

PUBLIC

Document Version: 2605 – 2026-06

Procurement Data Import and Administration Guide

SAP Ariba Buying and Invoicing
SAP Ariba Buying

© 2026 SAP SE or an SAP affiliate company. All rights reserved.

Content

Procurement Data Import and Administration Guide. 11

Introduction to Defining and Importing Procurement Data. 12

Working with Procurement Data. 13

Configuration and Management of Budgets in Your Buying Solution. 14

Budget Checks in Your Buying Solution. 15

Budget Import. 15

Uploading Budget Data Using Microsoft Excel. 17

Budget Data Formatting in CSV Files. 17

Sample CSV File Formats for Generic ERP. 18

Sample CSV File Formats for PeopleSoft ERP. 19

Sample CSV File Formats for SAP ERP. 21

Budget Check Configuration. 22

Budget Check Configuration During Final Approval of Requisitions. 23

Budget Synchronization with an External ERP System. 24

Enabling Budget Check at a Customer Site That Is Already in Production. 25

Disabling Budget Check for an Accounting Combination. 25

Reservation of Funds for Requisitions. 26

Release of Funds from Reserved Budgets for Closed Purchase Orders. 27

Use Case Scenarios. 28

Budget Check on Imported Requisitions. 29

Budgets Validation. 29

Ranking Budgets. 30

Performing a Budget Search. 31

Budget Adjustments. 33

Importing Budget Adjustments. 34

Sample Budget Adjustments File. 34

Budget Updates. 35

General Guidelines for Budget Updates. 35

Import Tasks for Closing and Reopening Budgets. 36

Considerations for Reopening or Closing Budgets. 36

Importing Budget Code and Period Status. 37

Incremental Load of Budgets. 38

Sample Files for Incremental Load of Budgets for a Simple Generic ERP. 39

Budget Export. 41

Budget Transactions Export. 41

Sample BudgetTransactionExport.csv File. 41

Budget Checks in External Systems. 43
Enabling Budget Checks in External Systems. 43
Approvables Migrated from Locally-Integrated Budgets to Budgets in an External System. 44
Budget Checks on Aggregated Requisitions. 45
Configuring Web Services Integration Tasks for Budget Checks in External Systems. 46
Additional references. 46

Funds Management Accounting. 47
Configuring Funds Management Accounting. 48
Enabling Web Services Integration Tasks for Funds Management Account Assignment Fields. 48
Importing Flexible Master Data Templates for Funds Management Account Assignment Fields 49

Branding and Conditions. 54
Import Purchase Order Conditions Task. 54
Configuration of Purchasing Unit-Specific Terms and Conditions for Purchase Order Print. 55
Customizing Purchasing Unit-Based Branding of Purchase Order Prints. 56
Customization of Purchase Order Print Header. 57

Import of Requisitions. 58
Item Master File. 60
Item Master Mappings. 60
Item Master. 63
Creating the Item Master from the Administration User Interface. 63
Editing the Item Master Data. 64
Loading an Item Master Using the CSV File Import Task. 65
Sample Item Master CSV File Formats. 66
Use customer catalog prices in SAP Ariba solutions to replace user-defined prices of customer catalog line items in imported requisitions. 67
Import of Requisitions with Service Items. 67
Import of Requisitions Using Various Integration Channels. 68
Import of Requisitions Using CSV Files. 69
Import of Requisitions Using Web Services. 76
Import of Requisitions Asynchronously for Sites Integrated with SAP ERP Using SAP Integration Suite, managed gateway for spend management and SAP Business Network 86
Import of Requisitions Using cXML Posts. 89
Import of Requisitions with Kits. 92
Importing Requisitions with Kits. 93
Exporting Kits to an External ERP System. 94
Example data: Kit_Export.csv and KitItem_Export.csv. 95
Import of Requisitions with User Set Fields. 95

Import of Requisitions with Account Type Data. . . . 97
Import of Contracts with Requisitions. . . . 97
Import of Requisitions with Subcontractor Information for Suppliers. . . . 97
ERS Flag Value in Imported Requisitions. . . . 98
Considerations for Importing Requisitions. . . . 99
Defaulting of Line Item Price When Buyer Part Number and Facility Do Not Match. . . . 101
Restrictions on Requisition Import. . . . 101
Import of Amendments. . . . 103
Prerequisites for Importing Amendments. . . . 104
Import of Amendments Using Web Services. . . . 104
Error Messages for Failed Amendment Import. . . . 108
APIs to Access Submit Failure Records for Imported Requisitions. . . . 110

Rules for Demand Aggregation. . . . 113
Demand Holding Rules. . . . 113
Assignment Rules for Demand Aggregation. . . . 114
Demand Aggregation Rules. . . . 115
Import Demand Holding Rules Task. . . . 116
Import Assignment Rules for Demand Aggregation Task. . . . 118
Import Demand Aggregation Rules Task. . . . 121
Running the Auto-Aggregate Held Line Items Scheduled Task. . . . 123
Modifying Demand Holding, Assignment, and Aggregation Rules. . . . 124
Deactivating Demand Holding, Assignment, and Aggregation Rules. . . . 125
Demand Aggregation for MRP-run Requisitions in Sites Enabled for Demand Aggregation and External Budget Checking. . . . 125
Prerequisites. . . . 126
Restrictions. . . . 126
Web Services Tasks. . . . 127
Workflow for Managing Demand Aggregation and External Budget Checks for MRP-run Requisitions. . . . 128

Collaborative Requisitions. . . . 130
Dynamic Collaboration. . . . 131
N Bids and a Buy. . . . 131
Data Import Tasks for Collaboration Information. . . . 132
Import Collaboration Reasons Task. . . . 132
Import Bidding Types. . . . 133
Quick Enablement Collaboration Request. . . . 133
Enabling Quick Enablement Collaboration Request. . . . 134
Quick Enablement Collaboration Request Workflow. . . . 135

Predefined Notes in a Requisition. . . . 136
Adding Rules in the **Document Manager** . . . 137

Adding Notes in the **Document Manager** . . . 138
Examples: Configuring rules and defining values for the rules to retrieve predefinied notes. . . . 139

Understanding Order Methods. . . . 142
Order Methods. . . . 144
Available Order Methods. . . . 144
Carbon Copy Direct Order (CC:DO) Orders. . . . 146
General Order Methods. . . . 146
SAP Ariba Order Method. . . . 147
SAP Ariba PCard Master Order Method. . . . 147
SAP Ariba PCard Order Method. . . . 147
SAP Ariba GhostPCard Order Method. . . . 148
SSP All Direct Order Method. . . . 149
ERP Order Methods. . . . 149
Order Numbers. . . . 150
Order Numbers That Appear to Be Skipped. . . . 150

Configuration of Incoterms. . . . 152
Import Incoterms Task. . . . 152
Payment Terms Filtering Based on the Incoterms Location Field on Requisition Line Items and Invoices for SAP Variants. . . . 153

Asset Management for SAP-ERP Integrated Sites. . . . 154

Export of Purchase Orders. . . . 156
Export of Attachments and Comments with Purchase Orders. . . . 157
Purchase Order Import Integration with ERP Systems. . . . 158
Purchase Order Import Integration with SAP ERP. . . . 158
Purchase Order Import Integration with PeopleSoft. . . . 159
Import Purchase Order Close Status Task. . . . 159
Workflow of Failed ERP Purchase Order Export. . . . 160
Exporting Purchase Orders. . . . 161
Export Legal Fields in Purchase Order cXML sent to SAP Business Network. . . . 161
Purchase Order Data Files. . . . 162
Purchase Order Header Data Reference. . . . 162
Purchase Order Detail Data. . . . 164
Purchase Order Split Accounting Data. . . . 167
Purchase Order Balance Amount Data. . . . 169
Resend Purchase Orders Stuck in **Ordering** or **Canceling** State. . . . 171
Resending Stuck Purchase Orders. . . . 172

Export of Financial Transaction Information. . . . 174
Financial Information in the Buying Solution. . . . 175
Financial Data Extract Event Types. . . . 175

Considerations for Exporting Financial Information. 176
Exporting Financial Information. 177
Loading Financial Extract into an ERP System Database. 177
Financial Data in FinancialExtractExport.csv. 177
FinancialExtractExport.csv: Accounting Columns for SAP ERP. 179
FinancialExtractExport.csv: Accounting Columns for PeopleSoft ERP. 179
FinancialExtractExport.csv: Accounting Columns for Generic ERP. 180

Configuration of Messaging. 181
Import Messaging Topics Task. 181
Import Conversation Context Rules for Messaging Task. 182
Configuring the Messaging Setup. 189

Configuration and Management of PCards. 191
About Ghosted PCards. 192
Creating a Ghosted PCard. 192
Individual PCards. 193
Accounting PCards. 193
PCard Charges. 193
PCards Import. 194
Accounting PCards. 195
Accounting PCards with Existing Pcard Order Methods. 195
Accounting PCard Ranking. 196
Accounting PCard Selection for Split Lines. 198
PCard Considerations. 198
Accounting Associations and Limits Import. 199
Configuring PCard Accounting Associations. 200
Configuring PCard Limits. 201
Accounting Associations and Limits Export. 202
Exporting PCard Accounting Associations. 202
Exporting PCard Limits. 203
Charge Tax Code Mappings Import. 203
Level 2 Charge Import. 204
Automating Level 2 Charge Importing. 205
Level 3 Charges. 205
CDF to cXML Mapping. 206
Examples for Understanding CDF to cXML Mappings. 207
Reconciliation of Level 3 Charges. 214
Working with PCards. 216
Viewing Information About PCards. 217
Modification of PCard Numbers. 217
Modifying General PCard Information. 218

Creating New PCards. . . . 219
Assigning PCards to Users. . . . 220
Associating PCards with Supplier Locations. . . . 220
Deleting PCards. . . . 220
Enabling Additional Users to Reconcile PCard Charge Exceptions. . . . 221
Specifying a User as a PCard Charge Reconciler. . . . 221
Specifying the Exception Types That Users Can Reconcile. . . . 222
PCard Adjustment Types. . . . 222
Adding a PCard Adjustment Type. . . . 222
Enhanced Charge Reconciliation for PCard Orders. . . . 223
Enabling Enhanced Charge Reconciliation for Purchasing Card Orders. . . . 224

