certificate for secure dispatch of DocumentSet messages to the service provider. You have to register an SSL communication certificate in the SAP back-end system for this purpose.

To do this, you proceed as follows:

1. Call transaction STRUST.
2. Create a new SSL client identity under *Environment -> SSL Client Identities*:
   a) Make all of the necessary entries, and select key length 124 or 248.
   b) Create a certificate request, and send the file for signing to your Certification Authority (CA).
   c) Import the signed certificate response for the SSL client identity.
3. Add your own CA using *Certificate -> Database -> Execute* to the list of trusted CAs.
   If your own CA is **not** available in the list, include it.
4. Link the newly created CA with the certificate.
   To do this, in the trust manager via *Certificate -> Import* create the link for the certificate file (*File* tab) with the CA inserted in the database (*Database* tab).

In addition, you have to make the following entries in table *EDX: Table: Service Provider, Logical Port* (EDX_SP_LP_NAME):

5. *ID of Service Provider*:
   Enter the ID of the service provider here. Note that only numbers are allowed.
   You can find an overview of the service provider IDs in the system in table *EDX: Business Partner/Service Provider/Reporting Details*(EDX_BP_SP).
6. *Name of Service Provider*:
   Enter a recognizable name for the service provider here.
7. *Name of Logical Port*:
   Name of the corresponding port that you have created in the SOA Manager.

Report *EDX: Send Messages* (EDX_SEND) is used to send the DocumentSet messages to the service provider. In table *EDX: Configuration (General)* (EDX_SETUP), select the middleware *Web Service*.

Environment

Define Partner Profiles

Use

In this IMG activity, you define the partner profiles for inbound and outbound messages so that the system knows where to send the message and what to do with messages that are received.

Requirements

- You have defined the port **IDOCXML** in the IMG activity Define Ports in IDoc Processing.
- You have defined the middleware that is to receive the outbound messages in the *Partner Type LS*.
- You have defined the message types used in your documents in the IMG activity Define Conditions for Message Control.
- Your system contains process codes that begin with EDX.

Activities

- Define outbound messages by opening the *Partner Type LS* folder and choosing your middleware.
- Under *Outbound parmtrs.*, choose *Create outbound parameter*.
- On the *Outbound Options* tab page, define the following points for each of the *message types* DESADV, INVOIC, ORDCHG, ORDERS, ORDERSP, and SYNCH: The *receiver port* **IDOCXML**, the *output mode* **Transfer IDoc immed.**, the value shown in table 1 as the *Basic type* and the value shown in table 2 on the *Message Control* tab page.

Table 1

Message type	Basic type
DESADV	DELVRY
INVOIC	INVOIC2
ORDCHG	ORDERS
ORDERS	ORDERS
ORDERSP	ORDER
SYNCH	SYNCHRON

Table 2

Message type	Application	Message type	Process code
DESADV	V2	LAVA	DELV
INVOIC	V3	RD	SD9
ORDCHG	EF	NEU	ME11
ORDERS	EF	NEU	ME1
ORDERSP	V1	BA	SD1
SYNCH	SYNCHRON	no	entry

- Define inbound messages from vendors by creating them in the *Partner Type LI* folder.

- Add the message types DESADV, INVOIC, and ORDRSP by choosing the *Create inbound parameter* pushbutton for each message type under *Inbound parmtrs.* and choosing the *process codes* shown in Table 3 for each message type.

Table3

Message type	Process code
DESADV	EDX_DELS
INVOIC	EDX_INVL / EDX_INVF
ORDRSP	EDX_ORDR

- Define inbound messages from customers by creating them in the *Partner Type KU* folder.
- Add the message types ORDCHG and ORDERS by choosing the *Create inbound parameter* pushbutton for each message type under *Inbound parmtrs.* and choosing the *process codes* shown in Table 4 for each message type.

Table 4

Message type	Process code
ORDCHG	EDX_ORDC
ORDERS	EDX_ORDE

Define Ports in IDoc Processing

Use

In this IMG activity, you define a port so that *SAP BCONS Connector* can save XML files to the file system.

Activities

1. Create a new entry in the *XML File* node.
2. Enter **IDOCXML** in the Port field.
3. Enter **EDX: Write IDoc as XML File** in the Description field.
4. Choose the *Unicode* radio button in the XML Format field group.
5. Choose the *Outbound File* tab page.
6. Choose the *Physical Directory* radio button.
7. Accept the path proposed in the *Directory* input field or enter a directory to which the back-end system can write files.
8. Choose the *Access Test* pushbutton to check whether the access works.
9. Enter **EDI_PATH_CREATE_CLIENT_DOCNUM** in the Function Module input field.

Example

The *Directory* input field contains the path **/usr/sap/E4A/SYS/global/**

Define Conditions for Message Control

Use

In this IMG activity, you define outputs for the system documents purchase order, order, delivery, and billing document so that the system can save them as IDoc files in the file system and as PDF files in the archive. The send report (EDX_SEND) sends IDocs saved in the file system and the associated PDF files from the archive to the middleware.

Activities

To enable outputs to be sent as IDoc files, use the existing (print/paper) outputs and adjust the condition records. Create the new output type ZPDF for PDF files.

- Purchase order / purchase order change from customer

1. IDoc

   a) Adjust the conditions by selecting the row with the application *EA Purchasing RFQ*, choose the *Condition Records* pushbutton, and choose the output type *NEU*. The system displays the *Key Combination* dialog box.

   b) Click the *Choose* pushbutton or press *Enter*.

   c) Choose *Execute* or press *F8*.

   d) Create a *condition record* with the *medium A Distribution (ALE)*.

2. PDF

   a) Create the new *output type* by copying your existing message type NEU to ZPDF.

   b) On the *Default Values* tab page, choose the entry *4 Send Immediately* in the *Dispatch Time* selection menu.

   c) On the *Storage System* tab page, choose the entry *2 Archive Only* from the *Storage Mode* selection menu.

   d) Save your entries.

   e) Create a *condition record* as described above with the *medium 1* and choose the *Communication* pushbutton.

   f) Define the *Output Device* as **LOCL** or **LP1**.

   g) Save your entries.

   h) Extend the *Structures* to include an entry for the *condition type* **ZPDF**.

- Purchase order confirmation

   i) Proceed as described above for the application *V1 Sales*, but adjust the output type *BA Order Confirmation* and create output type **ZPDF** with the description **PDF Order Conf.**

- Delivery

   j) Proceed as described above for the application *V2 Shipping*, but adjust the output type *LAVA Outg. Ship. Notifica.*, and create output type **ZPDF** with the description **PDF Shipping Notif.**

- Billing document

   k) Proceed as described above for the application *V3 Billing*, but adjust the output type *RD Invoice*, and create output type **ZPDF** with the description **PDF Billing Doc.**

Cash and Liquidity Management

General Settings

Market Data

You can transfer market data to the system in two ways:

- Datafeed: You have an exchange information system such as Reuters or Dow Jones and a relevant external partner program installed on your system with which the datafeed can communicate.
- File interface: You can store your market data in SAP format and transfer this to the system using a report program. Make sure you use the correct file format.
- This section describes all the settings which allow you to transfer market data to the system.

Master Data

Currencies

Check Currency Codes

The currency table must have entries for all currencies which occur in your business transactions.

Standard settings

In the SAP standard recipient system, all currencies are defined according to the international ISO standard.

SAP recommendation

SAP recommends that you use the ISO standard for your additional entries.

If your entries do not correspond to the ISO standard, you will not be able to use data exchange in international communication (e.g. bank clearing transactions).

Actions

1. Check the currency entries for completeness.
2. Add the missing currency entries as required.
3. Use the ISO standard for your additional entries.

Check Decimal Places for Currencies

Each currency has a different number of decimal places.

If you are adding new currencies, which do not have two decimal places, you have to enter these currencies in the menu option 'Decimal places'.

Example

Currencies with different decimal places

1. The German mark has two decimal places for amounts less than a mark.
2. The Kuwaiti dinar, on the other hand, has three decimal places.
3. The Italian lira does not allow any decimal places.

Standard Settings

In the settings delivered with the system, the most important currencies which do not have two decimal places are pre-set according to the ISO standard.

Activities

1. Check whether in your company currencies are in use which do not have two decimal places, in addition to those delivered.
2. Set the appropriate number of decimal places for these currencies.

Check Rate Types

Exchange rates for different purposes for the same date are defined in the system as exchange rate types (you cannot delete existing entries).

If you need to carry out currency translations between a number of different currencies, you can simplify exchange rate maintenance by entering a base currency for the exchange rate type. Instead of entering translation rates between every single currency, you then only need specify the translation rate between each currency and the base currency. All currency translations then take place in two steps - into the base currency and from the base currency into the target currency.

Example

The base currency is DEM. You want to translate FRF to CHF. To do this, the following entries must be made in the table for maintaining currency translation rates:

- Ratio for FRF -> DEM
- Ratio for CHF -> DEM

Translation from FRF to CHF is then carried out automatically. The translation is done as though this exchange rate (FRF -> CHF) was actually entered in the conversion table.
In this example, the base currency (DEM) is the To-currency and FRF or CHF the From-currency.

For exchange rate relations within the EU, it is a legal requirement that the base currency (EUR) is the From-currency. You set the indicator for this in the *BCurr=from* field.

Hinweis

When posting and clearing documents, the system uses the exchange rate type "M" for foreign currency translation. This exchange rate type must be contained in the system.

Standard settings

The standard system includes the exchange rate types for the bank buying rate (G), bank selling rate (B), and average rate (M).

You can have the system calculate the bank buying or selling rates from the average rate and the spread. The term spread refers to the difference between the average rate and the bank buying rate, or between the average rate and selling rate. You can find more information on this in the documentation on the activity Maintain spreads.

Activities

1. Find out which exchange rate types are needed in your company.
2. Check the standard exchange rate types. Create additional exchange rate types if necessary.
3. If you want to specify that all currency translations for a rate type must be carried out using a base currency, enter a currency (such as the group currency) in the *Base cur* field.
4. If you want to use the base currency specified as the From-currency, select the *BCurr=from* field.
5. If you want to have the system calculate the buying and selling rates from the average rate and the spread, enter the rate type for the average rate in the *Buy.rt.to* or *Sell.rt.to* field. Then maintain the spreads under the activity Maintain spreads.
6. If you want to use the inverted rate for translating two currencies, select the *Inv* field. **Note:** The reversed rate is used only if you have not made an entry for the corresponding exchange rate in the activity Enter exchange rates.
7. If you want to calculate the amounts according to the European Monetary Union's legal directives, select the *EMU* field.
8. If you want the system to check whether the application uses an exchange rate other than the fixed exchange rate, select the indicator in the *Fixed* field.

   This indicator must be set for the exchange rate type that is used for currency translation within the EMU.

Check Rate Spreads

In this activity you can derive the bank buying rate and/or bank selling rate using the average rate and the spread. By spread, we mean the difference between the average and buying rates or between average and selling rates.

You convert to the buying or selling rate by first determining the average rate, then adding or substracting the spread from that average rate.

Example

If the average rate between USD -> DEM is 16 and the spread is 4, the bank buying rate derived is 16.

To make that calculation you must make the following entries in this step:

M (rate type)   USD (From)   DEM (To)   4 (spread)

Requirements

In the "Check rate types" activity, you have to designate the "bank buying rate" and "bank selling rate" contained in the standard R/3 System as follows:

Specify that the buying and/or selling rates will be derived from the average rate.

Activities

Enter the spreads for the average rates of all relevant currency pairs.

Check Rounding Rules for Currencies

For the company code/currency combination for which payments are to be made not in the smallest denomination, but in a multiple of it, enter the currency unit (rounding unit) to which amounts are to be rounded.

This ensures that the amounts in this currency are always rounded to this unit (providing the amounts you enter manually are also rounded in line with your entry). The payment program evaluates your entries to determine the cash discount and rounds off the amount accordingly.

Example

For Switzerland, you decide that five centimes is to be the smallest denomination for payment transactions. For Swiss company codes and the Swiss franc, you therefore enter in the "Rounding unit" column. Cash discount amounts are then rounded off to this currency unit. In the case of Swiss franc amounts entered manually, that part of the amount after the decimal point must be divisible by five.

Standard settings

The currency unit has been specified for the standard Swiss company code and for the Swiss franc currency key.

Activities

For the required company code/currency key combinations, specify the currency unit to which the cash discount is to be rounded.

Check Exchange Ratios for Currency Translation

In this activity, you enter the translation ratios for currency translation. You enter these ratios for each exchange rate type and currency pair. You also specify whether you want to use an alternative exchange rate type for specific currency pairs.

Example

As of 1/1/1999 the exchange rate for DEM to FRF will be calculated via EUR. To translate amounts enter:

- **1/1/1999** in the *Valid from* field:
- The exchange rate type under which the exchange rates of the European national currencies are stored in the *Alt. ERT* field:

Requirements

- You must have defined the exchange rate types under which you want to define your translation ratios. You do this in the activity Check Exchange Rate Types
- You must have defined the required currency keys. You do this in the activity Check Currency Codes

Activities

1. Specify existing currency translation ratios for each exchange rate type for the currency pairs you need in your company.
2. Enter the exchange rate type to which you want to switch with specific currency pairs in the *Alt. ERT* field.

Further notes

The ratios you enter here for currency translation are displayed again when maintaining the exchange rates. For information on how to maintain exchange rates, see Enter Exchange Rates.

Manual Entry of Market Data

Enter Exchange Rates

Exchange rates are required to:

- Translate foreign currency amounts when posting or clearing or to check an exchange rate entered manually
- Determine the gain and loss from exchange rate differences
- Evaluate open items in foreign currency and the foreign currency balance sheet accounts The exchange rates are defined by period ("valid from").

Note

The function has been improved. Choose the activity Define Exchange Rate Input Worklist or Assign Exchange Rate to Worklist.

The system uses the type **M** exchange rates for foreign currency translation when posting and clearing documents in the activity *Enter Exchange Rate*. An entry must exist in the system for this exchange rate type. The exchange rates apply to all company codes.

The exchange rate relations for currency conversion are only displayed here. Read the chapter Define Translation Ratios for information about exchange rate relations maintenance.

Requirements

You define:

- the rate types under which you want to store your exchange rates in the step "Check Exchange Rate Type"
- the required currency keys in the step "Check Currency Codes"
- the conversion factors for
- the currency pair
- the exchange rate type
- the date
  in the step "Define the Conversion Factors for Currency Conversion"

Standard settings

Sample exchange rates are already stored in the standard system.

Recommendation

You should delete the exchange rates you do not need.

Activities

1. Check whether the exchange rates you need are in the system.
2. Delete the entries you do not need.

Define Market Data Monitoring

Definition of the tolerance limits for differences between exchange rates which are checked as market data is imported. Market data monitoring is activated by both the datafeed and the market data file interface. A warning message appears in the error log when there are differences in the exchange rates. The deviant rate is otherwise treated as a correct exchange rate. In other words, it is also saved in the R/3 market data table. Using the error log, however, you can process incorrect exchange rates manually at a later date.

Standard settings

The tolerance limit for all instrument classes is percent.

Activities

Define the maximum tolerance limit for differences in exchange rates for each instrument class (currencies, securities, ...).

Further notes

In Customizing of the translation table in datafeed, you can also define an individual rate tolerance limit which can deviate from the global setting defined here.

File Interface

In this step, you define the data provider for the file interface.

For each data provider, you can define the codes used by the data provider which correspond to the names used in the system (foreign exchange, securities, reference interest rates, and indexes).

Define Source of Market Data and Conversion Codes

In this IMG activity, you define the name for your data providers. This name is used for the automatic determination of the data provider from the file header of a market data file, provided that this information is defined there. To activate a conversion for a data provider, you need to set the conversion indicator for this data provider.

Requirements

To achieve an automatic conversion of the instrument names, the name of the data provider in the SAP TR system must be identical with the name of the data provider defined in the file header. If this is not the case, you can also specify the data provider when you import the market data.

Recommendation

If you get data from several different data providers, you should choose a separate name for each data provider. In this way you can guarantee that the quotations used by the data providers can be converted correctly in your SAP TR system.

Create Tables for Code Conversion

In this step you can generate the conversion code tables automatically with the names the SAP system uses for the following:

- Currencies
- Currency exchange rate types
- Exchanges
- Security price types
- Reference interest rates
- Indexes
- Index types

Requirements

Before you maintain the IMG activities for the conversion code tables you have to have maintained the relevant master data for currencies, securities, interest rates and indexes in the system. To do this read the requirements in the overview step Translation table. Conversion code tables are transferred. You also have to have maintained the data provider for the file interface.

Recommendation

Execute this step and transfer the master data you require and the data providers for the file interface to the conversion code table. Enter the relevant external names for the new entries in the conversion code tables.

Further notes

You can find further details in the relevant report documentation.

Currencies

In this unit, you maintain the conversion code tables for foreign exchange names and for exchange rate types.

Convert Codes for Currency Names

In this step you define how to map the foreign exchange names your data provider uses to the foreign exchange names used in the system.

Example

Notation:

- DEM

Data provider notation:

- 19

Requirements

In Customizing you have to define the:

- Currency symbols.

Recommendation

Ask your data provider which foreign exchange names he or she uses. You need this information to create conversion code tables for foreign exchange names individually.

Activities

Assign the name used by the data provider to each foreign exchange name in the system. Make sure that you get a 1 : 1 match for the names. There is no code conversion for this name unless you enter the relevant external name for a foreign exchange name in the SAP system.

Further notes

Using the IMG activity Create tables for code conversion you can automatically fill the conversion code table for foreign exchange names with the foreign exchange names used in the SAP system.

Convert Codes for Exchange Rate Types

In this step you define how to map the exchange rate types of the data provider to the exchange rate types used in the system.

Example

Notation:

- M

Data provider notation:

- MITTEL

Requirements

In Customizing you have to define the:

- Exchange rate types.

Recommendation

Ask your data provider which exchange rate types he or she uses. You need this information to create the conversion code tables for exchange rate types individually.

Activities

Assign the exchange rate type used by your data provider to each exchange rate type in the SAP system. Make sure that you get a 1 : 1 match for the names. There is no code conversion for this name unless you enter the relevant external name for an exchange rate type in the SAP system.

Further notes

Using the IMG activity Create tables for code conversion you can automatically fill the conversion code table for exchange rate types with the exchange rate types used in the SAP system.

Datafeed

In this chapter you define and maintain all settings to be able to put the datafeed into operation. These settings are split into four:

- Technical settings: Communication between the SAP system and the external partner program and settings for the results/error log
- Translation table: Definition of the desired market data, the external datafeed notation and the query and update attributes
- Planning market data retrieval: Specification of the market data scope and the time periods for market data retrieval
- Market data monitoring: Definition of tolerance limits for exchange rate deviations, which are checked when you import the market data. If there are exchange rate differences, the system issues a warning message in the error log. Market data monitoring is used both by datafeed and the market data file interface.

Procedure to Start Up the Datafeed

Procedure for start-up of datafeed

Requirements

This procedure describes the steps that must be carried out before you make your Customizing settings for SAP datafeed. It also covers who is responsible for these steps.

The knowledge of the following **people** is required to set up the datafeed:

- **Customer**: specifies his requirements.
- **Datafeed provider**: e.g. Dow Jones, Reuters. These supply the knowledge about the available data and the external interface program.
- **SAP basis administrator**: a person who knows how to install an SAP system or parts of it. This person must be able to install an SAP Standalone Gateway. This could be a customer employee or an external SAP consultant.
- **SAP application administrator**: A person who has Treasury- specific knowledge and can perform Customizing in the SAP system.

The following steps are carried out prior to or during datafeed Customizing. Note that the steps are partly based on one another.

Step 1: Datafeed provider installs the datafeed **(feed handling hardware, communications link)**.

Step 2: Install the **hard- and software of the datafeed provider**. In this procedure, we are assuming that, as is normally the case, the software of the datafeed provider is installed on a separate computer.

Step 3: Install **SAP Treasury Management** together with your SAP basis administrator. If you are reading this document in the system, this step has already been completed.

Step 4: In SAP Treasury Management, make the **Customizing** settings that are relevant for market data and for the customer. In particular, this includes all master data for currencies, securities, indexes, reference interest rates, forwards, and volatilities. The prerequisites are described in more detail in the "Translation table" chapter in the datafeed IMG documentation. Important: You do not perform the Customizing for step 4 in the "Datafeed" section of the IMG. In this step, you only make Customizing settings in other areas in order to fulfill the prerequisites for datafeed. Step 4 can be carried out in parallel to steps 1 and 2.

The market data range must be specified by the customer and/or SAP applications administrator. If the market data is to be fed into the system via datafeed, the availability of the market data must be discussed with the datafeed provider.
Step : Make sure the **TCP/IP-network link** between the computer with the datafeed provider's software and the SAP system works and runs smoothly. This is the customer's responsibility.

Step 6: Install the **interface program of the datafeed partner** on the separate computer, and ensure it is ready for operation.

Step : Install the **Standalone SAP Gateway** on the separate computer. This is the responsibility of the customer or the SAP Basis administrator. Note that this requires very specific knowledge! In some cases, you can do without the Standalone Gateway, but then settings must be carried out at operating system level. Ask your SAP administrator about this. In the following description, we assume that the Standalone Gateway has been installed.

Step 8: Make the **technical settings** in datafeed Customizing. You should do this directly after steps 6 and . This involves customers, SAP basis administrators, and datafeed providers.

Step 9 Maintain the **translation table** in datafeed Customizing. To do this, you must have completed step 4. The customer and the datafeed provider must agree upon the datafeed parameters in the translation table.

Step 1: Make the Customizing settings for market data monitoring, variants and jobs in order to supply the SAP system with market data on a regular basis. Customers and/or SAP application administrators are responsible for this step. Once you have completed step 1, you can go live with your datafeed.

Further notes

Steps 1, 2, 6, and are not necessary if you are connected to a WEB server. Ask your datafeed provider if he can deliver the data to you from his WEB server and if he is certified to do this. SAP provides both an RFC interface and a WEB interface and lets the datafeed providers decide how the data reaches the SAP system.

Note that the web interface can deliver the market data only synchronously and thus on request of the SAP system.

Technical Settings

In this step you maintain the technical settings for the datafeed. You define how the datafeed communicates with the external partner program. You can also define the printing parameters for the results/error log.

The activities described assume system management knowledge of the SAP system. You also need to know the external partner program of your data provider. You should already be familiar with the installation environment.

Define Datafeed Name

In this step, you define a name for your datafeed. You use this name to get queries from within the application. You also define whether you want to record an entry in a user log each time you access the datafeed.

Requirements

You must have installed the external partner program (e.g. Dow Jones Rate Feed Engine for system) before you can begin customizing the technical settings. You also have to determine whether you need to record a user log for the external datafeed provider. To do this, contact the external datafeed provider or read the documentation about the external partner program.

Standard settings

TELERATE for Dow Jones TTRS/RFE, user log is switched off.

Activities

1. Define the name for your datafeed.
2. In sample Customizing, the "Feed active?" field is set to 'not active' so that in a complete client copy is not a feed is not recognized as active in the productive clients if it does not exist.
   If you wish to use datafeed explicitly in the applications, you must mark them as being active.
3. Indicate whether you need a user log.
4. Save your entries.

Further notes

You have to choose a different name for each provider if you want to link up to datafeeds from different providers.

RFC Settings for External Partner Program

This unit contains all technical settings necessary for RFC-based datafeed. Ask your datafeed provider which settings are necessary here.

Define RFC Destination

In this step, you define all the logical links between the external partner program of the datafeed provider and your SAP system. These logical links are called RFC destinations.

Requirements

You have to have installed the external partner program (e.g. Dow Jones Rate Feed Engine for system) before you can begin customizing the technical settings. You must be able to access the computer upon which the external partner program runs from the SAP system by means of TCP/IP (network link is set up). You also have to install an SAP gateway for your SAP system on this computer. The SAP gateway (UNIX process gwrd) can either be installed as a standalone component or as part of a dialog instance. Read the installation guidelines for further details about this.

For Customizing you require the following information:

- The host name upon which the external partner program runs.
- The path and name of the external partner program (if necessary, ask your datafeed provider).
- The host upon which the SAP gateway runs.
- The name of the gateway service (sapgwXX; XX stands for the SAP system number of your live system).
- Does the external partner program support synchronous and transactional (asynchronous) links? If so, you have to set up at least two logical links (in this case, ask your datafeed provider).
- Likewise, if you require backup links, these also need to be defined. The datafeed then activates a backup link if the main link breaks down. Choose a different host for the backup link. Do not choose the one the SAP gateway runs on.

Activities

Create a new RFC destination via "Create":

1. Enter the name of the RFC destination.
2. Choose "T" as the connection type: start an external program via TCP/IP.
3. Press RETURN.
4. Describe your RFC destination (body text).
5. Choose the pushbutton "Explicit host" and enter the program path and name of the external partner program.
6. Choose *Destination -> Gateway options* and enter the gateway host and gateway service.
7. For transactional (asynchronous) links you have to maintain the TRFC options (this does not apply to synchronous links). Choose *Destination -> TRFC options* and enter the number of connection attempts to be made and the time between each attempt [mins].
8. Save your entries.

Example

Set up a synchronous and transactional RFC destination for one datafeed provider:

- Synchronous example:

a)	Destination:	RFC_DATAFEED_SYNC
b)	Connection type:	T
c)	Program:	/usr/aws/bin/rfc_datafeed_sync
d)	Target host:	xx1.your-domain
e)	Description: System	Synchronous datafeed link: Dow Jones Rate Feed Engine for
f)	Gateway:	xx1.your-domain
g)	Gateway service:	sapgw

- Asynchronous example:

a)	Destination:	RFC_DATAFEED_ASYNC

b) Connection type: T

c) Program: /usr/aws/bin/rfc_datafeed_async

d) Target host: xx1.your-domain

e) Description: Transactional datafeed link: Dow Jones Rate Feed Engine for System

f) Gateway: xx1.your-domain

g) Gateway service: sapgw

h) TRFC options: No. of connection attempts 6, Pause: 1 minute.

Further notes

When you set up backup links you have to set up two more RFC destinations. Make sure that you address a different SAP gateway, for example computer xx2.your-domain. It is also recommended that you install the external partner program on computer xx2.your-domain to protect your data in the event of a breakdown.

You configure the backup links as follows:

- Synchronous example:

a) Destination: RFC_DATAFEED_SYNC_BUP

b) Connection type: T

c) Program: /usr/aws/bin/rfc_datafeed_sync

d) Target host: xx2.your-domain

e) Description: Synchronous datafeed link: Dow Jones Rate Feed Engine for system backup link

f) Gateway: xx2.your-domain

g) Gateway service: sapgw

- Asynchronous example:

a) Destination: RFC_DATAFEED_ASYNC_BUP

b) Connection type: T

c) Program: /usr/aws/bin/rfc_datafeed_async

d) Target host: xx2.your-domain

e) Description: Transactional datafeed link: Dow Jones Rate Feed Engine for system backup link

f) Gateway: xx2.your-domain

g) Gateway service: sapgw

h) TRFC options: No. of connection attempts 6, Pause: 1 minute.

How many links do you have to define per datafeed?

- Only synchronous requests, no backup link: 1 RFC destination
- Only synchronous requests, backup link: 2 RFC destinations
- Synchr/asynchr. requests, no backup link: 2 RFC destinations - Synchr/asynchr. requests, backup links: 4 RFC destinations

Assign Datafeed RFC Destination

Define the modes (for synchronous/transactional links), the RFC destinations and the external functions for each datafeed. The external function is part of the external partner program.

Requirements

You have to have already maintained the RFC destinations. You also need the following information from your datafeed provider:

- Does the external partner program support synchronous/transactional modes?

- What is the external function of the external partner program called?

Standard settings Example:

Entries for Dow Jones -

Mode = 1 (synchronous):

a) Function: RATE_FEED_ENGINE_SYNC
b) RFC destination: NONE
c) Backup RFC destination: NONE

- **Mode = 2 (transactional):**

a) Function: RATE_FEED_ENGINE_ASYNC
b) RFC destination: NONE
c) Backup RFC destination: NONE

The above RFC destinations are not delivered with the system as they have to be Customized according to your system environment. Replace NONE with the RFC destinations defined by yourself.

Activities

For each mode (synchronous/transactional link), define the name of the external function, the RFC destination and, if applicable, the backup RFC destination (optional).

Define Users for Transactional RFC

In this step you define a user, which is used by the external partner program in transactional mode. If your external partner program does not support the transactional (asynchronous) mode, you can ignore this step.

Requirements

The external partner program must support a transactional mode. Ask your datafeed provider which user/password you have to define since the password is used by the external partner program and has be agreed upon. The external partner program needs the user/password to deliver the market data to the SAP system.

Standard settings

No user-related Customizing settings are delivered with the system.

Example

Settings for the TR_DATAFEED user:

a)	User:	TR_DATAFEED
b)	User category:	CPIC
c)	User group:	freely-definable
d)	Authorization profile:	F_DTFEED_ALL

Recommendation

For security reasons, we recommend that you only use the authorization profile F_DTFEED_ALL. You do not need an additional profile.

Activities

Define the user and the corresponding parameters. Choose CPIC as the user category.

Define Print Options for Log (Transactional RFC)

In this step, you define the print parameters for the error log when you are using a transactional link. The error log shows you whether the system managed to deliver all the market data or whether there were any problems. You can choose whether you want a log of all errors and correct results or whether you want only a log of errors or correct results.

Requirements

Your external partner program supports the transactional link and you have defined the user you need for this.

Standard settings

Customizing settings related to printing are not delivered with the system.

Example

You want to configure the print output for the TR_DATAFEED user so that the error log is output to the spool and stored there for 8 days. You only want to generate one error log.

a)	User name:	TR_DATAFEED
b)	Output device (printer):	P999 (to enable you to print)
c)	Copies:	1
d)	Print immediately:	(leave field empty)
e)	Delete immediately:	(leave field empty)
f)	Retention period:	8
g)	Log category:	1

Recommendation

There are two different procedures:

- The error/results log remains in the spool system for a certain period. From there, it is checked regularly on screen by the department responsible and only printed when required.
- The error/results log is printed directly in the department responsible; it is checked on paper.

Choose the procedure which best suits your organization. The example describes the first procedure.

Activities

1. Enter your user name.
2. Set the print parameters you require.
3. Save your entries.

Internet Settings for External Partner Program

HTTP Proxy Configuration

In the SAP System a HTTP client is integrated. This allows you to access Web servers from within the ABAP application, for example, to retrieve current share or currency rates.