Configuration and Management of Virtual Cards. . . . 225
Scheduled Tasks Used for Virtual Card Support. . . . 227
Virtual Cards. . . . 227
Buyer Account for the Virtual Card Account – Process Flow. . . . 228
SAP Buyer Account for Virtual Cards. . . . 228
Setting Up a Buyer Account for a Virtual Card. . . . 229
Viewing Details of a Virtual Card Account for Buyers. . . . 233
Modifying the Details of a Buyer Account for the Virtual Card. . . . 233
Master Data to SAP. . . . 234
Virtual Card Support for Suppliers. . . . 234
Enabling Virtual Card Support for New Suppliers. . . . 235
Enabling Virtual Card Support for Existing Suppliers. . . . 236
Import Virtual Card Configuration Details to the Vendor Location. . . . 237
Procurement Using a Virtual Card Order – Process Flow. . . . 238
SAP Ariba Virtual Card Order Method. . . . 239
Virtual Card Order. . . . 239
Virtual Card Details for Suppliers. . . . 240
Charge Reconciliation and Payment – Process Flow. . . . 240
Charge a Virtual Card and Perform Charge Reconciliation Against the Virtual Card Order. . . . 241
Importing Charges. . . . 242
Viewing the Details of Virtual Card Charges. . . . 243
Creating the Virtual Card Charge Reconciliation Document. . . . 244
Reconciling Charges. . . . 244
Exporting Virtual Card Charge Payment Requests to the Buyer's ERP system. . . . 245

Ad Hoc Vendors for Non-Catalog Items in Virtual Card Orders. . . . 246
Process for Supporting Ad Hoc Vendors for Non-Catalog Items in Virtual Card Orders. . . . 247
Importing Ad Hoc Vendors to SAP Ariba Buying Solutions. . . . 247
Adding an Ad Hoc Vendor and Vendor Location in a Purchase Requisition. . . . 248
Deactivating Supplier Locations Associated with Ad hoc Vendors 250

Charge Reconciliation. 251
Skip Validation Feature for Charge Reconciliation. 251
Auto-Accept Feature for Charge Reconciliation. 252
Charge Exception Types. 252
Data Import Tasks for Charge Exception Types. 253
Import Charge Exception Types Task. 253
Translations for Charge Exception Types. 254
Import Charge Exception Type Validation Data Task. 254
Charge Exception Type Validation Data to Avoid PO Variance Exceptions. 255
Adjustment Types. 255
Data Import Tasks for Adjustment Types. 256
Import Adjustment Type Task. 256
Adjustment Types in Ariba Administrator 256

Charge Payment Requests in SAP Ariba Buying and Invoicing 257
Charge Payment Request Data Export Tasks. 257
Adding Custom Accounting Fields to Charge Payment Export. 258
Exporting Charge Payment Requests. 259
Charge Payment Request Data for PeopleSoft ERP. 259
Charge Payment Header Data for PeopleSoft ERP. 260
Charge Payment Detail Data for PeopleSoft ERP. 261
Charge Payment Bank Data for PeopleSoft ERP. 262
Charge Payment Split Accounting Data for PeopleSoft ERP. 263
Charge Payment Request Data for SAP ERP. 263
Charge Payment Header Data for SAP ERP. 264
Charge Payment Detail Data for SAP ERP. 265
Charge Payment Request Data for Generic ERP. 266
Charge Payment Header Data for Generic ERP. 267
Charge Payment Detail Data for Generic ERP. 268
Charge Payment Bank Export Data for Generic ERP. 269
Charge Payment Split Accounting Data for Generic ERP. 270
Charge Payment Detail with Split Accounting Data for Generic ERP. 271

Configuration of Receiving. 273
Receiving Types. 274
Amount Thresholds for Auto-Receiving. 276
Amount Thresholds Based on Part Number or Commodity Code. 276
Amount Thresholds for All Orders and Line Items. 277
Workflow for Amount Thresholds. 277
Example:Auto-Receiving by Part Number. 278
Example: Receiving a Purchase Order Based on Commodity Code. 278
Example: Auto-Receiving All Non-Catalog Items. 279

Asset Data Information on Receipts. . . . 279
Receiving Tolerances. . . . 280
Tolerance Operators. . . . 281
Configuration of Receiving Tolerance at the Purchasing Unit Level. . . . 281
Under and Over-Receiving Attributes. . . . 282
Data Import Tasks for Receiving Types. . . . 283
Receiving Type Values in CSV Files. . . . 283
Import Receiving Types by Commodity Code. . . . 284
How Commodity Code Hierarchy Works. . . . 285
Configuration of Receiving Types Based on Fields in the Requisition. . . . 286
Importing Receiving Types by Part Number. . . . 289
Using Wildcards for Part Numbers. . . . 290
ReceiptByPartNumber.csv: PeopleSoft Example. . . . 290
ReceiptByPartNumber.csv: SAP Example. . . . 290
ReceiptByPartNumber.csv: Generic Example. . . . 291
Managing Receiving Types by Commodity Code. . . . 291
How to View Receiving Type Rules for Commodity Codes. . . . 291
How to Add or Modify Receiving Type Rules for Commodity Codes. . . . 292
How to Delete Receiving Type Rules for Commodity Codes. . . . 293
Managing Receiving Type Rules by Part Number. . . . 294
Configuring Auto-Receiving Lead Times for Non-Catalog Items. . . . 297
Managing Asset Receiving. . . . 297

Imported Receipts. . . . 299
ReceiptImport.csv File for Receipt Import. . . . 300
ReceiptImport.csv: Generic ERP Example. . . . 300
Receipts Import. . . . 302
Status of Import Receipt. . . . 303
Importing Receipts with Custom Fields. . . . 304
Conditions When Receipt Import May Fail. . . . 304
Receiving on Imported Receipts. . . . 305
Export Receipt Status Task. . . . 306
Receipt Import Error Codes Reported by Export Receipt Status. . . . 307
Exporting Receipts to SAP ERP. . . . 307
Receipt Export Errors. . . . 307
Asynchronous Import of Receipts. . . . 308

Export of Receipt Data. . . . 309
Receipt Integration with ERP Systems. . . . 309
Workflow of a Failed Receipt Export. . . . 310
Exporting Receipts. . . . 311
Receipt Data. . . . 312

Receipt Header Data. 312
Receipt Detail Data. 313
Receipt Asset Data. 314

Export of Accrual Information. 316
Accrual Information in Accrual Export. 317
Transaction Amount in Accrual Export. 317
Encumbrance Transactions in Accrual Export. 318
Considerations for Exporting Accrual Information. 318
Export Accrual Transactions Data Export Task. 318
Exporting Accrual Information. 319
Loading Accrual Information into an ERP System Database. 319
Export Transaction Accruals Data Export Task. 320
Common Data in AccruedTransactionExport.csv. 320
Accounting Data in AccruedTransactionExport.csv. 322

Support for the Plant Entity from SAP Backend Systems. 324
Workflow for Defaulting Based on the Plant Entity. 324
Setting the Plant Field in the User Profile. 325
Setting the Value for the Plant Field for a User from the User Manager Workspace. 325
Default Purchasing Organization Based on the Selected Plant. 326

Integration of Orders and Return Receipts for Goods Receipt-Based Invoice Verification. 328

Data Import Tasks Reference. 330
Data Import Tasks for Procurement Data. 330
About SAP Ariba Buying solutions Data Import Order. 331
Data Import Order for PeopleSoft. 331
Data Import Order for SAP ERP. 334
Data Import Order for Generic. 337
Data Import Tasks for SAP Ariba Buying 339
Data Import Order for SAP Ariba Buying 343
PeopleSoft Data Import Order for SAP Ariba Buying 343
SAP Data Import Order for SAP Ariba Buying 345
Generic Data Import Order for SAP Ariba Buying 346

CSV Files Reference. 349

Procurement Data Import and Administration Guide

This guide is for SAP Ariba administrators responsible for defining ordering methods, configuring receiving types, and importing procurement transaction data.

> **Note**
>
> Before you define and import procurement data, you must define and import common master data such as users, suppliers, and accounting information. See Common Data Import and Administration for SAP Ariba Procurement Solutions.

This guide applies to:

- SAP Ariba Buying
- SAP Ariba Buying and Invoicing

Related Guides

Common Data Import and Administration for SAP Ariba Procurement Solutions

Administration and Data Maintenance Fundamentals

Guided Buying Administration

Tax and Charge Configuration

Invoicing Data Import and Administration Guide

Purchasing Guide for Procurement Professionals

SAP Ariba Procurement Solutions Group Descriptions

Customer Catalog Administration Guide for Buyers

Approval Process Management Guide

Introduction to Defining and Importing Procurement Data

You define, import, and maintain procurement data for use in SAP Ariba Buying and SAP Ariba Buying and Invoicing.

Prerequisites for defining and importing procurement data

- You should be familiar with the Administration and Data Maintenance Fundamentals guide for information on:
  - The data import process
  - Entering data in comma-separated-value (CSV) files
  - Downloading and using the Data Dictionary
  - Using Ariba Administrator
- You must define and import common data for your buying solution before you define and import procurement data. For more information about managing common data, see Common Data Import and Administration for SAP Ariba Procurement Solutions.
- If your buying solution is SAP Ariba Buying and Invoicing, make sure that you define and import invoicing and payment data. For more information about managing invoicing and payment data, see Invoicing Data Import and Administration Guide.
- The order in which you import common data depends on the SAP Ariba solutions you use. For more information about data import order, see Common Data Import and Administration for SAP Ariba Procurement Solutions.

Related Information

Administration and Data Maintenance Fundamentals
Invoicing Data Import and Administration Guide
Common Data Import and Administration for SAP Ariba Procurement Solutions

Working with Procurement Data

Working with procurement data involves configuring and managing data, configuring and managing charges, and configuring and managing receiving types, and exporting receipts.

The following table lists each type of procurement data and identifies the SAP Ariba solution in which it is used:

Type of data	SAP Ariba Buying and Invoicing	SAP Ariba Buying
Terms and conditions text for purchase orders	X	X
Budget data	X	X
Requisitions collaboration information	X	X
Order methods	X	X

The following table lists each type of charge data and identifies the SAP Ariba solution in which it is used.

Type of data	SAP Ariba Buying and Invoicing	SAP Ariba Buying
Purchase cards	X	X
Charges	X	X
Charge exception types	X	X

The following table lists the transactional data that can be exported and identifies the relevant SAP Ariba solution.

Type of data	SAP Ariba Buying and Invoicing	SAP Ariba Buying
Receiving types	X	X
Receipts	X	X
Exporting receipts	X	X

Related Information

Administration and Data Maintenance Fundamentals

Configuration and Management of Budgets in Your Buying Solution

Organizations define budgets to control the expenses that they incur when procuring material or services. Budgets are defined and implemented at various organizational and accounting levels, such as Projects, GL Accounts, Cost Centers, and Internal Orders, besides a combination of many of these levels within companies.

A budget entity comprises of a collection of budget items. Each budget item is a unique combination of accounting fields and a budget period, with an allocation amount that is defined for each such combination.

Depending on how your buying solution is configured, you can validate requisitions for budgets using one of the following features:

- Budget checks on requisition items against budget data loaded from the ERP system into your buying solution: In this case, because budget data is stored in the ERP system and in the buying solution, it needs to be synchronized between both systems.
- Budget checks in external ERP systems: Refers to real-time budget checks to validate requisitions against budget information hosted in an external ERP system. In this case, the budget data is stored only in the external system.

> **Note**
>
> For sites that are enabled for both budget checking within your buying solution and budget checking in external systems, the ability to perform budget checks within your buying solution is automatically disabled starting from the date that is specified to enable budget checking in external systems.

The information in these topics is applicable to configuring budget checks within your buying solution. For information about configuring budget checks in an external ERP system, see Budget Checks in External Systems [page 43].

Budget Checks in Your Buying Solution [page 15]

Budget Import [page 15]

Budget Check Configuration [page 22]

Budget Check Configuration During Final Approval of Requisitions [page 23]

Budgets Validation [page 29]

Ranking Budgets [page 30]

Performing a Budget Search [page 31]

Budget Adjustments [page 33]

Budget Updates [page 35]

Import Tasks for Closing and Reopening Budgets [page 36]

Incremental Load of Budgets [page 38]

Budget Export [page 41]

Budget Transactions Export [page 41]

Budget Checks in Your Buying Solution

When the budget check feature is enabled on your site, it provides you control over spends against established budgets and helps you restrict spends that are not associated with budgets.

The budget check feature within buying solution includes certain validations to ensure there is an adequate budget available and is associated to a requisition when the requisition is created. These validations also ensure the requisition is submitted for approval only when the budget check is successful.

The budget check feature is a standard functionality in SAP Ariba Buying and Invoicing, and an optional, fee-based service in SAP Ariba Buying. If you are an SAP Ariba Buying customer and are interested in the budget check functionality, ask your Designated Support Contact to log a service request and an SAP Ariba Customer Support representative will contact you.

As a budget manager, you can create, import, export, search, and review budgets in the supported buying solution. At the time of creating the budget, you must define all the required budgeting fields, including the header, line item, allocation, and periods values. All budget fields are validated during budget checking.

If you are a member of the **Budget Manager** group, you can access the budget check feature from the **Budget Manager** workspace in Ariba Administrator.

Budget Import

You can import or load the budget information in your buying solution by running data import tasks either from the **Core Administration** option in the user interface or using the SAP Ariba data transfer tool.

> ⓘ Note
>
> The budget data file that you import must be less than 10,000 records. If the records are greater than 10,000, it is recommended that you split the data into multiple files with a maximum limit of 10,000 records. Then, run the import task separately for each file.

The following table lists the procurement data import tasks for budgets and indicates if there is a corresponding feature in the SAP Ariba integration toolkit. Some import tasks are deprecated, and should not be used. Deprecated import tasks include the word "Deprecated" in parentheses as part of the import task name as it appears on the user interface.

Import Task	Integration Toolkit Name	Description
Import CSV Budget in 4 file format		This task loads the budget data in the CSV file format from a single zip file. You can also run this task to perform an incremental budget data import.
Import CSV Budget in 4 file format (Deprecated)		Do not run this task. This task has been deprecated and replaced with Import CSV Budget in 4 file format.
Import Excel Budget		This task loads the budget data in the Excel file format. You can also run this task to perform an incremental budget data import.

Import Task	Integration Toolkit Name	Description
Import Excel Budget (Deprecated)		Do not run this task. This task has been deprecated and replaced with Import Excel Budget.
Import Budget Adjustments		This task allows you to import adjustment data to existing budgets.
Import Budget Code Status		This task updates the budget code status. Run this task to open or close an existing budget code. If a budget code is closed, no new requisitions can reserve funds from that budget code in any of the associated budget periods.
Import Budget Period Status		This task updates the budget period status. Run this task to open or close an existing budget period. If a budget period is closed, no new requisitions can reserve funds from any budget code that is associated with the closed budget period.
Import Budgets in ZIP File Format (Deprecated)	Import Budgets	Do not run this task. This task has been deprecated and replaced with Import Budgets in ZIP File Format.
Import Budgets in ZIP File Format	Load Budgets	This task imports budget data, and is similar to the Import CSV Budget in 4 file format, but uses files in the ZIP format to load budget data.
Import Budget Adjustments in ZIP File Format	Load Budget Adjustments	This task allows you to import adjustment data to existing budgets. It is similar to the Import Budget Adjustments task, but uses files in the ZIP format to load budget adjustments.
Import Budget Code Status in ZIP File Format	Import Budget Code Status	This task is similar to the Import Budget Code Status task, but uses data files in the ZIP format to update the budget code status.
Import Budget Period Status in ZIP File Format	Import Budget Period Status	This task is similar to the Import Budget Period Status task, but uses data files in the ZIP format to update the budget period status.

- In Ariba Administrator, go to the **Budget Manager** workspace and choose the appropriate import task to load your budget data:
- In the SAP Ariba data transfer tool, run a transactional data upload operation and specify the correct event name (specified in the **Integration Toolkit Name** column) in the `event` option.

> **Note**
>
> Budget data can be successfully imported only when all field (column headers) in the CSV files are in English. Non-English field names are not supported.

Related Information

About Performing Incremental Load of Budgets [page 38]

Uploading Budget Data Using Microsoft Excel

You can load budget data from an Excel spreadsheet.

Procedure

1. On the dashboard, click **Manage** > **Core Administration** .
2. Click **Budget Manager** and select **Data Import/Export** in the Budget Manager workspace.
3. Select the **Import Excel Budget** import task.
4. In the **Data Import Task** page, enter or navigate to the location of the Excel sheet containing the budget data, and click **Run**.

Budget Data Formatting in CSV Files

Before importing budget information from an external ERP system, you must format it in four separate CSV files.

The following table lists the CSV files you use to format budget data.

CSV File	Description
`BudgetHeader.csv`	Contains header information and can have only one record. This file provides the budget title.
`BudgetLines.csv`	Contains line item information. These values are unique budget line identifiers (also called budget codes) for each line item in the requisition. Each budget line in this file contains a unique combination of accounting fields, such as cost center, company code, general ledger, and so on.
`BudgetPeriods.csv`	Contains unique accounting periods (time period) for all budget lines (or budget codes).

CSV File	Description
`BudgetAllocations.csv`	Contains information on the amount set aside for a budget line specific to a time period, along with the tolerance values corresponding to each budget line. **Note** You can specify a null value or zero (0) as the allocation amount. This file allows you to associate budget codes with the accounting periods, and defines the allocation for each budget code. **Note** A budget code cannot have overlapping accounting periods. This file also defines the tolerance for the budget period. `Allocated Amount:` This field indicates the total amount that is allocated to a particular budget code, for the corresponding accounting period. `Available Amount:` This field indicates the available amount for a particular budget code. If this field does not have any values, your buying solution automatically picks up the value specified in the **Allocated Amount** field. If there are requisitions or POs that have consumed from a budget code and period, those amounts are deducted from the allocated amount to arrive at the available funds for all future purchases.

Sample CSV File Formats for Generic ERP

Generic ERP sample CSV budget definition files include `BudgetHeader.csv`, `BudgetLines.csv`, `BudgetPeriods.csv`, and `BudgetAllocation.csv`.

Following is an example of `BudgetHeader.csv`:

```
    UTF8

    Cp1252

    Title
Demo Budget-4
```

Following is an example of `BudgetLines.csv`:

```
    UTF8

    Cp1252
```

```
LineNumber,BudgetCode,Company,BusinessUnit,CostCenter,Period,Project,Region,SubAccou
nt,Rank,Budget Check Disabled

1,BC1,04,US015,6400,,,,1,False
2,BC2,00,US012,1200,,,,2,False
```

Following is an example of `BudgetPeriods.csv`:

```
     UTF8


     Cp1252


     FiscalYear,AccountingPeriod,StartDate,EndDate

2008,Q108,1/1/2008 00:00:00 +0530,3/31/2008 23:59:59 +0530
2008,Q208,4/1/2008 00:00:00 +0530,6/30/2008 23:59:59 +0530
2008,Q308,7/1/2008 00:00:00 +0530,9/30/2008 23:59:59 +0530
2008,Q408,10/1/2008 00:00:00,12/31/2008 23:59:59
```

> ### Note
>
> The date format to be followed is `MM/DD/YYYY HH:mm:ss zzz`, where `zzz` is the GMT offset.

Following is an example of `BudgetAllocations.csv`:

```
     UTF8


     Cp1252



LineNumber,BudgetCode,FiscalYear,AccountingPeriod,AllocationAmount,Currency,Percenta
geTolerance,AvailableAmount,AvailableCurrency

1,BC1,2008,Q108,5010,USD,10,,
1, BC1,2008,Q208,5020,USD,10,,
1, BC1,2008,Q308,5030,USD,10,,
1, BC1,2008,Q408,5040,USD,10,,
2, BC2,2008,Q108,6010,USD,10,,
2, BC2,2008,Q208,6020,USD,10,,
2, BC2,2008,Q308,6030,USD,10,,
2, BC2,2008,Q408,6040,USD,10,,
```

Sample CSV File Formats for PeopleSoft ERP

PeopleSoft ERP sample CSV budget definition files include `BudgetHeader.csv`, `BudgetLines.csv`, `BudgetPeriods.csv`, and `BudgetAllocation.csv`.