If the Web server is accessed via the Internet, in most cases a proxy server is installed for safety reasons. You define the parameters of the proxy server here.

Define Proxy Configuration

In this step you specify the following configuration data for the HTTP proxy server:

- Host name and port. For example: proxy:88
- User name
- Password
- Function module   Default: HTTP_PROXY_CONFIG

You determine in the function module whether there is a Uniform Resource Identifier (URI see RFC 163) in the Intranet (access without proxy) or in the Internet (access via proxy).

You can use function module HTTP_PROXY_CONFIG as a template. You can adapt it to your network configuration and enter the new function module here.

Define Internet Settings for WEB Server Access

Here you find the settings for access to the partner's WEB server, especially the Universal Resource Indicator (URI), user and password. You can define a URI, user and password for each data provider.

Requirements

Note that this functionality can only be used if you connect to a WEB server of a certified SAP partner. The datafeed certificate must be explicitly valid for WEB server access. If not, SAP support cannot be provided.

Further notes

In the case of questions and problems, contact your datafeed provider. The datafeed provider will also give you your URI, user and password. The user and password are encrypted and not visible. Note that the proxy settings are a prerequisite for WEB-datafeed operation.

Workflow

In this unit, you find all settings in order to be able to use the workflow with technical errors or for transmission of market data logs.

Note that workflows for transmission of market data logs are only generated in batches. Workflows for technical errors are generated online and in batches.

Activate Workflow for Error Messages and Logs

Activities

Define what type of errors you want to use workflow for. You can generate workflows for technical errors and/or to output the market data log. You can also define the scope of the market data log.

Recommendation

It is recommended that the workflow is used for technical errors by sending them to the system administrator. The work item sent has priority class 1 so that the system administrator is directly informed provided he is working in the system at the time (Express work item).

You can, for example, send the market data log to the employee in the relevant department. You can only send the incorrect entries to the relevant employee, for example, while the complete list can be seen in the print spool.

Further notes

Read the workflow documentation.

Adjust/Copy Workflow Template

The procedure for implementing DATAFEED workflow is described in this section. Several steps are necessary for this.

Requirements

1. Copying of the SAP workflow template WS 283 with the name DATAFEED *(optional)*

   This is only necessary if you want to change the workflow template in a customer-specific manner. Otherwise, you can use the SAP workflow template.

   Procedure:

   a) Choose Tools -> Business workflow -> Development -> Definition tools -> Tasks / Task groups -> Copy.

   b) Enter the task type WS and the task 283.

   c) Choose Task -> Copy.

   d) The "Copy task" dialog box appears. Enter an identification code and a name (look at the space available for the name!).

2. Change standard role 232 *(mandatory)*

   The standard role describes the responsibilities for steps 39 (display market data) and 48 (list technical errors).
   Procedure:

   a) Choose Tools -> Business workflow -> Development -> Definition tools -> Standard roles -> Change.

   b) Enter standard role 232 and choose 'Change'.

   c) Choose Goto -> Responsibilities and create the following two responsibilities there:

      - Technical error log / DF Admin
      - Market data log / User department

   d) Change the responsibilities in such a way that the 'Variant of the flow' has value 1 in the technical error log / DF Admin and value 2 in the market data log / user department.

   e) To enter the agent, choose Edit -> Agent maintenance -> Create, after placing the cursor on the corresponding standard role. You can now enter the user as a complete position.

3. Activate event linkage *(mandatory)*

   As this workflow is triggered via an event, the event linkage of the workflow used by yourself must be activated.

   Procedure:

   a) Choose Tools -> Business workflow -> Development -> Definition tools -> Tasks / Task groups -> Change.

   b) Enter the customer task / SAP workflow template WS 283 and choose 'Change'.

c) Go to "Triggering events".

d) Choose Edit -> Event -> Activate / Deactivate linkage for object category BUS194 and event "PROTOCOL-TO BE_GENERATED".

e) Save the workflow task.

4. Activate Editor assignment *(mandatory)*

Standard task TS 22 (display market data).

a) Transaction PFTC

b) Enter and display standard task TS 22

c) Choose Additional data -> Agent assignment -> Maintain.

d) Activate the Agent assignment.

e) Repeat steps a)-d) for tasks TS 22 (list technical errors).

5. Activate workflow *(optional)*
Activation of workflow after copying and adjusting of workflow. This step is only required if step 1 was carried out and a (customer) workflow was created.

Translation Table

Information About Translation Table

In this unit, you can maintain all the price and rate information that you want to import using the datafeed. This involves mapping the market data notes to the datafeed notes in the external partner program (see example). The notes can be derived automatically from the existing master data in the system. You have to enter the datafeed notes and any additional information manually. Talk to your datafeed provider if you are not familiar with the relevant datafeed notes.

Example

Notes:

- Currency pair: USD EUR
- Rate type: B
- Currency factors: 1:1

Relevant datafeed notes (for Dow Jones):

- Instrument name: LXDEM
- Data source: QFRecord
- Instrument property: ASK
- External currency factors: 1:1

Requirements

Before you customize the translation table, you must maintain the relevant master data for currencies, securities, interest rates, and indexes in the system. These can be transferred to the translation table automatically.

For currencies, you have to define:

- the currency symbols, by choosing SAP NetWeaver -> General Settings ->Currencies -> *Check Currency Codes.*
- the permitted currency pairs and currency factors, by choosing SAP NetWeaver -> General Settings -> Currencies -> *Define Translation Ratios for Currency Translation.*
- the currency rate type, by choosing SAP NetWeaver -> General Settings -> Currencies -> *Check Exchange Rate Types*

For securities, you have to define:

- the relevant exchanges, by choosing Treasury and Risk Management -> Transaction Manager -> Securities -> Basic Settings -> *Define Exchange*.
- the relevant exchanges and the quotation currency in the system for the securities master data. You do this in the securities class data.

For interest rate master data, you have to define:

- the reference interest rate, by choosing Financial Accounting -> Accounts Receivable and Accounts Payable -> Business Transactions -> Interest Calculation -> Interest Calculation -> *Define Reference Interest Rates.*

For index master data, you have to define:

- the index, by choosing Treasury and Risk Management -> Basic Functions -> Master Data -> Indexes -> *Define Index*.
- the index type, by choosing Treasury and Risk Management -> Basic Functions -> Master Data -> Indexes -> *Define Index Types*.

Define Data Sources for Datafeed

In this step you define all the data sources which can be addressed from the system.

Example

Dow Jones:

- QFRecord	Standard data source for the TDQF feed
- PubRecords	Data source for the publisher mechanism

Reuters:

- IDN_SELECTFEED	Standard data source for the Reuters feed

Requirements

Ask your datafeed provider which data source the external partner program supports.

Activities

Identify all the data sources used by your external partner program that you want to address from the SAP system.

Define Datafeed Conversions

In this step you define all the conversions that are relevant for datafeed. You can convert all exchange rate types and currency symbols. For example, you can map a currency exchange rate type to the external instrument property.

When the conversion has been defined, the exchange rate type/currency symbols are converted as the translation table is automatically filled, and the instrument property in datafeed notation is set to default in the translation table. This saves you from having to maintain it manually.

Make sure you define the conversion correctly as later manual amendments are time-consuming. Ask your datafeed provider how your exchange rate types are described externally in the system, in other words, what the relevant instrument properties are called.

Create Tables for Code Conversion

In this step you define the conversion codes you wish to use for exchange rate types and currency symbols.

Requirements

All the exchange rate types and currency symbols to be converted have been defined in the system. For more information, refer to the documentation for the Translation table

Activities

Define the conversion codes for your datafeed. Once you have completed this IMG activity, all the conversion tables you have chosen are available for further processing.

Convert Codes for Exchange Rate Types

In this step, you define the conversion code for an exchange rate type/currency name.

Recommendation

First execute the Customizing step Create tables for code conversion. This automatically fills the conversion code tables with all exchange rate types and currency names defined in the system.

Activities

Define the external names for the exchange rate types/currency names you require. You might want to ask your datafeed provider which names (instrument properties) your external partner program recognizes.

Convert Codes for Currency Names

In this step, you define the conversion code for an exchange rate type/currency name.

Recommendation

First execute the Customizing step Create tables for code conversion. This automatically fills the conversion code tables with all exchange rate types and currency names defined in the system.

Activities

Define the external names for the exchange rate types/currency names you require. You might want to ask your datafeed provider which names (instrument properties) your external partner program recognizes.

Import R/3 Master Data

In this step you can set up the system to generate parts of the translation table automatically.

Requirements

Before you make the Customizing settings for the translation table, you must have maintained the relevant master data for currencies, securities, interest rates, and indexes in the system. To do this, see the requirements in the overview documentation for the translation table. The master data you have defined can be transferred to the translation table automatically. You must also have entered the data sources for the datafeed.

Activities

Perform this step and transfer the master data you wish to use to the translation table. Also transfer the conversion codes you have already defined and the data source of the datafeed.

In the subset area, you will find the 'Rate type' field. Via this field, you can restrict the transfer of master data to one or several rate types.

In the 'Other' area, you will find the 'Permanently save market data' box. Depending on this box, all entries in the translation table are automatically provided with the corresponding setting.

Further notes

In securities master data, only data without the '2' (old) status is transferred to the translation table.

You can find further information in the program documentation.

Define Currencies

In this step you define how to request exchange rates via the external partner program of your datafeed provider.

Example

Notation:

- Currency pair: CHF DEM
- Exchange rate type: B
- R/3 currency ratios: 1:1

Corresponding datafeed notation (example for Dow Jones, April 199):

- Instrument name: FCHFDEM
- Data source: QFRecord
- Instrument property: ASK
- External currency ratios: 1:1

Example of a market value delivered by the datafeed: 16 (1 CHF = 16 DEM)

Make sure you enter the external currency ratios which depend on the value delivered by the datafeed. Here is another example:

- Instrument name: LXCHFDEM
- Data source: QFRecord
- Instrument property: ASK
- External currency ratios: 1:1

Example of a market value delivered by the datafeed: 116 (1 * 1 CHF = 1 * 116 DEM)

Ask your datafeed provider which market values are delivered and which external currency ratios you need to define.

Requirements

In Customizing you have to define:

- currency symbols, via Global Settings -> Currencies -> *Check currency codes*
- currency pairs and ratios, via Global Settings -> Currencies -> *Define translation ratios for currency translation*
- exchange rate types, via Global Settings -> Currencies -> *Check exchange rate type*.

The "Realtime price/rate supply" function operates on the assumption that your external partner program can operate in a realtime environment. Ask your datafeed provider if your external partner program meets this requirement and if it can provide market data from the system after one request.

Activities

1. Define how you want to import your exchange rates via the external partner program and hence via the datafeed.
2. If you wish to display a rate history in the system, set the *Permanently save market data in R/3* indicator.
3. You can also establish whether rate provision takes place via requests from the system or realtime via the external partner program of your datafeed provider.
4. Save your entries.

Check Translation Table

In this step you can check the translation table and all the entries in the table.

Activities

Execute this step and check all/parts of the translation table.

Recommendation

Correct any errors, otherwise you will be unable to import this market data.

Market Data Procurement

In this section you define when and how much market data you can get via the datafeed. The activities described assume SAP knowledge about job management/batch processing. Specialist knowledge is also required when the market data you need is available via the datafeed/external partner program.

Define Variants

If you want to use your own variants for the datafeed report programs, you create them here. **Standard**

settings

The following variants are delivered as examples for report RFTBDF, which updates market data and, if required, the tables on a one-off basis:

-	SAP_ALL:	all market data
-	SAP_COMMODIT:	all commodities
-	SAP_CURRENCIES:	all currencies
-	SAP_CURR_VOLA:	all currency volatilities
-	SAP_FORWARDS:	all forwards
-	SAP_INDEX_VOLA:	all index volatilities
-	SAP_INDICES:	all indexes
-	SAP_INTRST_RTS:	all interest rates
-	SAP_INTR_VOLA:	all interest volatilities
-	SAP_STOCK_PRC:	all security prices
-	SAP_STOCK_VOLA:	all security volatilities

Activities

Create the variants you need for the datafeed report program.

The report programs for which it is worth creating variants are called:

-	RFTBDF:	Display market data
-	RFTBDF:	Update market data

- RFTBDF14: Initialize real-time rate/price delivery (as long as your external partner program supports this function). Normally it is only necessary to initialize real-time delivery when you start the whole SAP system/batch management, or when the scope of your real-time market data changes.

Further notes

The variants SAP delivers are defined for one transactional (asynchronous) request mode. If your external partner program does not support transactional mode, you have to create your own variants with synchronous request mode.

Schedule Jobs

The following describes how to schedule a job for background processing and how the job is to be executed:

1. Schedule the job.
   a) Identify the job.
   b) Specify a mail recipient (optional) for the spool requests generated by the job.
   c) Specify the start time or start condition for the job as well as whether the job is to be repeated automatically.
   d) Define the job steps.
   e) Save the complete job so that it can be transferred to the background processing system.
2. Release the job so that it can be executed. A job can only be executed after it has been released. This is also the case for jobs that have been scheduled for immediate processing. If you have administrator authorization or release authorization for end users, your job is automatically released during scheduling.

Note:

You can see more detailed documentation via *Help -> Application help from the selection screen of the report.*

Define Rating

Use

In this IMG activity you can define ratings. Ratings are part of the business partner's *credit standing data*.

You can store an internal, time-independent rating for a business partner from an institute providing credit standing information in the credit standing data.

Caution

If you have not used the *rating* field to date, then we recommend that you store business partner ratings on the *Ratings* tab page. The *rating, credit-rating institute, date of credit standing information*, and *additional information for credit standing information* will be transferred to *Ratings* in a subsequent release. You can use *Ratings* to store several time-dependent rating results.

Bank Account Management

Basic Settings

Define Number Ranges for Bank Account Technical IDs

Use

In this Customizing activity, you create number range intervals for bank account technical IDs. The system automatically assigns a technical ID to a bank account once it is created in the bank account master data.

Activities

To create a new number range interval, enter a number for the interval and specify a number range. To decide from which number interval the bank account technical IDs should be taken, specify the interval number in Customizing activity Define Settings for Bank Account Master Data.

Define Number Ranges for Change Requests

Use

In this Customizing activity, you create number range intervals for change requests used in Bank Account Management. The system automatically assigns a number to a change request once it is created in Bank Account Management.

Activities

To create a new number range interval, enter a number for the interval and specify a number range. To decide from which number interval the change request IDs should be taken, specify the interval number in Customizing activity Define Settings for Bank Account Master Data.

Define Settings for Bank Account Master Data

Use

In this Customizing activity, you define the following basic settings for the bank account master data:

- **Assign a number range interval for bank account technical IDs** To do so, you must have maintained number ranges in Customizing activity Define Number Ranges for Bank Account Technical IDs.

This setting is required for the migration of house bank accounts.

- **Assign a number range interval for change request IDs**

To do so, you must have maintained number ranges in Customizing activity Define Number Ranges for Change Request Numbers.

This setting is required for the migration of house bank accounts.

- **Define bank account types**

Define different types of accounts to suit different business purposes. Account types can be used as an analysis dimension in reporting and planning.

This setting is required for the migration of house bank accounts.

To add a new bank account type, choose ***New Entries*** and then specify the following:

- *Type*: Enter a unique type ID no longer than 1 characters.
- *Description*: Enter a short description for the account type, for example, current account.
- *Direction*: Specify whether the cash flow direction is incoming, outgoing, or can be either of the two directions.
- *Attribute*: Specifies whether the bank account is an operating account or a functional account.
- Operating accounts are bank accounts that are used for daily business transactions, such as receiving incoming payments and issuing outgoing payments. - Functional accounts are bank accounts that are used in other financial activities, such as loan, investment, and fund raising.

- **Define sensitive fields to be protected for changes**

Making changes to sensitive fields defined here will trigger workflow change requests if SAP Business Workflow is enabled.

For the *Signatories* section and the *Overdraft Limit* section, both changes to existing data and new entries are regarded as changes.

To specify a sensitive field, first select an object that contains the field:

- *Bank Account*

  This object contains fields on the *General Data* tab, for example, *Bank*, *Country*, *Company Code*, and so on.
- *Currency*

  This object contains the account currency field on the *General Data* tab.
- *Overdraft Limit*

  This object contains fields on the *Overdraft Limit* tab.
- *Signature*

  This object contains fields on the *Signatories* tab.
- *House Bank Account*

  This object contains fields on the *Connectivity Path* tab.

- **Define import methods for bank statements**

On the *Additional Data* tab for each bank account master record, users can select one of defined methods as the ***Import Method for End-of-Day Statements***, and one as the ***Import Method for Intra-Day Statements***.

- **Define signatory groups for payment approvals**

Define different groups of authorized signatories to suit different business purposes. To do so, you must have enabled the signatory function in Customizing activity Enable Signatory Control.

- **Define the signatory groups and the approval sequence for approval patterns** Define two types of approval patterns for payments: sequential and non-sequential.
- **Assign approval patterns to company codes and account types, and specify the approval pattern priority**

Activities

Define Approval Patterns

- In the *Define Signatory Groups* view, define a signatory group
- In the *Define Approval Patterns* view, define patterns for payment approvals:
- For a sequential pattern, the signatory groups defined in this pattern approve payments in a sequential order. You can assign up to four signatory groups for a sequential approval pattern.
- For non-sequential patterns, the signatory groups approve payments regardless of sequential order. You must assign at least two signatory groups for non-sequential patterns. You can create non-sequential approval patterns by specifying the approval sequence as **Non-Sequential.**

  You can then maintain the non-sequential approval patterns in the *Maintain Non-Sequential Approval Patterns* view.
- In the *Assign Approval Patterns* view, you can assign patterns to company codes and account types.

The assignment works as follows:

- If an approval pattern is assigned to a company code but not to any account type, then the pattern is applicable to all account types under that company code.
- If an approval pattern is assigned to an account type but not to any company code, then the pattern is applicable to the account type for all the company codes.
- If multiple approval patterns are assigned to an individual account type or company code, approval pattern priority can be defined to determine the sequence of approval patterns.

  The priority value has the highest priority.

Example

Example for defining a sequential approval pattern

Assume that you want to define signatory groups G1 and G2 for a sequential signatory pattern P1, and assign signatory pattern P1 to company code 1. You do the following:

1. You create signatory groups G1 and G2 in the *Define Signatory Groups* view.
2. You create the signatory pattern P1 in the *Define Approval Patterns* view, and specify the signatory sequence of **G1** as ***First Step***, the signatory sequence of G2 as ***Second Step***.
3. You assign signatory group G1 to company code 1 in the *Assign Approval Patterns* view.

Example for defining a non-sequential approval pattern

Assume that you want to define signatory groups G3 and G4 for a non-sequential signatory pattern P2, you do the following:

1. You create signatory groups G3 and G4 in the *Define Signatory Groups* view.

2. You create the signatory pattern P2 in the *Define Approval Patterns* view, and specify the approval sequence of P2 as ***Non-Sequential***. Then you select the entry and go to the *Maintain Non-Sequential Approval Patterns* view, and assign G3 and G4 to P2.

Examples for determining the approval pattern to use

- Approval patterns P1 and P2 are both assigned to company code 1 and account type Balance:
  If you assign priority value to signatory pattern P1, and priority value 1 to pattern P2, when a payment is made in a bank account of company code 1 with the account type of Balance, pattern P1 is applied for the approval process.

- Approval pattern P1 is assigned to company code 1 and account type Balance, approval pattern P2 is assigned to company code 1 without specifying the account type:
  When a payment is made in a bank account of company code 1 with the account type of Balance, pattern P1 is applied for the approval process.

- Approval pattern P1 is assigned to company code 1 without specifying the account type, approval pattern P2 is assigned to account type Balance without specifying the company code:
  When a payment is made in a bank account of company code 1 with the account type of Balance, pattern P1 is applied for the approval process.

Maintain the Event Type Linkage for Triggering Workflow Processes

Use

In this Customizing activity, you can check and maintain the event type linkage for workflow processes used in Bank Account Management.

The linkage is used to assign the predefined event ***Created*** (initialize a workflow process) in Business Object Repository (BOR) object ***FCLM_CR*** (predefined BOR object for Bank Account Management) to a receiver (it can be the SAP predefined workflow template WS4343 or the customer-defined workflow template), so that the system triggers workflow processes in Bank Account Management.

For more information about event type linkages, see the standard documentation.

Standard settings

SAP offers a standard linkage for triggering workflow processes used in Bank Account Management (workflow template WS4343), including:

- Workflow for opening bank accounts (creating new bank account master records)
- Workflow for changing bank accounts (changing existing bank account master records)
- Workflow for closing bank accounts (marking the bank account records as closed)
- Workflow for changing a signatory in multiple bank accounts
- Workflow for reviewing bank accounts

By default, the standard linkage for Bank Account Management is activated. The settings are as follows:

- Object Category: BOR Object Type

- Object Type: FCLM_CR
- Event: Created
- Receiver Type: WS4343 **(Customizable)**
- Receiver Call: Function Module **(Customizable)**
- Receiver Function Module: SWW_WI_CREATE_VIA_EVENT **(Customizable)**
- Event Delivery: Using tRFC (default) **(Customizable)**

Activities

You may use or change the event type linkage according to your business requirements.

Note:

Make sure there is only one activated linkage for event ***Created*** in BOR object ***FCLM_CR***. Otherwise, the system will not be able to trigger the defined workflow processes in Bank Account Management.

Define Responsibilities for Rules Used in Workflow Steps

Use

In this Customizing activity, you can define responsiblities for the workflow rules used in your workflow processes.

Requirements

1. You have defined the relevant rules in *Maintain Rule* (transaction PFAC).
2. You have assigned the rules to the corresponding workflow steps in *Workflow Builder* (transaction SWDD).

Activities

Define the responsiblities for each rule.

SAP predefines the following rules for Bank Account Management:

- 436: Cash Manager
- 43: Bank Accountant
- 438: Key User

Enable Signatory Control

Use

In this Customizing activity, you enable the payment signatory control in Bank Account Management by assigning the required function modules.

When signatory control is enabled, users defined as signatories are able to approve or reject payment documents in Bank Account Management. The signatory information maintained in the bank account master data does not take effect until the signatory control is enabled here.

If the signatory control is not enabled, payment approvals are handled using the logic in the Bank Communication Management component.

Note

In the following exceptional cases, the system may fail to find any eligible signatory for a payment request and reports an error message. You can contact your administrator to reassign the request.

- The payment amount exceeds the maximum amount defined with the signatories
- Signatories are inactive
- Payment approval pattern cannot be determined

Note

If an automatic approval process has been defined for small payments in the Bank Communication Management component, small payments will be approved automatically regardless the signatory settings you defined in Bank Account Management.

Requirements

You have maintained the folllowing settings in Bank Communication Management:

- In the Customizing activity Rule Maintenance, define rules according to your business requirements.
- In the Customizing activity Additional Criteria for Payment Grouping, specify **HKTID** (house bank account) in the *Grpng. Field 1* field for defined rules.

Activities

To enable the signatory control in Bank Account Management, add the following two entries to the table as below:

- Process BANK2, Function Module FCLM_BAM_BCM_AGT_PRESEL, Product INGA
- Process BANK4, Function Module FCLM_BAM_BCM_REL_PROC_CTRL, Product INGA

Business Add-Ins (BAdIs)

BAdI: Field Statuses and Field Checks

Use

The Business Add-In (BAdI) is used in the Cash and Liquidity Management (FIN-FSCM-CM) component.

You can use this BAdI to do the following:

- Define field status for fields used in bank account master data. You can define the following statuses:
- *Read-only*
- *Changeable*

- *Mandatory*
- *Hidden*

  If a field's status is not defined by the BAdI implementation, the field is set as *Hidden* by default. For the following tabs, when all its fields are hidden, the entire tab is hidden:

- *Payment Signatories*
- *Overdraft Limits*
- *Connectivity Path*
- Define additional field checks for bank account master data.

You can create your own implementation by using the following methods:

- ADDITIONAL_CHECK: Defines additional field checks that are not defined in the standard system for bank account master data.
- NOTE_CONTROL: Defines whether it is mandatory or not for users to enter the *Note* field in the popup screen.
- BAM_FIELDS_CONTROL: Defines field statuses for bank account master data.
- REQ_FIELDS_CONTROL: Defines field statuses for bank account master data in workflow process.

Standard settings

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

Activities

For information about implementing BAdIs as part of the Enhancement Concept, see SAP Library for SAP NetWeaver under BAdIs - Embedding in the Enhancement Framework.

BAdI Implementation

SAP provides a default implementation FCLM_BAM_FIELDS_CTRL. The implementation defines the following:

- Default field statuses for bank account master data.
- Bank control key field status would be set to mandatory when the bank country is Brazil, France, Spain, Portugal, and Italy.

BAdI: Bank Account Number Mapping Between BAM and HBA

Use

The Business Add-In (BAdI) is used in the Cash and Liquidity Management (FIN-FSCM-CLM) component. You can use this BAdI to define the mapping logic of bank account numbers from Bank Account Management (BAM) in Cash and Liquidity Management (FIN-FSCM-CLM) to house bank account (HBA) data in Bank Master Data (FI-BL-MD), and another way round. The mapping logic influences the following activities:

- Using program Generate Bank Account Master Data from House Bank Accounts (transaction FCLM_BAM_MIGRATION) to generate bank account master records from existing house bank account data.

When executing this program, by default, the system copies the value of the *Bank Account Number* field from house bank accounts to Bank Account Management as the bank account number of the new bank account master record. To override the default logic, you can use this BAdI to define another method of deriving bank account numbers.

- Assigning house bank accounts in bank account master data

  When a new bank account master record is created in Bank Account Management, house bank accounts must be assigned. When assigning the house bank accounts, the system provides a F4 search help listing all the house bank accounts that contain the same bank account number as the bank account master record. By default, the system maps the first 18 digits of the *Bank Account Number* field in bank account master record onto the *Bank Account Number* field in house bank accounts. To override the logic, you can use this BAdI to determine which field to map to in the house bank account data.

You can create your own implementation by using the following methods:

- CONVERT_TO_AMD_ACC_NUM: convert bank account numbers from house bank accounts to bank account master data.
- CONVERT_TO_HBA_ACC_NUM: map bank account numbers from Bank Account Management onto house bank accounts.

Standard settings

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

Activities

For information about implementing BAdIs as part of the Enhancement Concept, see SAP Library for SAP NetWeaver under BAdIs - Embedding in the Enhancement Framework.

Example

SAP provides an example implementation FCLM_BAM_ACC_NUM_MAP_IMPL for your reference. The example implementation defines the mapping logic as the following:

- With method CONVERT_TO_AMD_ACC_NUM, the implementation combines the *Bank Account Number* field with the *Alternative Bank Acct Number* field in house bank accounts. The combined value is used as the bank accounts number for the new bank account master record.
- With method CONVERT_TO_HBA_ACC_NUM, the implementation maps the *Bank Account Number* field in bank account master record onto the *Alternative Bank Acct Number* field in the house bank account data.

BAdI: Events After Bank Account Activation

Use

The Business Add-In (BAdI) is used in the Cash and Liquidity Management (FIN-FSCM-CLM) component. The BAdI is called after a bank account master record is activated in Bank Account Management. The system activates bank account master records automatically when:

- A new bank account master record is submitted and approved.

- An exisiting bank account master record is changed and approved.

With BAdI method ON_AFTER_ACTIVATE, you can define events that you want to trigger after a bank account master record is activated. The new bank account data can be accessed in this BAdI. To view the original bank account data beofore activation, call function module FCLM_BAM_AMD_READ.

Standard settings

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

BAdI Implementation

- FCLM_BAM_AMD_ACTIVATION_IMPL
  The default implementation defines that when a bank account is activated, the system automatically sends an e-mail from the person who activated the account to the person who created the account master record.

Activities

For information about implementing BAdIs as part of the Enhancement Concept, see SAP Library for SAP NetWeaver under BAdIs - Embedding in the Enhancement Framework.

BAdI: Bank Account Master Data Fields in Change Requests

Use

The Business Add-In (BAdI) is used in the Cash and Liquidity Management (FIN-FSCM-CLM) component.

With BAdI method SAVE, you can define which fields in bank account master data should be recorded in change requests when their values are changed. Changes made to fields included in change request can be tracked by change history.

Standard settings

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

Activities

For information about implementing BAdIs as part of the Enhancement Concept, see SAP Library for SAP NetWeaver under BAdIs - Embedding in the Enhancement Framework.

BAdI: Payment Approval Pattern Determination

Use

The Business Add-In (BAdI) is used in the Cash and Liquidity Management (FIN-FSCM-CLM) component. You can use this BAdI to replace the payment approval patterns defined in Customizing activity Define Settings for Bank Account Master Data with your own logic and definition.

You can create your own implementation by using the following methods:

- DETERMINE_PATTERN: Override the default approval pattern determination logic with your own logic.
- GET_PATTERN_DEFINITION: Get the approval pattern definition you defined.

Requirements

You have enabled the integration between Bank Communication Management and Bank Account Management in Customizing activity Enable Signatory Control.

Standard settings

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

There is no activated BAdI implementation in the standard system.

Activities

For information about implementing BAdIs as part of the Enhancement Concept, see SAP Library for SAP NetWeaver under BAdIs - Embedding in the Enhancement Framework.

Cash Management

Planning Levels and Planning Groups

Define Planning Levels

In this step, you specify a planning level for each cash account and allocate this planning level to the cash position by using the appropriate source symbol.

Financial transactions in Cash Management are displayed using planning levels in order to explain beginning and ending account balances.

In the standard R/3 System, levels starting with "F" or "B" are reserved for automatically updating data during posting so that you can easily analyze the cash position. Manually created payment advice notes or planned items, however, are allocated to different levels.

Recommendation

SAP recommends using the following levels:

- Level F for bank accounts
- Level F1 for customers and vendors
- Levels B1 to Bn for bank clearing accounts
- Level CP, for example, for confirmed payment advice notes
- Level UP, for example, for unconfirmed payment advice notes
- Level NI, for example, for noted items

When displaying the cash position, you can then see because of level F that the amounts are posted to bank accounts. The other levels reflect planned bank account transactions, which include postings to bank clearing accounts or entered payment advice notes. Therefore, you can use the display to compare planned data with actual data.