Following is an example of `BudgetHeader.csv`:

```
     UTF8
```

```
     Cp1252


     Title
Ariba Budget -psoft1
```

Following is an example of `BudgetLines.csv`:

```
     UTF8


     Cp1252



LineNumber,BudgetCode,ProcurementUnit,Account,BusinessUnit,GLBusinessUnit,Location,D
epartment,
Product,Project,Rank,Budget check Disabled

1,BC1,,,BUY01,FRAE1,,,,2,False
2,BC2,,,BUY02,EGV01,,,,1,False
```

Following is an example of `BudgetPeriods.csv`:

```
     UTF8


     Cp1252


     FiscalYear,AccountingPeriod,StartDate,EndDate

2008,M108,1/1/2008 00:00:00 +0530,1/31/2008 23:59:59 +0530
2008,M208,2/1/2008 00:00:00 +0530,2/29/2008 23:59:59 +0530
2008,M308,3/1/2008 00:00:00 +0530,3/31/2008 23:59:59 +0530
2008,M408,6/1/2008 00:00:00,6/30/2008 23:59:59
```

> **Note**
>
> The date format to be followed is MM/DD/YYYY `HH:mm:ss zzz`, where `zzz` is the GMT offset.

Following is an example of `BudgetAllocations.csv`:

```
     UTF8


     Cp1252



LineNumber,BudgetCode,FiscalYear,AccountingPeriod,AllocationAmount,Currency,Percenta
geTolerance,AvailableAmount,AvailableCurrency

1,BC1,2008,Q108,5010,USD,5,,
1, BC1,2008,Q208,5020,USD,5,,
1, BC1,2008,Q308,5030,USD,5,,
1, BC1,2008,Q408,5040,USD,5,,
2, BC2,2008,Q108,6010,USD,5,,
2, BC2,2008,Q208,6020,USD,5,,
```

```
2, BC2,2008,Q308,6030,USD,5,,
2, BC2,2008,Q408,6040,USD,5,,
```

Sample CSV File Formats for SAP ERP

SAP ERP sample CSV budget definition files include `BudgetHeader.csv`, `BudgetLines.csv`, `BudgetPeriods.csv`, and `BudgetAllocation.csv`.

Following is an example of `BudgetHeader.csv`:

```
     UTF8


     Cp1252


     Title
SAP Demo Budget-2
```

Following is an example of `BudgetLines.csv`:

```
     UTF8


     Cp1252


     Line
Number,BudgetCode,PurchaseOrg,PurchaseGroup,CompanyCode,CostCenter,GeneralLedger,WBS
Element,Asset,SubNumber,InternalOrder,ProjectNumber,Rank,Budget Check Disabled

1,BC1,,,3000,0000004230,0000894020,,,,,1,False
2,BC2,1000,0000001210,0000231000,,,,2,False
```

Following is an example of `BudgetPeriods.csv`:

```
     UTF8


     Cp1252


     FiscalYear,AccountingPeriod,StartDate,EndDate

2008,Q108,1/1/2008 00:00:00 +0530,3/31/2008 23:59:59 +0530
2008,Q208,4/1/2008 00:00:00 +0530,6/30/2008 23:59:59 +0530
2008,Q308,7/1/2008 00:00:00 +0530,9/30/2008 23:59:59 +0530
2008,Q408,10/1/2008 00:00:00,12/31/2008 23:59:59
```

> **Note**
>
> The date format to be followed is MM/DD/YYYY `HH:mm:ss zzz`, where `zzz` is the GMT offset.

Following is an example of `BudgetAllocations.csv`:

```
     UTF8


     Cp1252


     LineNumber,BudgetCode,FiscalYear,AccountingPeriod,Allocation Amount,Currency
1,BC1,2008,Q108,5010,USD,10,,
1, BC1,2008,Q208,5020,USD,10,,
1, BC1,2008,Q308,5030,USD,10,,
1, BC1,2008,Q408,5040,USD,10,,
2, BC2,2008,Q108,6010,USD,10,,
2, BC2,2008,Q208,6020,USD,10,,
2, BC2,2008,Q308,6030,USD,10,,
2, BC2,2008,Q408,6040,USD,10,,
```

Budget Check Configuration

Your buying solution can be configured to perform a budget check for every purchase at your site. The budget check feature is not enabled on your customer site in the default configuration.

Contact your SAP Ariba Customer Support to enable this feature, and provide them with the budget activation date. When the activation date is specified and the budget check feature is enabled, it means that every requisition when submitted, is subjected to a budget validation if the specified activation date is prior to that of the date and time of submission of the requisition. The budget check is not performed if the activation date is either not specified or is in the future.

Approval Rules and Budget Checking

If your site enables budget checking, you can use three approval rules that are available and enabled by default to facilitate your budget check approval process.

> **Note**
>
> In sites where budget check is enabled, when a budget check is done on collaborative requisitions, the approval graph is redrawn.

Budget Check Configuration During Final Approval of Requisitions

When budgets are commonly available in both the buying solution and the external ERP system, budgets are consumed independently to approve requisitions and create purchase orders. In such scenarios, it is possible that the availability of funds as shown in these two systems is different from each other.

Because the external ERP system is designated as the source of truth for budget data, it is important that the two systems are kept in sync with each other by frequently loading the latest available funds or the adjustments from the external ERP system into the buying solution.

If at any point of time, the two systems are not in sync, it may lead to situations where the buying solution finds and reserves the required funds when a user submits the requisition, but these funds are no longer available when the requisition is fully approved and a purchase order is raised. When such a purchase order is processed, the ERP system is likely to reject the purchase order because of insufficient funds. This problem becomes significant when the approval process takes a long time.

To avoid this scenario, you can configure your site to perform a budget check to reconfirm the availability of funds in the budgets associated before allowing the final approver to approve a requisition.

To configure this feature, have your Designated Support Contact log a service request and an SAP Ariba Customer Support representative will follow up to complete the request.

When you have configured this feature on your site, the final approver cannot approve a requisition against a budget for which:

- The ordered amount is already greater than the maximum funds (Allocated + Tolerance + Adjustments) that can be utilized, or
- The current requisition amount if approved, will cause the ordered amount to exceed the maximum amount that can be utilized

> **Note**
>
> The budget calculation during the final budget reconfirmation step does not include the amounts reserved for the requisition being approved, or any other requisitions associated with that budget.

This budget re-confirmation is done only when the last approver attempts to approve the requisition. When the error is displayed, the approver can then either deny the requisition, or manually add a Budget Manager or a Budget Approver to the approval flow.

> **Note**
>
> If the approver adds a Budget Manager or Budget Approver to the approval flow, that approver will no longer be the last approver, and so, will be able to approve the requisition. The Budget Manager or Budget Approver (that is added to the approval flow) must then attempt to appropriately resolve the error and complete the approval process.

Budget Synchronization with an External ERP System

Budgets that are loaded into the buying solution are also present in the external ERP systems. When the same budget is consumed independently in both the external ERP system as well as the buying solution, the budget data is required to be synchronized in both systems.

A discrepancy in the budget data between the two systems can lead to errors in accumulation and allocation of budgets.

For example, consider a scenario where a budget BC001 with an allocation of $1000 is present in both the buying solution and an external ERP system.

First, a requisition is created in the buying solution, approved, and a PO (PO-01) is generated for $100. The available amount now in the buying solution is $900. When this data is exported to the external ERP, the purchase order consumes $100 from the same budget. The available amount for BC001 is $900 in both the buying solution and the ERP system.

Later, a purchase order (ERP_PO_01) for $150 is generated in the external ERP for the same budget code BC001. When the ERP purchase order consumes the budget, the available amount for BC001 in the ERP is $750, whereas the same budget code reflects an available amount of $900 in the buying solution.

When you export this budget data back to the buying solution for adjustments, the allocated amount becomes $750; the buying solution then deducts the consumed amount ($100 for PO-01) again and updates the available amount for the budget to $650, which is incorrect. This happens because every time the budget is adjusted or changed, the buying solution considers the new amount as the allocated amount and then calculates the available amount as:

Available Amount = Allocated Amount - [Consumed Amounts (from purchase orders) + Reserved Amounts (of requisitions)]

In scenarios such as this, the purchase order amounts sent to the ERP are subtracted twice from the original allocated amount, resulting in incorrect available amounts in the buying solution.

To ensure such discrepancies do not occur between the buying solution and external ERPs, you must have both systems updated and synchronized with each other.

To synchronize budget data, update the **Available Amount** field in the `BudgetAllocations.csv` file (or Excel sheet) and re-import the budget into the buying solution by running the `Import CSV Budget in 4 File Format` or `Import Excel Budget`.

The buying solution adds the amounts of all the purchase orders that have been consumed for that budget to this new available amount value (coming from the ERP system) and sets this sum total as the allocation amount.

Using the example, when the budget is re-imported after setting the **Available Amount** field (available amount of $750), the buying solution adds the consumed amount (of $100) to this available amount and sets the allocation amount correctly to $850.

Now the consumed purchase order amount is subtracted only once from the allocated amount.

This is the recommended approach to synchronize budget data between thebuying solution and an external ERP system.

> **Note**
>
> The `Available Amount` field is not mandatory. But, if the `Available Amount` is specified in the CSV file, the buying solution considers only this value and not the allocation amount for that budget code. after the

budget has been imported with the available amount value for the budget codes, the new allocation amount is calculated as:

```
Allocation Amount = Available Amount + Ordered Amount
```

Enabling Budget Check at a Customer Site That Is Already in Production

Enabling the budget check feature at a customer site that is already in production should to be handled with care. In addition to defining the master data and performing regular updates to the master data, you must define the processes that determine how budgets are managed and updated, and how exceptions are handled.

Because the previously existing requisitions are not checked against a budget, you need to adjust the budgets for such requisitions outside the procurement system and upload only the remaining budgets. In such a scenario, ensure you provide the budget activation date (to SAP Ariba Customer Support) to the date on which budgets are loaded into the buying solution. Requisition submitted prior to the budget activation date do not consume the budgets.

> **Note**
>
> If requisitions submitted prior to the budget activation date are changed, the budget check is performed only for the changed items in the requisition, that is, if the split line amount has increased in amount, the budget is then consumed only for the increased amount, but if the amount has decreased, there is no impact on the associated budget.

You can choose to set the budget activation date to a date in the future as well. In such a scenario, the budget check is performed only after system date crosses the budget activation date.