Actions

Check the standard planning levels and change them if necessary.

Define Planning Groups

In this step, you define the planning groups for customers and vendors. A planning group represents particular characteristics, behaviors or risks of the customer or vendor group. Therefore, you can break down incoming and outgoing payments according to the amount, the probability of the cash inflow or outflow, and the type of business relationship.

Example

- O1 Domestic vendors
- O2 Foreign vendors
- O3 Affiliated companies: vendors
- O4 Major suppliers
- I1 Bank collection: customers
- I2 Domestic customers
- I3 Foreign customers
- I4 Risk customers

You assign customers and vendors to a planning group by making a entry in the master record.

In addition, you allocate your planning groups to a planning level.

Actions

1. Create planning groups according to your requirements.
2. Allocate the planning groups to a planning level you have created for your subledger accounts.

Define planning levels for Logistics

In this step, you allocate the transactions from Materials Management (MM) and Sales and Distribution (SD) to the planning levels you have defined for updating this data.

This allocation is necessary since the system cannot determine the planning level by using the master record fields, as it does to access data from Financial Accounting (FI).

In FI, the planning level for G/L accounts is specified in the master record. The specification for subledger accounts is made using the planning group with which the level for FI postings can be determined. However, a business transaction in Logistics is represented by an internal ID:

- 1 = purchase requisition
- 2 = purchase order
- 3 = scheduling agreement
- 11 = order

You must assign these transactions to a planning level so that you can distinguish them in Cash Management.

Actions

Allocate a planning level you defined before for MM and SD to each of these business transactions.

Special G/L Transaction Levels

Customer

Define Down Payment Request

In this step you can freely choose the planning levels for the down payment requests you want to use for controlling in Cash Management. In doing this, you must, however, take note of specific naming conventions.

Use levels beginning with F, since these are FI postings. The special G/L indicator is useful for the second letter, for example, FF for down payment request.

Activities

Enter the appropriate value in the ***planning level*** field.

Further notes

For more information see the Accounting Implementation Guide.

Define Bill of Exchange

In this step you can freely choose the planning levels for the bill of exchange payments from your customers which you want to use for controlling in Cash Management. In doing this, you must, however, take note of specific naming conventions.

Use levels beginning with F, since these are FI postings. The special G/L indicator is useful for the second letter, for example, FF for down payment request.

Activities

Enter the appropriate value in the ***Planning level*** field.

Further notes

For more information see the Accounting Implementation Guide.

Define Other

In this step you can freely choose the planning levels, which you want to use for controlling in Cash Management, for posting to a customer account. In doing this, you must, however, take note of specific naming conventions.

Use levels beginning with F, since these are FI postings. The special G/L indicator is useful for the second letter, for example, FF for down payment request.

Activities

Enter the appropriate value in the ***Planning level*** field.

Further notes

For more information see the Accounting Implementation Guide.

Vendor

Define Down Payment Request

In this step you can freely choose the planning levels, which you want to use for controlling in Cash Management, for vendor down payments. In doing this, you must, however, take note of specific naming conventions.

Use levels beginning with F, since these are FI postings. The special G/L indicator is useful for the second letter, for example, FF for down payment request.

Activities

Enter the appropriate value in the ***Planning level*** field.

Further notes

For more information see the Accounting Implementation Guide.

Define Bill of Exchange

In this step you can freely choose the planning levels, which you want to use for controlling in Cash Management, for bill of exchange payments in the general ledger. In doing this, you must, however, take note of specific naming conventions.

Use levels beginning with F, since these are FI postings. The special G/L indicator is useful for the second letter, for example, FF for down payment request.

Activities

Enter the appropriate value in the ***Planning level*** field.

Further notes

For more information see the Accounting Implementation Guide.

Define Other

In this step you can freely choose the planning levels, which you want to use for controlling in Cash Management, for posting special G/L transactions. In doing this, you must, however, take note of specific naming conventions.
Use levels beginning with F, since these are FI postings. The special G/L indicator is useful for the second letter, for example, FF for down payment request.

Activities

Enter the appropriate value in the ***Planning level*** field.

Further notes

For more information see the Accounting Implementation Guide.

Liquidity Items

Edit Liquidity Items

Use

In this Customizing activity, you can create and change liquidity items. You create liquidity items to represent the source and use of cash flows in your company. It can then be used as a dimension for financial planning and reporting in Cash and Liquidity Management (FIN-FSCM-CLM).

Activities

To create a liquidity item, you must define the following:

- Liquidity Item Key

- Liquidity Item Name
- Liquidity Item Description
- Cash Flow Direction

  Cash flow direction denotes whether the cash amount carried in the liquidity item is inflow or outflow. It is essential for cash flow calculation in all the reports and planning applications in Cash and Liquidity Management. For example, you create a liquidity item **Vendor 1** to present the cash flows related to the outgoing payments to one of your vendors, you should specify the cash flow direction as **Outflow**.

Further notes

To use the *Develop Liquidity Plans* app, define the following liquidity items as appropriate:

- If you want to integrate with cash position to get bank balance data, define a liquidity item with the key as **LP_CASHOP**.
- If you want to use the currency exchange function, define the following two liquidity items:
- **LP_EXF**: Cash converted from another currency -**LP_EXI**: Cash converted to another currency

Define Liquidity Item Hierarchies

Use

In this Customizing activity, you can create, change, transport, and display hierarchy structures for liquidity items. The structures will be reflected in cash and liquidity related reports and functions. You can define different structures for different entities, periods, or purposes, according to your business needs.

Requirements

You have created and defined liquidity items in the Customizing activity Edit Liquidity Items.

Activities

With this Customizing activity, you are able to:

- Create and define liquidity item hierarchies.

  To view the newly created hierarchy in the hierarchy list, refresh the page first.
- Split an existing hierarchy into two by dividing the original validity period. Use the *Split Validity Period* function to shorten the validity period of the original hierarchy and create a new hierarchy with a identical liquidity item structure and with the rest of the validity period.

  The new hierarchy created is with the same *Hierarchy Name* as the original hierarchy.

  You can enter a description to differentiate the new hierarchy from the original one.

  Example :

  You are using liquidity item hierarchy Cash Reports (valid from January 1, 21 to December 1, 22) for reporting purpose. Starting from January 1, 214, you will be using a slightly different hierarchy for this purpose.

To use the *Split Validity Period* function for hierarchy Cash Reports, you perform the following actions:

a) Enter the new *Valid From* date as **January 1, 214**, and the new description as **214**.

- The original hierarchy is now valid from January 1, 21 to December 31, 213. - A new hierarchy is created (with the *Description* as 214) , and it is valid from January 1, 214 to December 31, 22.

b) You change the new hierarchy structure accordingly.

- Transport liquidity item hierarchies to other clients.

  To make your settings available in other clients, you need to manually transfer the hierarchies. If there are hierarchies that are with the same name, you must transport them all together in one transport request, because these hierarchies origin from the same hierarchy and were created using the *Split Validity Period* function.

- Export the hierarchy list or individual hierarchy structures into spreadsheets.

- Deactivate obsolete hierarchies.

  You can deactivate a hierarchy if you no longer use it. Once it is deactivated, you can no longer change the hierarchy itself, but it does not affect the hierarchy display in reports and other functions for the data within its validity period.

Derivation Rules for Liquidity Items

Define Queries for Liquidity Item Derivation

Use

In this Customizing activity, you can create, edit, and delete queries for liquidity item derivation.

After a query is created, you can assign the query to query sequences by using Customizing activity Assign Queries in Query Sequences (transaction FLQQA).

Note:

The configuration in this Customizing activity cannot be transported to other clients. You must make the configuration in the production client.

Requirements

You have defined liquidity items in Customizing activity Edit Liquidity Items.

Activities

If the query is only intended for a certain paying company code, you can restrict the company code when creating the query.
Define a query sequence with one of the following origin types:

- Origin C: Applies to line items with account type (`BSEG-KOART`) of D (customer) or K (vendor)

- Origin D: Applies to line items with account type (`BSEG-KOART`) other than D (customer) or K (vendor)

- Origin X: One Exposure from Operations (except accounting documents)

Define Query Sequences

Use

In this Customizing activity, you can define query sequences based on existing queries.

Requirements

You have defined the relevant queries in Define Queries for Liquidity Item Derivation.

Activities

Line Items with Account Type (`BSEG-KOART`) of D (customer) or K (vendor):

The system automatically assigns the source symbol C.

Line Items with Account Type (`BSEG-KOART`) other than D (customer) or K (vendor):

The system automatically assigns the source symbol D.

Data from the One Exposure from Operations Hub (Except Accounting Documents):

For data from the One Exposure from Operations hub, for example, transactions from TRM, MM, and so on, the system automatically assigns the source symbol X.

Note: For accounting documents, origin C or D applies.

Assign Queries to Query Sequences

Use

In this Customizing activity, you can assign queries to query sequences.

Note:

The configuration in this Customizing activity cannot be transported to other clients. You must upload the configuration to the production client or make the configuration directly in the production client.

Requirements

You have defined the relevant queries in Define Queries for Liquidity Item Derivation (transaction FLQQA1) and query sequences in Define Query Sequences (transaction FLQC1).

Define Liquidity Item Derivation Settings for Company Codes

Use

In this Customizing activity, you can define the default derivation logic for liquidity items for company codes that have Cash Management activated. Each entry assigns a query sequence and a derivation function to a company code, which the system uses to derive liquidity items for that company code. For example, liquidity items can be determined based on the accounts referenced by the line items in documents posted to ledgers (G/L, accounts receivable, or accounts payable).

Note: The derivation rules that you define here only work for new postings. To generate liquidity items against existing posting data, you must run the rebuild program.

Requirements

- You have activated Cash Management in the global parameters of these company codes.
- You have defined the queries you need in Define Queries for Liquidity Item Derivation (transaction FLQQA1).
- You have defined the query sequences referenced here in Customizing activity Define Query Sequences (transaction code FLQC1).
- You have assigned queries to the query sequences referenced here in Assign Queries to Query Sequences (transaction code FLQQA).

Standard settings

You must create an entry for each company code that has Cash Management activated. Additionally, you can create an entry with the company code empty, as the default derivation rule for all company codes that have Cash Management activated but do not have corresponding entries defined here.

Activities

Define a query sequence with one of the following origin types:

- *From Clearing Information* (C)
- *From Invoices* (D)
- *One Exposure from Operations* (X)

Define Default Liquidity Items for G/L Accounts

Use

In this customizing activity, you can define a default liquidity item for each G/L account. Liquidity items are derived according to the customer-configured query sequence. If the query sequence fails to determine a liquidity item, default liquidity item defined here is used and recorded in accounting document line item table.

Requirements

You have defined liquidity items in Customizing activity Edit Liquidity Items.

Note:

The configuration in this Customizing activity cannot be transported to other clients. You must upload the configuration to the production client or make the configuration directly in the production client.

Example

You have made the following configuration:

- Assign query sequence 1 to company code 1
- Define EXP1 as the default liquidity item for G/L account 4 in company code 1 The result of the derivation can be one of the following:

- If query sequence 1 derives a liquidity item for G/L account 4, for example, liquidity item EXP2, the system takes this liquidity item as the result.
- If query sequence 1 fails to derive a liquidity item for G/L account 4, the system takes liquidity item EXP1 (default liquidity item for this G/L account) as the result.

Regenerate Condition Strings

Use

Using this Customizing activity, you can regenerate condition strings for queries in the Cash and Liquidity Management (FIN-FSCM-CLM) component.

When a query is created and saved by using Customizing activity Edit Queries for Liquidity Item Derivation (transaction FLQQA1), the query conditions are saved to table FLQQRRG as raw information. The system then automatically generates condition strings based on the raw information and stores the condition strings into table FLQQRCOND. The use of condition strings instead of the traditional query conditions improves the performance of query matching logic in Cash and Liquidity Management.

You can use this Customizing activity for the following purposes:

- Generate condition strings for old queries to improve query performance For customers who migrated to SAP S/4HANA Finance, we recommend to execute this program to generate condition strings for old queries, in order to benefit from the new logic of condition strings.
- Correct query condition inconsistencies between table FLQQRRG and FLQQRCOND
  By executing this program, you can regenerate the condition strings based on the original condition information stored in table FLQQRRG and overwrite the records in table FLQQRCOND.

Activities

Execute the program. It generates the number of checked queries, the number of changed condition strings, and a detail list of the changed condition strings.

You can perform a test run if you do not want to update the results to the database.

Flow Types

Define Flow Types

Use

In this Customizing activity, you can **optionally** define additional flow types (see Flow Type).

In addition, you can do the following:

- Change the flow type description if the standard description does not match the terms you use
- Assign an existing flow type to a different flow category.

Requirements

Each flow type must be assigned to an SAP-defined flow category.

Assign Planning Levels to Flow Types

Use

In this Customizing activity, you can assign planning levels to flow types.

When uploading bank cash balances into the One Exposure from Operations hub using the transaction FQM21 (Import Bank Cash Balances), the following flow types are assigned to the transactions generated:

- 912 = bank cash balance increases BALANCE_AMOUNT >
- 913 = bank cash balance decreases BALANCE_AMOUNT <

If you want to differentiate according to planning level as well when evaluating a cash position, you have to assign the relevant planning levels to these two flow types.

Assign Flow Types to Planning Levels

Use

In this Customizing activity, you can assign flow types to planning levels.

When using distributed cash management, in the central system the following flow types are assigned to the transactions generated:

- 911 - Incoming Cash (IDoc) when amount >
- 9111 - Outgoing Cash (IDoc) when amount <

If you want to differentiate confirmed cash when evaluating the cash position, you have to assign different flow types to those planning levels that represent confirmed cash. The flow types '918 - Cash Balance Increase (IDoc)' and '919 - Cash Balance Decrease (IDoc)' are available for this purpose.

Example

For planning level 'F - Posting to bank account', you assign the flow types '918 - Cash Balance Increase (IDoc)' and '919 - Cash Balance Decrease (IDoc)'. The uploaded data from distributed cash that is mapped to planning level F is then shown as confirmed cash in the cash position.

Assign Flow Types to G/L Accounts

Use

You should use this Customizing activity if you want to process additional accounting document items that the system does not automatically classify into one of the categories mentioned below.

In general, the integration of Financial Operations into One Exposure from Operations covers information on invoices, payments, and bank statements.

Within this integration, the system classifies accounting document line items into the following categories:

- Line item posting on a customer or vendor account (Receivables or Payables)
- Line item posting on a bank clearing account (Cash in Transit)
- Line item posting on a bank account (Cash)

The determination of the category is done by the system in the following way:

- Receivable or Payable

Items having account type K (Vendor) or D (Customer)

- Cash in Transit

Items having a G/L account that has one of the following characteristics:

- Maintained in account determination of payment program (T42I), is a balance sheet account (SKA1-XBILK), and is not a reconciliation account (SKB1-MITKZ)
- Maintained in account determination of bank-to-bank transfer (T42Y), is a balance sheet account (SKA1-XBILK), and is not a reconciliation account (SKB1-MITKZ)
- Marked as cash relevant (SKB1-XGKON) and with open item management (SKB1-XOPVW)
- Cash

Items having a G/L account that has one of the following characteristics:

- Maintained in house bank accounts (T12K)
- Marked as cash relevant (SKB1-XGKON) without open item management (SKB1-XOPVW)

For these accounting document items, the system derives the following flow types:

- Receivable or Payable
- 6 - Regular Receivables Increase
- 61 - Regular Payables Increase
- 62 - Unallocated Receivables Increase
- 621 - Unallocated Payables Increase
- Cash in Transit
- 86 - Incoming Cash in Transit Increase
- 88 - Outgoing Cash in Transit Increase
- Cash
- 96 - Incoming Bank Confirmed Cash Increase

- 98 - Outgoing Bank Confirmed Cash Increase

For items representing payables, receivables, or cash in transit, One Exposure from Operations derives cash forecasts.

Accounting documents containing other items than the ones described above are not handled by the integration of Financial Operations into One Exposure from Operations.

If you want (as an exception) additional accounting document items to be processed by the integration of Financial Operations into One Exposure from Operations, you have to define them as belonging to one of the above mentioned categories. You do this by assigning the corresponding G/L accounts to a flow type, which refers to the following flow categories:

- Flow Categories in Cash in Transit
- 86 - Incoming Cash in Transit Increase - 88 - Outgoing Cash in Transit Increase -

  Flow Categories in Cash

- 96 - Incoming Bank Confirmed Cash Increase
- 98 - Outgoing Bank Confirmed Cash Increase

If an item posts to such an account in debit, the system uses the first flow category; in credit, the system uses the second flow category. The corresponding flow types, which refer to these flow categories, are delivered. There is no need to create additional customer-specific flow types for this.

Memo Records

Define Number Ranges

In the "Number ranges" step, you specify one or more number ranges for manual planning. The system allocates a number from the appropriate interval to each memo record entered manually.

Actions

Enter under a two-digit key the number interval for memo records entered manually.

Define Planning Types

In this step you define the respective planning types for manual planning. Using the planning type, you control the manual entry of planned memo records. For each planning type, you specify

- the level to which the planning type is allocated
- the archiving category in which a memo record is stored after it becomes invalid
- whether memo records expire automatically
- the number range to which the planning type is allocated
- which fields are displayed for the respective planning type, and whether an entry is required or optional in the fields.

In addition, you specify a mnemonic name that is also displayed when memo records are created.

Actions

1. Create your planning types with a two-digit key.
2. Define each planning type according to your requirements.
3. Specify the field status (--> *Break down field status)* for each planning type.
   You must make these specifications for two field status groups:
   - general data
   - additional details

Additional information

For more information on planning types, see the online documentation via F1 and the Treasury manual.

Data Setup

Activate Individual Source Applications

Use

You use this activity to activate or deactivate source applications for One Exposure from Operations on a company code basis.

Once a source application is activated, the forecasted transactions from this source application are integrated into the One Exposure from Operations hub, and the forecasted cash flows are therefore available in Cash Management reporting.

You can initialize the One Exposure from Operations hub for existing source application data using the activity *Load Transaction Data Source Application into One Exposure from Operations Hub*.

The source application *One Exposure* is a prerequisite for all other source applications. Activating this source application means that the One Exposure from Operations hub accepts data at all. It is especially necessary to activate this source application if data from remote systems shall be transferred into the One Exposure from Operations hub.

The source application *Financial Operations* comprises accounting documents representing invoices and payments as well as bank statements. This source application is a prerequisite for all other source applications except for the source application *One Exposure*.

In this activity, you enter each combination of source application and company code individually. To activate or deactivate multiple company codes for several source applications simultaneously, use the activity *Activate Multiple Source Applications*.

Note:

The source application *One Exposure* enables in addition the derivation of Cash Management parameters like planning level, planning group, and liquidity item in accounting document line items, such as items belonging to customers/vendors, payment request clearings, bank clearings, and bank statements. For accounting documents that already exist, you can trigger the derivation of these Cash Management parameters using the activities *Rebuild Planning Levels, Groups, Dates in Accounting Documents*, *Rebuild Liquidity Items in Accounting Documents*, and *Rebuild Flow Types in Accounting Documents*.

Activities

To activate a source application for a company code:

1. Select a source application.

2. Enter a company code.
3. Select *Active*.
4. Save.

Activate Multiple Source Applications

Use

You use this activity to activate or deactivate multiple company codes for several source applications simultaneously.

To activate or deactivate source applications for company codes individually, use the activity Activate Individual Source Applications**.**

Activities

To activate source applications:

1. Select *Activate*.
2. Enter ranges for company codes.
3. Select one or more source applications.
4. Execute.

Rebuild Planning Levels, Groups, Dates in Accounting Documents

Use

You can use this Customizing activity to perform the following activities:

- Enter planning groups into customer and vendor master records
- Data setup for table BSEG
- Check Customizing

Activities

1. Customer/Vendor Data

You can use this function to enter planning groups in vendor and customer master records. Planning groups must be in place for open items to be updated in cash management. The lines in the table represent selection criteria for the various groups. The logic is as follows:

Ranking is determined by the priority. Priority 1 is highest priority. This criterion is checked first. If the customer or vendor meets these criteria, the group for this record is entered in the master record.

Example for customers:

Prio.	Ctry	PT low	PT high	Dunn.	Pyt meth	Group	Pyt.hist.			
1					GB	ZB1 ZB 1		U	E1	X
2					GB	E2				
3					E9					

If the terms from line 1 apply to a customer, the customer is assigned to planning group 1 and the flag *Record payment history* is set in the master record.

All the other customers in Great Britain are assigned group **E2**; foreign customers are assigned group **E9**.

Use the F1 and F4 help when filling out the selection criteria.

To ensure that all master records are assigned a planning group, you should always specify at least one structure (with lowest priority) to which a vendor or customer can be matched if no other will do.

Use the *Check* button to determine the number of master records to which no group has yet been assigned.

You can use the *Download* and *Upload* menu option (under *Utilities*) to store the selection criteria locally and import them again.

2. Data Setup

   You can use this function to fill data for planning level, planning group, and planning date in table BSEG.

3. Check Customizing

   This function allows you to check the configuration, for example, if a level is used in several applications or if a planning group is used in customer master records as well as vendor master records. (Both are not allowed and will cause errors.)

Insert House Bank and House Bank Account Data to Accounting Documents

Use

If you want to use historical cash management data in the Cash Management applications, you need to execute this program first to insert house bank (field *HBKID*) and house bank account (field *HKTID*) data into table *BSEG* before you use the applications.

Note:

The data update is not cross client. You must upload the data to your production client or excecute the program directly in your production client.

The program retrieves the house bank and house bank account data from historical transactions and copies the data into table *BSEG* for reporting purposes:

1. The program checks table *BSEG* and identifies entries that are with one or both of the following fields empty:
   - *HBKID* (house bank)
   - *HKTID* (house bank account)
2. The program retrieves data from the following sources and updates the entries in table *BSEG* accordingly.
   - Bank statements: line items with planning level of F
   - Payments for open items (transaction F11)
   - Payments for payment requests (transaction F111)

Activities

You only have to run this program once for inserting the house bank and bank account data from historical transactions. After the program finishes, the system displays the number of the rows that are updated.

Rebuild Liquidity Items in Accounting Documents

Use

In this Customizing activity, you can derive liquidity items for line items that are already posted to ledgers. Typically, liquidity items are automatically derived as postings are made. However, you must run this Customizing activity to generate liquidity items data for line items that are already posted but are missing liquidity items.

When building liquidity items data, the system reads required data from the BSEG table, which stores line items that are posted on accounting documents. Then it populates the derived liquidity items data back to the BSEG table so that other application features can consume the liquidity information directly.

This Customizing activity provides two modes that are optimized for different scenarios:

Initial-Load

- Optimized for first-time runs
- Intended to process large numbers of line items within an acceptable amount of time
- Does not support deriving liquidity items by using derivation functions
- Allows you to derive liquidity items by using a query sequence
- Allows you to specify a package size (the number of line items processed before the data is committed) for best performance

Rebuild

- Optimized for rebuilding liquidity items data after a first-time run
- Takes longer to process large numbers of line items

- Supports deriving liquidity items by using derivation functions
- Allows you to select a fraction of line items by certain criteria

Requirements

- You have activated the necessary source applications for FI in the relevant company codes by using either of the following Customizing activities:
- Activate Individual Source Applications
- Activate Multiple Source Applications

Activities

The general steps to build liquidity items data include:

1. Create queries in Customizing activity Define Queries for Liquidity Item Derivation (transaction FLQQA1).
2. Create query sequences for origin C, D, or X in Customizing activity Define Query Sequences (transaction code FLQC1).

   Note:

   For the *Initial-Load* mode, you can only specify query sequences with the orign ID of C or D.
3. Assign the queries to the defined query sequences accordingly in Customizing activity Assign Queries to Query Sequences (transaction code FLQQA).
4. Define the liquidity item derivation logic for company codes in Customizing activity Define Liquidity Item Derivation Settings for Company Codes.
5. Define the default liquidity items for G/L accounts as a fallback in Customizing activity Define Default Liquidity Item for G/L Accounts (transaction FLQINFACC).
6. Run this Customizing activity in *Initial-Load* mode.
7. Run this Customizing activity in *Rebuild* mode. Determine on which line items you want to generate liquidity items data.

Rebuild Flow Types in Accounting Documents

Use

In this Customizing activity, you derive flow types for accounting document line items that have already been posted.

Typically, flow types are automatically derived as postings are made. However, you must run this Customizing activity to derive flow types for accounting document line items that have already been posted but for which the flow types are missing.

Note: The way the system derives flow types for accounting document line items is described in the documentation of the Customizing activity *Assign Flow Types to G/L Accounts*.

Requirements

You have activated the source application *Financial Operations* in the Customizing activity Activate Individual Source Applications or Activate Multiple Source Applications.

Load Trans. Data from Source Appl. into One Exposure from Operations Hub

Use

You use this Customizing activity to load transaction data from source applications into One Exposure from Operations.

Transaction data from source applications is automatically integrated into One Exposure from Operations if the source application is activated. However, you must run this Customizing activity to integrate transaction data that had already been created before the relevant source application was activated.

The activation of a source application is done on a company code by company code basis.

Note: You can also use the Customizing activity *Build Cash Flows from Accounting Documents* to load transaction data from the source application *Financial Operations*.

Activities

You select one of the following modes:

- *Load Transaction Data*
  You use this mode to initially load transaction data into One Exposure from Operations.
- *Reload Transaction Data*
  You use this mode to reload transaction data into One Exposure from Operations after an initial load has already been carried out successfully. This is necessary if during the initial load some configuration settings (for example the derivation of liquidity items) were wrong.

Requirements

You have activated the corresponding source applications in the Customizing activity *Activate Individual Source Applications* or *Activate Multiple Source Applications*.

Build Cash Flows from Accounting Documents

Use

In this Customizing activity, you can derive cash flows and relevant information from accounting line items and store the results to the One Exposure from Operations hub for analytical purposes. It helps you do the following:

- Generate cash flows by loading accounting transactions from database table BKPF and BSEG to the One Exposure from Operations hub (database table FQM_FLOW)
- Derive important attributes that can be used to analyze the source and use of cash by building document chains in a bi-directional way

The attributes include but not limited to liquidity item, house bank account, WBS element, profit center.

You can define the maximum number of steps you want the program to execute when tracing document chains.

Requirements

You have configured Customizing activities under the following nodes:

- *Financial Supply Chain Management -> Cash and Liquidity Management -> Liquidity Items*
- *Financial Supply Chain Management -> Cash and Liquidity Management -> Flow Types*

Activities

You select one of the following modes:

Note:

This activity is client-specific and does not trigger transport. You must perform the activity for each client that you work with.

- Mass Run

  You use this mode to load all accounting line items from database table BKPF and BSEG massively into the One Exposure from Operations hub. You can use this mode for the following scenarios:

- Initial run

  Before you start to use cash management, it is recommended that you perform an initial run to load the historical data.

- Update run

  When the cash management is already in use and you have to change some of the relevant configurations, you can perform an update run after the configuration changes. This way, you can rebuild cash flows using the new logic.

- Delta Run Based on Queue Table

  You can use this mode to manually process accounting line items that are not yet loaded to the One Exposure from Operations hub.

  It is recommended that you schedule automatic background jobs for regular updates into the One Exposure from Operations hub.

Customize FI Flow Builder

Use

This Customizing activity consists of several tools and settings for the FI Flow Builder (transaction code FCLM_FLOW_BUIDLER).

Activities

You can perform the following activities:

> **Note**
>
> In the activities below, only the last one is a cross-client activity. Other activities are client-specific and do not trigger transports.

- Check the background job log

  You can check all the processed accounting transactions using the job log.

- Schedule background jobs

  You can schedule automatic background jobs for regular updates into the One Exposure from Operations hub.

- Delete obsolete flows derived from accounting line items

  You can use this feature to clean obsolete entries in the One Exposure from Operations hub (database table FQM_FLOW) that were derived from accounting line items.

  Example

  You have changed the flow type derivation logic to exclude transactions from company A. As a consequence, data that was already loaded into the One Exposure from Operations hub is no longer relevant for cash management and should be removed. However, with the configuration changes, the line items that are no longer identified as objects to be processed. As a result, you cannot clean the corresponding entries in the One Exposure from Operations hub by rebuilding cash flows using the mass run of the FI Flow Builder. In this case, you can use this feature to clean the obsolete entries from the One Exposure from Operations hub.

- Change default loading class

  You can use this tool to regenerate and assign your own loading class.

  Example

  When processing invoices, the FI Flow Builder uses the due date for net payment as the value date in forecasting cash flows. However, you want to use another date, for example, the due date for cash discount 1, as the value date. To do so, you must change the standard loading class as follows:

  a) Specify the *Generate Loading Class* radio button and specify a different date as the value date.

  b) Save the change to a loading class in customer namespace, for example, **ZCL_FCLM_BSEGFLOW_SAMPLE**.

  c) Specify the *Assign Loading Class* radio button and assign the loading class you created in the previous step.

Inbound Mapping for Integration of Remote Data into One Exposure

Reassign Company Codes

Requirements

You would like to implement distributed TR-CM.For the implementation, the settings in this table are mandatory.

If you do not make the settings, the TR-CM link is deactivated.

Standard settings

In this step, you do the following:

1. Set up the TR-CM subsystems

2. Convert the company code

Activities

Set up the TR-CM subsystems

You define an assigned subsystem by entering the logical system name in the **Subsystem** column.

Convert the company code

You can convert the company code of the sender TR-CM subsystem. To do this, enter the possible values of the sender TR-CM subsystem in the **Company Code (Subsystem)** column.

In the **Company Code (Central)** column, enter the company code valid for the central TR-CM system.

Further notes

You can also convert the company codes generically. If you specify a generic value, this is used for all the values in the 'Company Code' field that cannot be converted. Proceed as follows:

In the **Company Code (Subsystem)** column, enter an **asterisk (*)** instead of a concrete value.

In the **Company Code (Central)** column, enter the company code valid for the central TR-CM system, as above.

Distributed Cash Management

To distribute TR-CM data, you need to make cross-application and application-specific Customizing settings.

Cross-Application Customizing

In cross-application Customizing, you need to define logical objects and assign message types to the distribution model.

The following message types are relevant for the distribution model:

Message type	Use
CMSEND	TR-CM data from the subsystem to the central system
CMREQU	Request to send TR-CM data

Application Customizing

When you set up 'sending' TR-CM systems, you need to make cross-application Customizing settings only.