Disabling Budget Check for an Accounting Combination

You can choose to disable budget check for your entire site or for particular accounting combinations.You might do this when you do not want a budget approver automatically added to a requisition workflow when a matching budget is not found.

When you enable the budget check feature, it is applicable to all requisitions that are created in the buying solution. This means that whenever a requisition is submitted, the buying solution first tries to match a suitable budget code to it, based on the accounting combinations. If a match is found, it then validates (checks) the requisition against the matched budget code to reserve the required amount for that purchase.

If a suitable match is not found, the budget check fails with a warning, stating that no match was found. Users can still submit requisitions, but in such scenarios, a Budget Approver is added to the requisition approval flow.

This behavior, of failing the budget check and automatically adding a budget approver to the requisition workflow when a matching budget is not found, is driven by a parameter that is enabled in the default configuration.

You can choose to turn off this parameter by contacting SAP Ariba Customer Support. When this parameter is turned off, the buying solution does not display any warning when there is no budget match; the budget code, when disabled, is not consumed by any requisition; and a budget approver is also not added to the workflow.

If you want to have the parameter enabled for the entire customer site in general, but choose to disable the budget match for particular accounting combinations (budget codes), you must specify the appropriate value (Yes or No) against the corresponding budget codes, in the `BudgetLines.csv` file and import the budget.

For example, if you have loaded five budget codes (BC1, BC2, BC3, BC4, and BC5) into the buying solution, but do not want the budget match to be performed for budget codes BC2 and BC3, then you must specify the appropriate value (in the `Budget Check Disabled` field) against these budget codes, as shown in the below extract:

```
UTF8

Cp1252


LineNumber,BudgetCode,Company,BusinessUnit,CostCenter,Project,Region,SubAccount,Rank
,Budget Check Disabled

1,BC1,04,US015,6400,,,,1,No
2,BC2,00,US012,1200,,,,2,Yes
1,BC3,04,US015,6400,,,,1,Yes
1,BC4,04,US015,6400,,,,1,
1,BC5,04,US015,6400,,,,1,
```

Note that in the example, BC4 and BC5 have a null value in the `Budget Check Disabled` field.

> **Note**
>
> By default, the buying solution interprets a null value in this column as `No` .

After configuring and re-importing the budgets, you can view the updated budget information in the Budget Manager workspace.

Reservation of Funds for Requisitions

If the site is enabled for the budget check, whenever a requisition is submitted, it is matched against the existing budgets. To reserve funds, every split line (or line item if the split accounting is not used) of the requisition is matched to a unique budget.

If sufficient funds are found in the budget(s), the funds are reserved and it is not available for any other purchase. If funds are not available, a budget approver is added to the approval flow and this budget approver can take an appropriate action on it. When this requisition is approved and one or more purchase orders are created from the requisition, the funds that are reserved are ordered. If the requisition is changed at any point in time, the budget check is performed again on the changed portion of the requisition. If the requisition is canceled, the reserved or encumbered funds are released back.

Every budget has an allocated amount for an accounting period. Additionally, budgets can have a tolerance percentage set. The total allocated amount for an accounting period is the allocated amount specified for the accounting period plus the tolerance amount. The total available amount for the budget period is derived as follows:

Available Amount in a period = Allocated Amount + Adjustment Amount - [Reserved Amount (from requisitions) + Consumed Amount (from purchase orders)]

This table provides a description of the key terms associated with budgets:

Term	Description
Available Amount	The amount of money that is still available in a budget.
Allocation Amount	The total amount of money that is allocated to be spent for a budget.
Tolerance	The amount that can be spent in addition to the allocated amount for a budget. For example, if the allocated amount for a budget line is $100.00, and the tolerance is defined as $10.00, the budget check will be successful until a spend of $110.00 (100+10, which is the tolerance amount). The budget check will fail if the total spend for that budget crosses this tolerance amount. The tolerance value you specify here takes precedence over the tolerance value set while configuring this feature (at the parameter level).
Adjustment Amount	Amount adjustments that have been made to a budget.
Reserved Amount	The amount that is reserved from the allocated amount against a requisition.

Limitation on Reserving Funds for Requisitions

By default, the value in the **Available Amount** field in `BudgetAllocations.csv` is used for budget validations against requisitions. If **Available Amount** is unspecified and **Available Currency** is specified, the buying solution considers the **Available Amount** to be 0 even if you specify a value in **Allocation Amount** for a budget line.

To work around this limitation, we recommend that you ensure one of the following:

- Specify valid values in both **Available Amount** and **Available Currency** fields.
- If you do not want to specify the **Available Amount**, then ensure the **Available Currency** field is NOT defined.

If you have not specified values in the **Available Amount** and **Available Currency** fields, but specified the **Allocation Amount**, then the buying solution automatically considers the **Allocation Amount** value to be the **Available Amount**.

Release of Funds from Reserved Budgets for Closed Purchase Orders

You can configure your site to release unconsumed funds reserved by a purchase order when the purchase order is closed with a close order action **Close for Receiving** or **Close All Actions** before it is fully received.

For example, if a purchase order for $100 is partially received (created receipt for $70) and the user decides to close this order for further receiving, or for all actions, this feature (if enabled) releases the unconsumed amount of $30 back to the associated budget, to be used by any other purchase order.

If this released amount is utilized by another purchase order, then the closed purchase order (that was partially received for $70 in this example) cannot be reopened, unless the buying solution can find $30 available in the same budget and period.

Releasing unconsumed funds is disabled by default. To enable this feature, have your Designated Support Contact log a service request so an SAP Ariba Customer Support representative follows up to complete the request.

Limitation on Release of Reserved Budgets for Closed Purchase Orders

If a requisition is created before the budget activation date (that is, before the budget check feature has been enabled on your site), then the unconsumed amount (if any) on the closed order will not be released from the purchase order back to the matched budget.

Use Case Scenarios

There are four examples of use case scenarios based on purchase order and budget details, user actions, and the result of those actions on the budget.

The following table presents use case scenarios that illustrate the behavior of release of reserved budgets when enabled, and the expected results:

Purchase Order and Budget Details	User Action	Result
Use case scenario 1:		
• Purchase Order: PO1 for $100 • Matched budget code: BC1 • Period: P1 • Allocated Amount: $500 Originally, the ordered amount for PO1 is $100 and the available amount in the matched budget (BC1/P1) is $400.	User partially receives for $70 and closes the order for receiving or closes for all actions.	The unconsumed amount of $30 is released back into the matched budget (BC1/P1), and the available amount in this budget is $430 (available amount - ordered amount).
	PO1 is reopened and is fully received for the remaining amount.	If the amount is still available in that budget, then the available amount in BC1/P1 becomes $400 (available amount of $430 - ordered amount of $30). **Note** If BC1/P1 is closed before the user attempts to reopen PO1, the reopen action is not allowed.
Use Case Scenario 2:		
• Purchase Order: PO2 for $100 • Matched budget code: BC2 • Period: P1 • Allocated Amount: $500 Originally, the ordered amount for PO2 is $100 and the available amount in the matched budget (BC2/P1) is $400.	User partially receives for $70 and closes the order for receiving or closes for all actions.	The unconsumed amount of $30 is released back into the matched budget (BC2/P1), and the available amount in this budget is $430 (available amount - ordered amount).
	A new purchase order PO3 for $425 is matched with BC2/P1 and is in the Ordered state.	The available amount in this budget (BC2/P1) is now $5 (available amount of $430 - ordered amount of $425).
	User attempts to reopen PO2.	PO2 cannot be reopened because the associated budget (BC2/P1) does not have sufficient funds.
Use Case Scenario 3:		
• Purchase Order: PO3 for $100 • Matched budget code: BC3 • Period: P1	User partially receives for $70 and closes the order for receiving or closes for all actions.	The unconsumed amount of $30 is released back into the matched budget (BC3/P1), and the available amount in this budget is $430 (available amount - ordered amount).

Purchase Order and Budget Details	User Action	Result
• Allocated Amount: $500 Originally, the ordered amount for PO3 is $100 and the available amount in the matched budget (BC3/P1) is $400.	User performs a change order action, but does not change the amount value. A new version of the PO is created and is matched to BC3/P1. This order is fully received.	The ordered amount on BC3/P1 is $100 and the available amount is $400.
Use Case Scenario 4:		
• Purchase Order: PO4 for $100 • Matched Budgets: BC4/P1 and BC5/P1 • Allocated Amount: Both budgets having $500 each • PO5 is associated with BC4/P1 Originally, the ordered amount for PO4 is $100 and the available amount in the matched budget (BC4/P1) is $400.	User partially receives for $70 and closes the order for receiving or closes for all actions.	The unconsumed amount of $30 is released back into the originally matched budget (BC4/P1), and the available amount in this budget is $430 (available amount - ordered amount).
	User performs a change order and this time, the new version of PO4 is matched with BC5/P1.	The ordered amount on budget BC4/P1 becomes $0 (zero) and the available amount is $500. The ordered amount on budget BC5/P1 shows $100 (total PO amount) and the available amount is $400.

Budget Check on Imported Requisitions

By default, imported requisitions are not validated against any budget. You can enable the budget check feature for imported requisitions. For help with this configuration, ask your Designated Support Contact to log a service request and an SAP Ariba Customer Support representative will contact you. When enabled, imported requisitions go through budget check validations as do requisitions created in the buying solution.

Budgets Validation

An imported budget is validated against a number of conditions.

Validation occurs for the following conditions:

1. Overlapping accounting periods: The buying solution does not allow you to configure overlapping accounting periods for the same budget code.
   That is, the same calendar date cannot fall in any two different time periods for the same budget code.
   For example, define budget codes `B001` and `B002`, and accounting periods `P1` and `P2` with the following start and end dates respectively:
   - `P1: 1/1/2010 00:00:00 +0530 and 1/15/2010 23:59:59 +0530`
   - `P2: 1/14/2010 00:00:00 +0530 and 1/30/2010 23:59:59 +0530`

   When associating the budget codes with the accounting periods (in the `BudgetAllocation.csv`), you cannot configure one budget code (say `B001`) to include both `P1` and `P2` accounting periods as there is an overlap in the dates. All dates within the same budget code must be mutually exclusive.