When you set up the central TR-CM system, you need to make some application-specific Customizing settings in addition to cross-application settings.

Convert Sender Planning Groups

Requirements

SAP recommends that you separate the sender TR-CM system groups from the central TR-CM system.

Before you convert the planning groups, you must have set up the sender TR-CM subsystems and company code conversions .

Standard settings

You make the settings for converting the planning groups from the sender TR-CM subsystems in the central TR-CM system. You also define which sender TR-CM system the conversion settings apply to.

Activities

1. Select a TR-CM send system from the list.
2. In the **Group (subsystem)** column, enter the possible values for the sender TR-CM subsystem.

   In the **Group (central)** column, enter the group valid for the central TR-CM system.

Further notes

You can also convert the planning groups generically. If you enter a generic value, this is used for all the values which cannot be converted.

To do this, proceed as follows:

In the **Group (subsystem)** column, enter an **asterisk (*)** instead of a concrete value.

In the **Group (central)** column, enter the group valid for the central TR-CM system, as above.

Convert Sender Planning Levels

Requirements

SAP recommends that you separate the planning levels of the sender TR-CM systems from the planning levels in the central TR-CM system.

Before you make the settings for converting the planning levels, you must first have set up the sender TR-CM subsystems and company code conversions.

The entries in this table are mandatory.

Standard settings

You make the settings for converting the planning levels of the sender TR-CM subsystems in the central TR-CM system. In addition, you define which sender TR-CM subsystems the conversions apply to.

Activities

1. Select a sender TR-CM subsystem from the list.

2. In the **Level (subsystem)** column, enter the possible values for the sender TR-CM subsystem.

   In the **Level (central)** column, enter the planning level valid for the central TR-CM system.

Further notes

You can also convert the planning levels generically. If you make a generic entry, this is used for the all the values that cannot be converted.

To do this, proceed as follows:

- In the **Level (subsystem)** column enter **+1** for **cash position** levels, instead of a concrete value

or

- In the **Level (subsystem)** column, enter **+2** for **liquidity forecast** levels, instead of a concrete value.

- In the **Level (central)** column, enter the planning level valid for the central TR-CM system, as above.

Convert Sender Business Areas

Requirements

SAP recommends that you separate the business areas of the sender TR-CM systems from those of the central TR-CM system.

Before you make the settings for converting the business areas, you must first have set up the sender TR-CM subsystems and the company code conversions.

Standard settings

You make the settings for converting the business areas of the sender TR-CM subsystems in the central TR-CM system. In addition, you define which sender TR-CM subsystems the conversions apply to.

Activities

1. Select a sender TR-CM subsystem from the list.
2. In the **Business area (subsystem)** column, enter the possible values for the sender TR-CM subsystem.

   In the **business area (central)** column, enter the business areas valid in the central TR-CM system.

Further notes

You can also convert the business areas generically. If you enter a generic value, this is used for all the values that cannot be converted.

To do this, proceed as follows:

In the **Business area (subsystem)** column, enter an **asterisk(*)** instead of a concrete value.

In the **Business area (central)**, enter the business areas valid in the central TR-CM system, as above.

Assign Time Zones to Subsystems

Use

If the TR-CM subsystems that you have defined belong to a different time zone than the central TR-CM system, you need to assign time zones to the subsystems in this activity.

Requirements

The time zones need to have been created in the activity Maintain Time Zones.

Standard settings

If all subsystems belong to the time zone assigned to the central system, no entries are required here.

Activities

1. Select a TR-CM send subsystem from the list.
2. Select the time zone of the subsystem.
3. Save your entries.

Check Settings Recommendation

Once you have made all the settings for the TR-CM link, you should run a check on the settings, regardless of whether you have set up a central TR-CM system or a sending TR-CM subsystem.

Activities

Start transaction **FF$6** by executing the Customizing activity or by entering the transaction code directly.

Further notes

The system generates a log which tells you to what extent the settings you have made for the TR-CM link are consistent.

If you have not made any changes, the system merely displays an error message.

Tools

Import and Export Liquidity Items

Import Liquidity Items

Use

In this Customizing activity, you can import liquidity items by uploading a local file.

Activities

Prepare a local file with required liquidity item information. You can download a template using Customizing activity Export Liquidity Items (transaction FLQLPOS). To do so, in this Customizing activity, execute the program, then choose *Local File -> Unconverted*.

After the import, you can check the imported liquidity items in Customizing activity Edit Liquidity Items (transaction FLQC1).

Export Liquidity Items

Use

In this Customizing activity, you can generate a list for the liquidity items in the system. You can export the list to a local file if needed.

Business Add-Ins (BAdIs)

Cash Flow Items

BAdI: Bank Details

Use

With this BAdI, you can display additional data (fields) for Bank Statement overview information within the *Check Cash Flow Items* application detail view.

In order to receive data enhancements, the **ENHANCE_DETAILS** method is called. With this method, you can transfer added fields to the corresponding fields of the internal gateway structure.

Requirements

- You have enhanced the **FCLM_CP_S_ENTITY_BS** data structure.

- You have redefined the **FCLM_CP_PAYMENTDETAIL** gateway project and enhanced the **BankStatement** entity type.

Standard settings

The BAdI has the following attributes:

- Not activated in the standard system
- Not filter-dependent
- Can be used in multiple ways

Activities

For information about implementing BAdIs as part of the Enhancement Concept, see SAP Library for SAP NetWeaver under BAdIs - Embedding in the Enhancement Framework.

See also

BAdI method documentation:

Enhance Bank Statement Details

BAdI: Bank Communication Management

Use

With this BAdI, you can display additional data (fields) for Bank Communication Management overview information within the *Check Cash Flow Items* application detail view.

In order to receive data enhancements, the **ENHANCE_DETAILS** method is called. With this method, you can transfer added fields to the corresponding fields of the internal gateway structure.

Requirements

- You have enhanced the **FCLM_CP_S_ENTITY_BCM** data structure.
- You have redefined the **FCLM_CP_PAYMENTDETAIL** gateway project and enhanced the **PaymentBatch** entity type.

Standard settings

The BAdI has the following attributes:

- Not activated in the standard system
- Not filter-dependent
- Can be used in multiple ways

Activities

For information about implementing BAdIs as part of the Enhancement Concept, see SAP Library for SAP NetWeaver under BAdIs - Embedding in the Enhancement Framework.

See also
BAdI method documentationEnhance Bank Communication Management Details

BAdI: Memo Records

Use

With this BAdI, you can display additional data (fields) for Memo Record overview information within the *Check Cash Flow Items* application detail view.

In order to receive data enhancements, the **ENHANCE_DETAILS** method is called. With this method, you can transfer added fields to the corresponding fields of the internal gateway structure.

Requirements

- You have enhanced the **FCLM_CP_S_ENTITY_MEMO** data structure.
- You have redefined the **FCLM_CP_PAYMENTDETAIL** gateway project and enhanced the **MemoRecord** entity type.

Standard settings

The BAdI has the following attributes:

- Not activated in the standard system
- Not filter-dependent
- Can be used in multiple ways

Activities

For information about implementing BAdIs as part of the Enhancement Concept, see SAP Library for SAP NetWeaver under BAdIs - Embedding in the Enhancement Framework.

See also

BAdI method documentation: Enhance Memo Record Details

BAdI: Accounting Documents

Use

With this BAdI, you can display additional data (fields) for Accounting Document overview information within the *Check Cash Flow Items* application detail view.

In order to receive data enhancements, the **ENHANCE_DETAILS** method is called. With this method, you can transfer added fields to the corresponding fields of the internal gateway structure.

Requirements

- You have enhanced the **FCLM_CP_S_ENTITY_ACC_DOC** data structure.
- You have redefined the **FCLM_CP_PAYMENTDETAIL** gateway project and enhanced the **AccountingDocument** entity type.

Standard settings

The BAdI has the following attributes:

- Not activated in the standard system
- Not filter-dependent
- Can be used in multiple ways

Activities

For information about implementing BAdIs as part of the Enhancement Concept, see SAP Library for SAP NetWeaver under BAdIs - Embedding in the Enhancement Framework.

See also

BAdI method documentation: Enhance Accounting Document Details

BAdI: Payment Items

Use

With this BAdI, you can display additional column (fields) for Payment Document Item information within the *Check Cash Flow Items* application ALV table list.

In order to receive data enhancements, the **ENHANCE_DETAILS** method is called. With this method, you can transfer added fields to the corresponding fields of the internal gateway structure.

Requirements

- You have enhanced the **FCLM_CP_S_ENTITY_PAYMENTITEM** data structure.
- You have redefined the **FCLM_CP_PAYMENTDETAIL** gateway project and enhanced the **PaymentDetailItem** entity type.

Standard settings

The BAdI has the following attributes:

- Not activated in the standard system
- Not filter-dependent

- Can be used in multiple ways

Activities

For information about implementing BAdIs as part of the Enhancement Concept, see SAP Library for SAP NetWeaver under BAdIs - Embedding in the Enhancement Framework.

See also

BAdI method documentation

Enhance Payment Item

BAdI: Treasury Management Details

Use

With this BAdI, you can display additional data (fields) for Treasury and Risk Management overview information within the *Check Cash Flow Items* application detail view.
In order to receive data enhancements, the **ENHANCE_DETAILS** method is called. With this method, you can transfer added fields to the corresponding fields of the internal gateway structure.

Requirements

- You have enhanced the **FCLM_CP_S_ENTITY_TRM** data structure.
- You have redefined the **FCLM_CP_PAYMENTDETAIL** gateway project and enhanced the **TRMDocument** entity type.

Standard settings

The BAdI has the following attributes:

- Not activated in the standard system
- Not filter-dependent
- Can be used in multiple ways

Activities

For information about implementing BAdIs as part of the Enhancement Concept, see SAP Library for SAP NetWeaver under BAdIs - Embedding in the Enhancement Framework.

See also

BAdI method documentation

Enhance TRM Details

Liquidity Planning

Specify a BPC Configuration Set

Use

In this Customizing activity, you specify a configuration set that you want to use for liquidity planning.

Requirements

If you want to use your own configuration set, make sure you have defined the following objects in SAP BusinessObjects Planning and Consolidation, version for SAP NetWeaver:

- *Environment*
- *Model*
- *Process Template*
- *InfoProviders*

Standard settings

By default, the standard configuration set provided by SAP is activated.

Activities

You can create your own configuration set by creating a new entry. After you have entered all the necessary data, activate the entry that you want to use.

Note

You can only activate one configuration set.

Maintain Planning Unit Settings

Use

In this Customizing activity, you can make and change settings for planning units used in liquidity planning. Planning units are organizational units that need to enter liquidity plan data. Each planning unit corresponds to one company code.

Activities

You can do the following:

- Create planning units and define the following attributes:
- *Description*: Specify a description for the planning unit.
- *Aggregation Currency*: Specify a currency that you want to use to consolidate the amounts in different planning currencies in the aggregation currency view. You define one aggregation currency for each planning unit.
- *Performer*: Specify the user responsible for preparing liquidity plans in this planning unit. The user submits the prepared plans to the reviewer for approval.
- *Reviewer*: Specify the user responsible for evaluating liquidity plans submitted in this planning unit. The user decides whether to approve or reject a plan.
- *Company Code*: Specify the company code for this particular planning unit.
- Define planning currencies for planning units

Specify the currencies to be used to enter plan amounts into the liquidity plans for the selected planning unit.

Activate Planning Unit Hierarchy

Use

You use this Customizing activity to activate a planning unit hierarchy. This is then the only planning unit hierarchy and is used for all liquidity planning activities.

Define Currency Conversion Rules

Use

To mitigate the risks of possible exchange rate fluctuations, you may want to hedge against exchange risks by exchanging your cash surplus into another currency. To do so, you must first define the rules for currency conversion using this Customizing activity.

Requirements

- You have defined the following liquidity items in Customizing activity *Edit Liquidity Items*.
- `LP_EXI:` Cash converted to another currency
- `LP_EXF:` Cash converted from another currency

- You have defined **both** the source currency and the target currency as planning currencies for **both** the source planning unit and the target planning unit specified in a currency conversion rule.

Activities

You define a currency conversion rule for each currency that you want to convert.

To define a rule, create a new entry and specify the following:

- *Outflow Liquidity Item* (cash outflow liquidity item): `LP_EXI`
- *InflowLiquidity Item* (cash inflow liquidity item): `LP_EXF`
- *Source Planning Unit:* The planning unit that is purchasing the currency
- *Target Planning Unit:* The planning unit that is selling the currency
- *Src. Curr.:* The currency to be sold
- *Tar. Curr.:* The currency to be purchased

Define Reference Data Sources

Use

You may want to define reference data sources other than the standard ones provided by SAP. To do so, you can define the data sources in this Customizing activity. This way, you can reuse the standard BW modeling provided by SAP.

Standard settings

SAP provides the following three data sources:

- Actual amount
- Forecasted amount from liquidity forecast
- Plan amount from previous planning cycle

Exclude Liquidity Items Representing Balance Values

Use

Liquidity items can be defined to represent cash flows or balance values. In liquidity planning, only the liquidity items that representing cash flows are taken into calculation. To exclude liquidity items that represent balance values from liquidity planning calculation, you must use this Customizing activity to register all the liquidity items defined in the system that representing balance values. Otherwise, the plan data may be miscalculated.

Define Liquidity Planning Types

Use

You may want to define planning types other than the standard ones provided by SAP. To do so, you can define the planning types that you want to use in this Customizing activity. This way, you can reuse the standard BW modeling provided by SAP.

Standard settings

SAP provides the following two planning type:

- Monthly rolling plan: Liquidity plans of this type are continuously updated in consecutive months based on the latest information. The plan resulting from each planning cycle is saved as a seperate version.
- Non-rolling plan

Country-Specific Settings

China

Cash Budgeting

Activate HR Organizational Hierarchy

Use

In this Customizing activity, you can activate the use of the HR organizational hierarchy.

This setting determines which organizational hierarchy is used in the workflow of an approval process. An organizational hierarchy can be generated in one of two ways:

- HR-Organization - the hierarchy is defined by HR
- User-defined - the user creates a customized hierarchy based on a template **Note:**

The activation action is **not** revocable. Once you activate HR Organization, you cannot deactivate it.

Activities

To activate the use of *HR Organization*, you select the Activate check box. Once you activate the selection and save it, the selection will be greyed out and you cannot revoke this action.

Selecting this option enables the system to use the HR organizational hierarchy as the basis in the workflow of an approval process.

To use a user-defined hierarchy, you must upload a customized hierarchy from a template.

Define Cash Budget Areas

Use

The structure of the cash budget plan is based on a cash budget area, that is, you budget a certain amount of cash to the cash budget units in a cash budget area.

In this Customizing activity, you determine the cash budget areas needed for your cash budget plan and their structure. To do this, you set up an organizational hierarchy for each cash budget area. Under the organizational hierarchy, you organize your cash budget units and their detailed settings such as company codes, controlling areas, cost centers, profit centers, and currencies.

Based on the settings that you make here, when users enter a cash budget area on the *Set Cash Budget Area* screen, the system displays the corresponding organizational hierarchy.

Activities

The cash budget area must be assigned a unique organizational hierarchy. In addition, the cash budget area has the following attributes:

- Fiscal year variant
- Factory calendar
- Currency
- Exchange rate type

Define Cash Budget Cycle Types

Use

In this Customizing activity, you define the default values of a cycle type, which will be proposed when the user creates cash budget cycles of this cycle type.

You can define the following attributes of a cash budget cycle:

- The frequency of creation of such cash budget cycles
- The number range that these cash budget cycles follow
- The possibility of setting the deadline of a cycle; and the default value of the deadline

When a user creates a cash budget cycle, on the *Create Cash Budget Cycle* screen, these cycle types defined in the back-end system are proposed as the possible field values in the *Cycle Type* field. When you select a cycle type, the settings defined in this activity applies.

Activities

To define a normal rolling cycle, select the *Rolling Cash Budget* check box and enter a number in the *Rolling No.* field.

If you do not want to delimit the weekly budget cycle at the end of a month by creating a new weekly budget cycle that starts on the first day of the next month, select the *Allow Cross-Period Cycles* check box.

Define Mixed Rolling Cycle Types

Use

A mixed rolling cycle allows users to start a series of multiple-period cycles for their cash budget that can include a daily cycle, a weekly cycle, or a monthly cycle. In this Customizing activity, you define mixed rolling budget cycle types. The mixed rolling cycles that you define must contain at least two levels of periods.

Requirements

You have defined regular cycle types in the Define Cash Budget Cycle Types activity. Only these cycle types can be used in your mixed rolling cycles. The regular period must be 1.

Standard settings

The default time units are monthly, weekly, and daily.

Activities

1. Enter a mixed rolling cycle type and its description.
2. Double-click the cycle type to display the *Assigned Mixed Rolling Cycle Types* view.
3. In the *Cycle Type* fields for day/week/month, enter the daily, weekly, and monthly cycle types.
4. For each cycle type, assign a unit of time and a regular cycle type.
5. For a monthly cycle, enter the rolling number. In the *Rolling Number* field, enter an integer between 1 and 12.
6. Save your entries.

Define Number Ranges for Cash Budget Cycles

Use

To create a cash budget cycle, the system requries a unique cycle number. You maintain the number ranges for cash budget cycles in this Customizing activity.

Define Checking Rules

Use

To prevent data inconsistencies between the different types of cycles, you can define a rule to check the budgeted amounts in this Customizing activity. For example, the check can determine if the budgeted amounts for a monthly cycle conflict with the amounts for weekly cycles defined in the same cash budget area.

Requirements

You have created the liquidity item hierarchy in the Define Liquidity Item Hierarchies activity.

Activities

You enter a cash budget area, a cash budget cycle type, and a liquidity item hierarchy. Then in the Assign Checking Rules to Cash Budget Areas and Cycle Type view, you enter the follow-on cycle types. For example, for a weekly cycle type, the follow-on types could be monthly or yearly.

If you want to void a checking rule without deleting it, deselect the Activated check box.

Example

Your users will create a weekly cash budget cycle. Within the same cash budget area, your users will also create a monthly cash budget cycle.

To allow them to use the checking function, you perform the following activities:

1. You enter weekly as the cash budget type value and the related liquidity item hierarchy.
2. You enter monthly as the cash budget type value in the Assign Checking Rules to Cash Budget Areas and Cycle Type view.

When users save the cash budgeting data, the system displays a warning message or an error message if the total budgeted amount for the four weeks in the month exceeds the monthly cycle budget data.

Define Approval Process Rules

Use

In this Customizing activity, you define an approval process rule that corresponds to a detailed workflow for cash budgeting approval in a cash budget area.

Example

You want to allow users to define an approval process for a cash budget item that requires approval first by a team lead, then by a manager, and finally by the general manager. To do so, you enter the approval process rule CASH and a description in this Customizing activity. As a result, users can enter the details of the CASH process rule with the Edit Process Rule transaction.

Assign Approval Rules to Cash Budget Areas and Cycle Types

Use

In this Customizing activity, you assign an approval process rule to the combination of a cash budget area and cycle types. You can assign multiple cycle types to a cash budget area and a process rule.

Once you have activated the assignment of an approval process rule in a cash budget area, the system determines the approval workflow for the cycle type. The workflow is triggered when the approver starts the approval process.

Requirements

You have created and defined cycle types in the Define Cash Budget Cycle Type Customizing activity .

Activities

With this Customizing activity, you are able to:

1. Assign a process rule for a cycle type in a cash budget area.
2. Select the Activated checkbox to enable the display of the process rule in the Assignment list on the Approval Process Rule screen.
   If you do not want users to see an assignment in the Assignment list, deselect the Activated checkbox.

Maintain Settings for Automatic Notifications and Alerts

Use

In this Customizing activity, you maintain the settings for the automatic notifications and alerts for a cash budget cycle. The automatic notifications and alerts are displayed in the Worklist of a user's homepage.

- *Notifications* - A notification is sent to the user when the status of a cash budget cycle has changed.
- *Alerts* - An alert is sent to the user when a target value has been reached. The target value may be a date or an amount.

Requirements

You have defined the following Customizing activities:

- Define Cash Budget Cycle Type
- Define Cash Budget Areas

You have created and started a cash budget cycle.

Activities

1. Specify the *Cash Budget Area.*
2. Specify the *Cycle Type*.
3. Select options for *Notification*.

   Cash Budget is Approved

   Indicates if a notification is received when a cycle is approved.

   If indicator is selected, the previous processor who submitted the cash budget cycle receives a notification on their homepage when the cash budget cycle is approved.

   Cash Budget is Completed

   Indicates if a notification is received when a cash budget cycle status is set to Completed.

   If indicator is selected, all processors in the cycle receive a notification on their homepage when a cash budget cycle status is set to Completed.

4. Select options for Alerts.

   Deadline Alert

   Determines whether the system triggers an alert when a deadline date is approaching. If indicator is selected, all processors in the process flow, except for those whose task has already been closed, receive an alert on their homepage when the deadline alert day has been reached.

   To specify the deadline alert, enter the alert days in the Days before Deadline field.

   This number indicates the number of days before the deadline to trigger the alert. **Note:** If you select to enable Deadline Alert, you must also enter the alert days in the Days before Deadline field to indicate when to trigger the alert.

   Consumption Alert

   Determines whether the system triggers an alert when the specified consumption amount is reached.

   If indicator is selected, all processors in the cycle who have Approval authority receive an alert when the consumption amount has been reached. To specify the consumption alert, do one of the following:

   - Enter an amount for the *Alert Tolerance* - This amount indicates the value that must be reached before an alert is triggered.
   - Enter an *Alert Percentage* - This value indicates a percentage of the cash budget cycle that must be reached before an alert is triggered.

Example

You have a cash budger amount of $12, and want users to receive an alert when 9% of the cash budget amount is used. The calculation is as follows: $12, * 9/1 = $1,8 To enable the alert trigger, you perform the following actions:

5. On the *Cash Budget Notification and Alert* overview page, select an entry and then double click the *Notification* or *Alert* column to configure the alert settings.

6. In the *Alert* group box, select *Consumption Alert*.
7. In the *%* field, enter 9. This number signifies the percentage of the cash budget amount.
8. Save your changes and return to the overview page. Ensure that the *Alert* check box is selected for the corresponding entry you have just configured.

When $1,8 of the cash budget has been used, users will receive an alert on their homepage.

Define Number Ranges for Cash Budget Documents

Use

In this Customizing activity, you define a number range for cash budget documents.

1 Define Payment Control Rules

Use

In this Customizing activity, you define a rule to specify when to warn users about budget deficits or prevent more payments from being made. To determine a budget status of an actual payment, the payment is added to the total of consumed budgets on a liquidity item and is compared with the budget amount or percentage. If the amount or the percentage exceeds the values you defined in this activity, a warning message or an error message is displayed. Users can ignore the warning and continue with the payment. If an error message is received, no further payments can be made.

Requirements

You have completed cash budget cycles.

Activities

You can define the currency types, cumulation options, and the characteristics used for an actual payment calculation.

You can specify whether to use the original values or to use the absolute values.
You can specify how to select characteristics for the calculation of budgeted and actual payments from the CMCB_PMT_CTRL_RULE_AC Web Dynpro configuration.

- Single Control Rules
  a) Select *Single* in the *Control Type* field.
  b) Select *Liquidity Item* or *Organizational Unit* in the *Characteristics* group box, for example, to sum the amounts of a selected set of liquidity items.
     The values of the characteristic F4 help is from BW InfoCube. The default values are liquidity items and organizational units.

- Relative Control Rules

  c) Select *Relative* in the *Control Type* field.

  d) In the *Base Measurement* field and the *Index Measurement* field, specify the sign of the payment amounts that you use in the base factor and the index factor in the fraction.

  e) Select *Liquidity Item* or *Organizational Unit* in the Characteristics group box, for example, to sum the amounts of a selection of liquidity items.

2 Change Message Controls

Use

In this Customizing activity, you can change the message types of the messages in the CMCB_MSG message class.

Activities

Enter **CMCB_MSG** in the *Application Area* field.

3 Define Payment Control Strategies

Use

In this Customizing activity, you can create, delete and edit payment strategies. A payment strategy contains any number of payment rules.

4 Assign Payment Control Strategies to Cash Budget Areas and Cycle Types

Use

In this Customizing activity, you assign a cash budget area and a cycle type to your payment control strategy.

Activities

1. Enter a cash budget area, a cycle type, and a payment control strategy. If the cycle type is a regular style and the payment control rule is assigned with a cumulative period, you must ensure the compliance between the cumulative period and the regular period.
2. Save your entries.

Business Add-Ins (BAdIs)

BAdI: Cash Budgeting Liquidity Item Derivation

Use

This Business Add-In (BAdI) uses the Liquidity Calculation: Forecast from Accounting (Open Items) (RFLQ_CASH_FORECAST_CCR) report to determine the right liquidity items for accounting documents. You can use this BAdI to define your derivation function modules or selection criteria to determine liquidity items.

Standard settings

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

See also

BAdI method documentation: Liquidity Item Derivation for Accounting Document

BAdI: Cash Budgeting Org Unit Derivation from BoE and EPIC

Use

You use this Business Add-In (BAdI) to specify the corresponding organizational unit for the budget consumption from the *Bill of Exchange Management* or *Electronic Payment Integration* solution. The BAdI is used when the organizational unit information from the two solutions is empty.

Standard settings

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

See also

BAdI method documentation:

- Org Unit Derivation from BoE
- Org Unit Derivation from EPIC

BAdI: Payment Document Information from BoE and EPIC

Use

You use this Business Add-In (BAdI) to specify additional information, such as payment document, to display in the *Planning and Consumption Report*. For example, you can specify a specific table of *Bill of Exchange Management* or *Electronic Payment Integration* solution that stores the globally unique identifiers (GUIDs) generated for consumption items. The information from the two solutions is displayed in the *Planning and Consumption Report*.

This BAdI is designed for SAP internal use only.

Standard settings

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

See also

BAdI method documentation: Get Payment Document Information from BoE and EPIC

BAdI: Get Planing Data from Real-time InfoCube

Use

You use this Business Add-In (BAdI) to add your own characteristics, key figures, and selection criteria when the system retrieves budget planning data from the *Real Time InfoCube for Cash Budgeting* (/EFI/CB_IC1) real-time InfoCube.

In addition, if you have created a new real-time InfoCube and use the new InfoCube to store the budget planning data, you can use this BAdI to replace the standard InfoCube with the InfoCube that you have created.

You use this BAdI when you have added customized InfoCube, characteristics, key figures, or selection criteria for cash budgeting purposes.

Standard settings

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

See also

BAdI method documentation: Prepare data for FM RSDRI_INFOPROV_READ

BAdI: Copy Suggested Amount into Planning InfoCube

Use

You use this Business Add-In (BAdI) to add your own filters, variable values, and planning function that the system uses to retrieve the relevant suggested cash budget amounts and saves the amount to the relevant planning InfoCube.

You use this BAdI when you have created your own filters, variables, and planning function. You can replace the standard filters, variables, and planning function with the ones you have created.

Standard settings

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

See also

BAdI method documentation: Copy Suggested Amount

BAdI: Delete Data in InfoCube

Use

You use this Business Add-In (BAdI) to add your own filters, variables, and planning functions that the system uses to delete the cash budget data in the relevant planning InfoCube.

You use this BAdI when you have created your own filters, variables, and planning functions. You can replace the standard filters, variables, and planning functions with the ones that you have created.

Standard settings

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

See also

BAdI method documentation: Delete Data in InfoCube

BAdI: Budget Planning and Budget Consumption Details

Use

You use this Business Add-In (BAdI) to enhance the query logic to retrieve the budget planning data and budget consumption data based on the customer's requirements.

Standard settings

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

See also

BAdI method documentation:

- Retrieve Budget Planning Data
- Retrieve Budget Consumption Data

BAdI: Display of Subtotals and Items of Suggested Amounts

Use

You use this Business Add-In (BAdI) to display the subtotals and the items of a suggested amount on the planning UI.

Standard settings

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

See also

BAdI method documentation:

- Derive Details
- Get Query and Selection

BAdI: Auto-Fill Fields on Cash Budget Planning UI

Use

You use this Business Add-In (BAdI) to enhance the auto-fill function on the cash budget planning UI.

Standard settings

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

See also

BAdI method documentation:

- Auto-Fill

BADI: Send Email

Use

You use this Business Add-In (BAdI) to define email addresses, subject, and content. The system can automatically send an email to the relevant email address when a task, notification, or alert is created.

Standard settings

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

See also

BAdI method documentation: Send Mail

1 BAdI: Check FI Postings for Cash Budget Consumption

Use

This Business Add-In is used in the Conumption Syncronization for Actual Payments Web Dynpro application.

For example, you can implement your own logic to check whether the postings are from FLQITEM table before the user saves the consumption of cash postings from accounts receivables and accounts payables in FI except for Bill of Exchange and Electronic Payment Interface for China.

Activities

For information about implementing BAdIs as part of the Enhancement Concept, see SAP Library for SAP NetWeaver under BAdIs - Embedding in the Enhancement Framework.

See also

BAdI method documentation Enable Check on Consumption

2 BAdI: Characteristics Relationship

Use

You can use this Business Add-In (BAdI) to change the logic of creating and checking characteristic relationships. The **RSPLAN** transaction code is used to define characteristic relationship in the Business Warehouse client. In the cash budget area, the characteristic relationship is **EXIT** type, which has the class **CL_CMCB_PLANNING_CR_EXIT**.

Activities

For information about implementing BAdIs as part of the Enhancement Concept, see SAP Library for SAP NetWeaver under BAdIs - Embedding in the Enhancement Framework.

See also

BAdI method documentation

- Create Relationship
- Check Relationship

3 BAdI: Planning User Interfaces

Use

You can use this Business Add-In (BAdI) to enhance the button control on the planning page.

Activities

For information about implementing BAdIs as part of the Enhancement Concept, see SAP Library for SAP NetWeaver under BAdIs - Embedding in the Enhancement Framework.

See also

BAdI method documentation

Override Button Control

4 BAdI: Virtual Providers of Suggested Amounts

Use

You can use this Business Add-In (BAdI) to change the logic of getting suggested amounts from the **/EFI/CB_VP1** InfoProvider.

Activities

For information about implementing BAdIs as part of the Enhancement Concept, see SAP Library for SAP NetWeaver under BAdIs - Embedding in the Enhancement Framework.

See also

BAdI method documentation:

- Read Data
- Define Data