> **Note**
>
> Different budget codes can have overlapping accounting periods. Taking the same example, you can associate `B001` with `P1` and `B002` with `P2` though the dates overlap, because the accounting periods are linked to different budget codes.

2. Duplication of budget period: When both start and end time periods are the same for a budget.
3. Invalid combination of accounting units: Based on the master date that exists, the buying solution may invalidate some accounting combinations.
4. Duplication of budget code: When the unique budget identifier is duplicated.
5. Unspecified allocation period: When the allocation file (`BudgetAllocations.csv`) is empty.
6. Accounting validations: ERP-specific accounting validations apply to budget imports. If the accounting validations fail, the budget import will be aborted.
   - For generic and PeopleSoft ERPs, the accounting field validations are determined by the records that are imported by the `AccountingCombinationPull` integration event.
   - For SAP ERPs, the accounting field validations are determined by the records that are imported by the following integration events:
     1. `CompanyCodeWBSComboPull`
     2. `CompanyCodeInternalOrderComboPull`
     3. `GeneralLedgerPull`
     4. `AssetPull`

> **Note**
>
> If you are trying to update to an existing budget that is already in use, a validation error occurs if any on the following conditions is true:

- An accounting combination has changed
- A previously defined budget line has been removed
- A time period has changed

Ranking Budgets

The buying solutions from SAP Ariba provide an option to rank budgets that have been loaded. Ranking is a way of defining the order of precedence when logical overlaps in budget codes occur during a budget match. You rank budgets by specifying a rank value for every budget in the `BudgetLineItems.csv` file.

To resolve conflicts arising from overlapping budgets, you can rank the matching budget codes (as 1,2,3, etc., where 1 is the highest). This will help the buying solution pick the budget code that is ranked the highest.

Ranking the budgets helps the buying solution in determining the correct budget that must be associated with a requisition line item when there are multiple budgets matching that line item.

Consider the following example where you have loaded the following budget codes:

```
UTF8
```

```
     Cp1252


     Line
Number,BudgetCode,PurchaseOrg,PurchaseGroup,CompanyCode,CostCenter,GeneralLedger,WBS
Element,Asset,SubNumber,InternalOrder,Rank

1,BC0001,,,3000,0000004230,0000405000,,,,1
2,BC0002,,,3000,0000004230,0000405000,,,,2
2,BC0003,,,3000,0000004230,0000405000,,,,2
```

When a requisition with the Company Code 3000 and Cost Center 0000004230 is submitted, a budget check will validate and match this requisition with all three budget codes (mentioned above). This can be a very common occurrence in any business.

In the above example, the buying solution automatically selects the budget code `BC0001` as it is ranked higher than the other budget codes that are matching.

In scenarios where there are multiple matches (budget overlaps), the ranking option allows the SAP Ariba solution to pick the budget code that should take precedence over others.

Performing a Budget Search

From the Budget Manager workspace, you can perform a budget search by specifying the appropriate budget title, budget code, or the budget period.

Context

The search criteria for budgets appear by default. You can also choose to add or remove the search filters to enhance your budget search.

Procedure

1. In the **Budget Manager** workspace, click **Budgets**.
2. On the **Search Budgets** page, click **Add/Remove Search Filters**.
3. In the pop-up window that appears, select the search filters you require and de-select the filters that are not required.
4. Click **OK** to save the selections.
5. Specify appropriate values in the search fields and click **Search**. The search results appear. To view details of a budget, click the **Open** icon corresponding to that budget.
6. The search results provide the following details for each budget:

Data Field	Description
Title	Title of the Budget
Budget Period	Unique time period for which the budget is defined.
Rank	A number that specifies the rank of the budget code. The rank is used to determine which budget line to use when a requisition line item split matches more than one budget line. The one with the highest rank (lowest number) is used.
Budget Code	Unique identifier of a budget line item. This is a combination of all accounting information.
Time Updated	The time when the budget was last updated.
Allocated	The total amount of money that is allocated to be spent for a budget. **Note** The **Allocated** amount in the Import Excel Budget file can contain a maximum of 18 digits before the decimal mark and two digits after the decimal mark.
Tolerance	The amount that can be spent in addition to the allocated amount for a budget. For example, if the allocated amount for a budget line is $100.00, and the tolerance is defined as $10.00, the budget check will be successful until a spend of $110.00 (100+10, which is the tolerance amount). The budget check will fail if the total spend for that budget crosses this tolerance amount. **Note** The tolerance value you specify here takes precedence over the tolerance value set while configuring this feature (at the parameter level).
Reserved	The amount that is reserved from the allocated amount, against a requisition.
Ordered	The amount from a budget that has been used already and has been approved.
Available	The amount of money that is still available in a budget. **Note** The **Available** amount in the Import Excel Budget file can contain a maximum of 18 digits before the decimal mark and two digits after the decimal mark.
Adjustment	Amount adjustments that have been made to a budget.

7. To review details of the selected **Budget Code**, click on the selected Budget Code link.
8. The Review Details for Budget page appears. This is a read-only page. Click **Done** when you are finished reviewing the details.
9. Similarly, click **Budget Periods** to view details of the selected budget period.

Budget Adjustments

Adjustments to budgets are imported into the buying solution by the `Import Budget Adjustments` import task. Budget adjustments are typically performed to supplement an existing budget allocation. The adjustment amount can be positive or negative depending on how you want to supplement the budget allocation amount.

For example, if there two budgets with the following allocations:

- Budget BC001: Allocation of $5000.00
- Budget BC002: Allocation of $5000.00

You realize during the budget period that for some reason, Budget A is being overused and Budget B has allocation that is not consumed, then you can choose to transfer some of the allocation amount from Budget B to Budget A. This is termed as a budget adjustment.

In this scenario, there is a positive adjustment for Budget A and a negative adjustment for Budget B.

> **Note**
>
> When performing adjustments, you must ensure the total allocated amount for a budget does not go into the negative. The buying solution does not support negative values for allocated amounts. The smallest possible value a budget allocation can have is zero (0).

Adjustments to existing budgets are not cumulative. The purpose of an adjustment is to ensure transactions are handled appropriately in a situation the following has occurred:

- More than one budget is defined using import, for example A and B – with allocation 2000 and 3000
- Each allocation is used by various requisitions over a period of time (for example, 3 months)
- After the period of time, you realize that allocations for A and B were incorrect, and the correct allocations should be: A=5000, B=2000

To take corrective action on the allocations, you can create an adjustment on each budget as follows:

- Budget A: Create an adjustment of +3000 (5000 - 2000)
- Budget B: Create an adjustment of -1000 (2000 - 3000)

  > **Note**
  >
  > If the initial allocation is used up and you want a higher amount to be made available, then the recommended approach is to update the allocation by re-importing the budget data files with the new (increased) allocation values.

> **Note**
>
> Achieving an increase in available amount by importing a positive adjustment value is possible, but this will not be a cumulative amount. Any subsequent adjustment will replace the previously existing amount.

Related Information

Importing Budget Adjustments [page 34]

Importing Budget Adjustments

Run the `Import Budget Adjustment` task to import budget adjustment data. You can run this task from the dashboard or use the SAP Ariba data transfer tool.

Context

If you use the SAP Ariba Data Transfer Tool, run a transactional data upload operation and specify "Load Budget Adjustments" in the event option. The following instructions assume you are using the dashboard.

Procedure

1. On the dashboard, click **Manage** > **Core Administration** .
2. Click **Budget Manager** and select **Data Import/Export** in the Budget Manager workspace.
3. On the **Import** tab, click the **Import Budget Adjustments** or **Import Budget Adjustments in ZIP File Format** task.
4. In the page that displays, navigate to the `BudgetAdjustments.csv` file and click **Run**.

Sample Budget Adjustments File

The `BudgetAdjustments.csv` file defines columns for `BudgetCode`, `FiscalYear`, `AccountingPeriod`, `Adjustment`, `Currency`, `TransactionDate`, and `Comment`.

For example:

`UTF8`

```
BudgetCode,FiscalYear,AccountingPeriod,Adjustment,Currency,TransactionDate,Comment

        BC001,2009,Q109,4000,USD,3/17/2009,Moved from BC002-Q109
        BC002,2009,Q109,-4000,USD,3/17/2008,Moved to BC001-Q109
```

Budget Updates

You can perform an update to the budget allocation amount at any point of time by uploading the new budget amount (in the `BudgetAllocations.csv` file) using the budget import tasks.

Consider the following when updating budgets:

- When updating budget allocation amount, ensure the CSV files contain all the budget codes from the previous version of the budget.
- Budget code can't be updated after the budget has been consumed, that is, after a budget match has occurred on submitted requisitions.

  > **Note**
  >
  > You can update the budget code and period only if:
  >
  > - A budget code or period hasn't been previously used
  > - A new budget code or period (not defined in any previous version) is imported

> **Note**
>
> When a budget record is modified, a new version of the budget record is created and the previous version becomes inactive. From the SAP Ariba 2302 release, the **Delete inactive budget records** scheduled task will automatically delete these inactive budget records that aren't referenced by any approvable document.

Related Information

About Importing Budgets [page 15]

General Guidelines for Budget Updates

To ensure budget updating and loading match your expectations, there are general guidelines to follow.

General guidelines include:

- The budget title must specify the budget that you want to update.
- Retain all budget code definitions that were specified in the previous version of the budget, with the following exceptions:
  - An existing budget code's accounting combination can be changed or deleted if there has been no consumption against it.
  - A new budget code with a new (valid) accounting combination can be added if the budget code (or its accounting combination) does not already exist.
- Retain all budget period definitions that were specified in the previous version of the budget, with the following exceptions:

    - An existing budget code's accounting combination can be changed or deleted if there has been no consumption against it.
    - A new budget period can be added if there is no overlap with any of the existing budget periods.
- Allocation amount for an existing "code + period" combination can be changed, irrespective of the consumption on it.
- The percentage tolerance for any budget allocation can be changed, irrespective of consumption on it.
- A new "code + period" combination with an allocation amount can be specified.

Import Tasks for Closing and Reopening Budgets

You can close or reopen budgets (budget codes and budget periods) by running the following import tasks from the Budget Manager workspace.

Import Task	Description
Import Budget Code Status	Updates the budget code status.Run this task to open or close an existing budget code status. If a budget code is closed, no new requisitions can reserve funds from that budget code in any of the associated budget periods.
Import Budget Period Status	Updates the budget period status.Run this task to open or close an existing budget period status. If a budget period is closed, no new requisitions can reserve funds from that budget code associated with the closed budget period.
Import Budget Code Status in ZIP File Format	This task is similar to the Import Budget Code Status task, but uses data files in the ZIP format to update the budget code status.
Import Budget Period Status in ZIP File Format	This task is similar to the Import Budget Period Status task, but uses data files in the ZIP format to update the budget period status.

Considerations for Reopening or Closing Budgets

When you close a budget code or period, the buying solution processes requisitions with respect to the budget code.

The buying solution processes requisitions to budget code as follows:

- New requisitions are not associated with the closed budget codes.
- Existing requisitions that were associated with budget codes before they were closed, are processed as follows:
    - If there are requisitions that are approved after the budget code is closed, but already have the amount reserved from the closed budget, such requisitions are allowed to consume from the closed budget unless an approver changes the requisition to trigger the budget check again.
    - If a requisition is withdrawn after the associated budget code is closed, the reserved budget amount is released back to the closed budget.
    - If the quantity of a requisition line is reduced after the associated budget code is closed, then the entire reserved amount for that requisition line is reverted to the closed budget. The SAP Ariba solution triggers the budget check again on this requisition when it is submitted.
    - If a requisition line is associated with a closed budget and the user makes changes to that line to trigger the budget check again, then the full amount that is reserved from the closed budget is released back to that

budget. The budget check then allows only open budgets to be associated with the affected requisition line.

- If a requisition that is in the approved state is canceled (and the associated budget is closed before the cancellation), such a requisition goes to the cancellation approval flow. After the cancellation is approved, the reserved amount is released to the closed budget.

- If a budget period - budget code combination is closed, that budget is not available for selection by the Budget Manager when attempting to assign a budget.
- If a closed budget code or budget period is reopened, it goes through all existing validations for budgets.
- If multiple budget codes are associated with one budget period, and that budget period is closed, then all budget codes associated with that closed budget period are automatically closed for transactions.

Importing Budget Code and Period Status

You can import budget code and period status to close and reopen budgets.

Prerequisites

You must be a member of the **Budget Manager** group.

Procedure

1. On the dashboard, click **Manage** > **Core Administration** .
2. Click **Budget Manager** and select **Data Import/Export** in the **Budget Manager** workspace.
3. On the **Import** tab, run the following tasks:
   - **Import Budget Code Status** - to open or close budget codes
   - **Import Budget Period Status** - to open or close budget periods

> **Note**
>
> To export budget code and period status, run the **Budget Code Status** and **Export Budget Period** export tasks in the **Export** tab.

Results

When a budget code is closed, this status is reflected when you review the details of the budget.

Similarly, the budget period status is also displayed when you click on the required budget period link to review its details from the search results.

CSV Files Used by Budget Import Tasks

The budget import tasks use `BudgetCodeStatus.csv` and `BudgetPeriodStatus.csv`,

Following is an example of the `BudgetCodeStatus.csv`:

```
     "UTF-8"


     BudgetCode,Status

"BC_CODE1","Open"
"BC_CODE2","Closed"
```

Following is an example of the `BudgetPeriodStatus.csv`:

```
     "UTF-8"


     FiscalYear,Status,AccountingPeriod

"2011","Open","P1"
"2011","Open","P2"
"2011","Closed","P3"
```

> **Note**
>
> The values to be used to open and close budget lines and periods are " `Open` " and " `Closed` " (NOT " `Close` ").

Incremental Load of Budgets

You can perform the incremental load of budgets using the `Import Excel Budget` and `Import CSV Budget in 4 File Format` import tasks.

When you perform an incremental load, keep the following two points in mind:

- The new budget lines that you are loading must belong to the same budget title (header value in the `BudgetHeader.csv`)
- The accounting field values you provide (in `BudgetLines.csv`) for the new lines must be unique.

The incremental load feature can also be used to update or change an existing budget line.

For example, if you have the following budget data:

- Title: `Budget-4`
- Budget Codes: `BC1 and BC2`

Using the incremental load feature, you can do the following:

- Import new budget codes - `BC3` and `BC4`
- Update the existing budget code (for example, change the budget allocation for `BC1`)

> **Note**
>
> If a budget code has already been consumed, you can only make changes to the allocation amounts. No other changes are allowed.

If there are budgets that are not yet consumed, you can make any update to such lines (including changing the budget period, accounting, and allocation).

In the example, you import the following CSV files for an incremental load of the new budget lines.

Sample Files for Incremental Load of Budgets for a Simple Generic ERP

Incremental loading of budgets makes uses of four CSV files: `BudgetHeader.csv`, `BudgetLines.csv`, `BudgetPeriods.csv`, and `BudgetAllocations.csv`.

`BudgetHeader.csv` :

```
     UTF8

     Cp1252

     Title
Budget-4
```

`BudgetLines.csv` :

```
     UTF8

     Cp1252


LineNumber,BudgetCode,Company,BusinessUnit,CostCenter,Project,Region,SubAccount,Rank
,Budget Check Disabled

1,BC1,04,US015,6400,,,,1,False
3,BC3,04,US015,6500,,,,3,False
4,BC4,00,US012,1300,,,,4,False
```

`BudgetPeriods.csv` :

```
     UTF8

     Cp1252

     FiscalYear,AccountingPeriod,StartDate,EndDate
2011,Q111,1/1/2011 00:00:00 +0530,3/31/2011 23:59:59 +0530
```

```
2011,Q211,4/1/2011 00:00:00 +0530,6/30/2011 23:59:59 +0530
2011,Q311,7/1/2011 00:00:00 +0530,9/30/2011 23:59:59 +0530
2011,Q411,10/1/2011 00:00:00,12/31/2011 23:59:59
```

> **Note**
>
> The date format to be followed is `MM/DD/YYYY HH:mm:ss zzz`, where `zzz` is the GMT offset.

`BudgetAllocations.csv` :

```
     UTF8


     Cp1252



LineNumber,BudgetCode,FiscalYear,AccountingPeriod,AllocationAmount,Currency,Percenta
geTolerance,AvailableAmount,AvailableCurrency

1,BC1,2011,Q111,
     5510
    ,USD,10,,
1, BC1,2011,Q211,
     6020
    ,USD,10,,
1, BC1,2011,Q311,5030,USD,10,,
1, BC1,2011,Q411,5040,USD,10,,
3
     , BC3,2011,Q111,5000,USD,10,,


     3, BC3,2011,Q211,6000,USD,10,,


     3, BC3,2011,Q311,6500,USD,10,,


     3, BC3,2011,Q411,7500,USD,10,,


     4, BC4,2011,Q111,4000,USD,10,,


     4, BC4,2011,Q211,6000,USD,10,,


     4, BC4,2011,Q311,7030,USD,10,,


     4, BC4,2011,Q411,8040,USD,10,,
```

> **Note**
>
> In this example, note that the data for budget code `BC2` has not been included in the files for import.

Budget Export

You can export budget information by using either Ariba Administrator or the SAP Ariba data transfer tool.

- In the **Budget Manager** workspace click **Data Import/Export**, click the **Export** tab, and run either of the following data export tasks:
  - `Export CSV Budget` - export the budget in a CSV file format (`BudgetCSVExport.csv`).
  - `Export Excel Budget` - export the budget in an Excel spreadsheet format (`BudgetExcelExport.xls`).

The budget export event provides information on the status of all budgets that have been imported, along with the budget line details, associated periods, allocation, reserved and ordered values for each budget line.

The export event also serves as a means for users to review budgets available in the buying solution.

> **Note**
>
> Users on SAP Ariba Buying and Invoicing and SAP Ariba Buying, Release 10s1 SP6 and higher can now export budgets with inactive accounting fields.

Budget Transactions Export

You can export budget transaction information by using either Ariba Administrator or the SAP Ariba data transfer tool.

- In the **Budget Manager** workspace click **Data Import/Export**, click the **Export** tab, and run the `Export Budget Transactions` export task to export the budget in CSV format.
- In the SAP Ariba data transfer tool, specify "Export Budget Transactions" in the `event` option.

Sample BudgetTransactionExport.csv File

The sample `BudgetTransactionExport.csv` file illustrates the column header definitions and an example of a single export record.

```
UTF-8

     Fiscal Year,Accounting Period,Start Date,End
Date,BudgetCode_Code,Accounting_ProcurementUnit,Accounting_Account,Accounting_Compan
y,Accounting_BusinessUnit,Accounting_CostCenter,Accounting_SubAccount,Accounting_Pro
ject,Accounting_Region,Accounting_Product,Currency,Allocated Amount,Transaction
Date,Transaction Accounting Date,TransactionType,Action,Document
Number,Requester,Line Number,LineText,Split Number,Line
Accounting_ProcurementUnit,Line Accounting_Account,Line Accounting_Company,Line
Accounting_BusinessUnit,Line Accounting_CostCenter,Line Accounting_SubAccount,Line
Accounting_Project,Line Accounting_Region,Line Accounting_Product,Originating
System,Originating System Reference ID,Transaction
Amount,TransactionAmount.Currency.UniqueName,Funds Available,Funds Available
Currency,Comment
```

```
2010,D1,05/01/2010 00:00:00 +0530,05/31/2010 23:59:59
+0530,BC1,,,,,,1210,,,,USD,"6,150",05/24/2010 18:55:57 +0530,05/24/2010 18:55:50
+0530,Requisition Approved,Req Fully Approved,PR37,Arnold Davis,2,Mac Monitor DB15M/
M,1,US005,7752,GD,US005,3300,1210,,,,,,11.95,USD,"6,033.15",USD,
```

This export event provides information on the budget usage in all processed transactions, including accounting period start and end dates, budget codes, budget and line item accounting data (such as company, business unit, cost center, project, etc.), transaction amount and currency, available budget amount and currency.

Budget Checks in External Systems

You can configure real-time budget checks to validate requisitions against budget information hosted in external ERP systems.

This feature enables users to perform the following actions in the procurement workflow:

- Optionally, initiate budget checks in the external ERP system to check for availability of funds.
- Reserve funds for submitted requisitions after budget checks in the ERP system are successful.
- Encumber funds for orders created after requisitions are fully approved.
- Withdraw reserved funds to release them back into availability for withdrawn or canceled transactions.

> **Note**
>
> Contract requests and invoices do not undergo budget checks in external ERP systems.

For integration of budget-related transactions on PeopleSoft or Generic ERP systems, the buying solution publishes a SOAP message. You can use an Enterprise Application Integration (EAI) tool to consume the SOAP message and enable transactions with the ERP system using web services.

For buyer organizations that use SAP as their external ERP system, this feature provides an end-to-end integration with the Materials Management module in SAP ERP. This setup supports real-time monitoring of budget information that is defined for Internal Orders (IO) and Work Breakdown Structure (WBS) elements. Budget checks in the SAP ERP also include document checking to verify the accounting information that users enter on approvable documents.

Enabling Budget Checks in External Systems [page 43]

Approvables Migrated from Locally-Integrated Budgets to Budgets in an External System [page 44]

Budget Checks on Aggregated Requisitions [page 45]

Configuring Web Services Integration Tasks for Budget Checks in External Systems [page 46]

Enabling Budget Checks in External Systems

In the default configuration, budget checking in external systems is disabled.

To enable budget checking in an external ERP system, please have your Designated Support Contact log a service request and an SAP Ariba Customer Support representative will follow up to complete the request. Additionally, provide the SAP Ariba Customer Support representative with the budget activation date. When the activation date is specified and the budget check functionality is enabled, it means that every requisition that is submitted after this activation date, is subjected to a budget validation in the external ERP system.

If your site is integrated with SAP ERP, you can also choose to enable the Funds Management accounting functionality in addition to the budget checking functionality.

> **Note**
>
> For sites that are enabled for both budget checking within the buying solution and budget checking in external systems, the ability to perform budget checks within the buying solution is automatically disabled starting from the date that is configured to enable budget checking in external systems.

Related Information

Funds Management Accounting [page 47]
Purchasing Unit Administration Guide
Funds Management Accounting Functionality

Approvables Migrated from Locally-Integrated Budgets to Budgets in an External System

When you migrate from budget checking within the buying solution to budget checking in an external ERP system, the locally-integrated budget check system is automatically disabled starting from the date that is configured to enable budget checks in the ERP system.

After migrating to the external budget checking system, when a user submits a requisition for approvals, the ERP system validates the requisition for availability of funds. After the validation checks are successful, the ERP system assigns an **ERP Reference Number** to the requisition. Additionally, for an SAP ERP, a copy of the requisition is created in the ERP system.

The following are the scenarios related to managing approvables after users migrate to the real-time budget check system from a locally-integrated budget check system:

A requisition in Composing state is submitted for approvals	A copy of the requisition is saved in the ERP system after funds are reserved in the ERP system and an **ERP Reference Number** is assigned to the requisition.
A requisition is in Submitted state and not fully approved	The requisition does not have an **ERP Reference Number** in the external ERP system and therefore, no funds are reserved in the ERP system even though it has funds allocated in the locally-integrated budget. The requisition is not sent to the ERP system for budget checks if a user edits the requisition that results in an increase or decrease of funds. The requisition is sent to the ERP system for reservation of funds only after it is fully approved. A copy of the requisition is saved in the ERP system after funds are successfully reserved in the ERP system and an **ERP Reference Number** is assigned to the requisition.
A purchase order has funds reserved in the locally-integrated budget	The associated requisition for the purchase order does not have an **ERP Reference Number** in the ERP system and therefore, no funds are reserved in the ERP system even though it has funds allocated in the locally-integrated budget check system.

Additionally, if the requisition is changed resulting in an increase or decrease of funds, the budget allocation is not validated in the external ERP system when the requisition is submitted for approvals after the changes. The requisition is sent to the ERP system for reservation of funds only after it is fully approved. A copy of the requisition is saved in the ERP system after funds are successfully reserved in the ERP system and an **ERP Reference Number** is assigned to the requisition.

Budget Checks on Aggregated Requisitions

Based on demand holding rules, line items on a requisition that are eligible to be consolidated as held line items in aggregated requisitions are not subjected to budget checks in the ERP system when a user initiates budget checks or submits the requisition for approvals.

Held line items on an aggregated requisition undergo budget checks in the ERP system only after a purchasing agent or the buying solution submits the aggregated requisition for approvals. The aggregated requisition undergoes additional budget checks in the ERP system if an approver makes changes before approving the requisition.

For an SAP ERP-integrated site, if all items on a requisition qualify for demand holding rules, the SAP ERP performs the following actions:

- For user-initiated budget checks (**Check** button action), the ERP system does not validate the requisition for budget information and returns a success message.
- For a requisition submitted for approval, the ERP system neither validates the requisition for budget information nor reserves funds for the requisition. However, the buying solution assigns an internal **ERP Reference Number** to the requisition to prevent the requisition from moving back to the **Composing** state. The internal **ERP Reference Number** is prefixed with the letter "D", for example D000000001. While finding requisitions in the ERP system using the **ERP Reference Number** as a search criteria, requisitions with internal reference numbers are not listed in the search results.

> **Note**
>
> For PeopleSoft and Generic ERP systems, we recommend that you configure similar rules to validate budget information for items on which demand holding rules apply.

Related Information

Rules for Demand Aggregation [page 113]
Prerequisites for Using Demand Aggregation
Enabling Budget Checks in External Systems [page 43]

Configuring Web Services Integration Tasks for Budget Checks in External Systems

To integrate with external systems, configure the end points for web services and then enable the web services.

Prerequisites

- Your site must be configured to support ERP Order or ERPCC as the ordering method.
- In multi-ERPI configurations, ensure that the budget check feature is enabled on all child sites.

Procedure

1. For SAP, PeopleSoft, and Generic ERP systems, build the integration using an Enterprise Application Integration (EAI) tool to support web services-based integration of transactional data between the buying solution and the ERP system.

   > **Note**
   >
   > For SAP NetWeaver PI, the buying solution provides an provide an end-to-end integration with SAP ERP using SAP NetWeaver PI.

2. Configure the end points for the following web services:
   - **Export Requisitions for User-Initiated External Budget Check**
   - **Export Submitted Requisitions for External Budget Check**
   - **Export Requisitions to External System to Revert Funds**
3. Enable the web services.

Additional references

- Multi-ERP Configuration Guide for information about multi-ERP functionality.
- Administration and Data Maintenance Fundamentals for more information about integrating data using web services and configuring an end point.

Funds Management Accounting

The Funds Management accounting functionality enables you to include accounting information for Funds Management account assignment fields on approvable documents, such as requisitions and invoice reconciliation documents for encumbrance accounting and budget availability control on fund-based budgets.

This functionality is supported only on buying solutions integrated with SAP ERP.

When the Funds Management accounting functionality is enabled, you can perform the following tasks:

- Include Funds Management account assignment fields as standard Flexible Master Data (FMD) fields on approvable documents based on pre-configured FMD templates.
- Derive accounting values from the SAP ERP for Funds Management account assignment fields on requisitions based on other accounting information that is included in such documents.

When you perform procurement transactions, Funds Management processes run in the SAP ERP using the data sent by the buying solution to validate the Funds Management accounting information on the approvable documents.

Enabling the Funds Management accounting feature

The Funds Management accounting feature is not enabled on your site in the default configuration.

To enable this feature, have your Designated Support Contact log a service request, and an SAP Ariba Customer Support representative will contact you. Additionally, include a request for the following:

- Add visibility conditions for specific Funds Management account assignment fields that you do not want to include for your site.
- Add visibility conditions for specific approvable documents for which you do not want to include Funds Management account assignment fields.
- Configure control attributes, such as mandatory, optional, or read-only, for specific Funds Management account assignment fields.

> **Note**
>
> If visibility conditions are not configured, there is no action possible to remove the Funds Management account assignment fields that are already added to any approvable document.

Prerequisites for the Funds Management accounting feature

- Your site must be configured to support ERP Order or ERPCC as the ordering method.
- In multi-ERP configurations, enable the required functionality in all child sites.
- Funds Management accounting information in your SAP Ariba solution must be synchronized with the master data for the Funds Management application in SAP ERP.

> **Note**
>
> There is no synchronization with SAP ERP to support organic data growth for the Funds Management account assignment fields. As a result, Funds Management accounting master data in your SAP Ariba solution is not automatically updated if there are changes to the master data in SAP ERP. In addition, your SAP Ariba solution does not maintain or validate the relationship between Funds Management account assignment fields.

> **Note**
>
> To disable the Funds Management accounting functionality, contact SAP Ariba Support.

Configuring Funds Management Accounting [page 48]

Configuring Funds Management Accounting

To configure Funds Management accounting functionality, you must run the related data import tasks and configure the end points for the web services used by this feature.

Procedure

1. Run the data import tasks to import Funds Management accounting information for your site. For more information, see Importing Flexible Master Data Templates for Funds Management Account Assignment Fields [page 49].
2. Configure and enable the end points for the communication channel used by the web services. For more information, see Enabling Web Services Integration Tasks for Funds Management Account Assignment Fields [page 48].

Enabling Web Services Integration Tasks for Funds Management Account Assignment Fields

You must configure the end points for the web services used by the Funds Management accounting functionality.

Context

For more information about integrating data using web services and configuring an end point, see Web Services Overview.

Procedure

1. Build the integration using an Enterprise Application Integration (EAI) tool to support web services-based integration of transactional data between the buying solution and the ERP system.

   > **Note**
   >
   > For SAP NetWeaver PI, the buying solution provides an end-to-end integration with SAP ERP using SAP NetWeaver PI.

2. Depending on your SAP Ariba solution, configure the end points for the following web services:
   - **Export Requisitions to Derive the Accounting Information for Funds Management**
   - **Export Invoice Reconciliation Documents to External System to Add Funds Management Accounting Information**
3. Enable the web services.

Importing Flexible Master Data Templates for Funds Management Account Assignment Fields

SAP Ariba provides pre-configured FMD templates to configure the Funds Management account assignment fields for use with SAP ERP-integrated sites. Depending on the requirements of your site, import the appropriate FMD templates to your buying solution.

Procedure

1. Go to **Manage** > **Core Administration** > **Site Manager** > **Data Import/Export**.
2. On the **Import** tab, perform a search for the Funds Management accounting data import tasks.

   You can use **Flex** as a keyword to perform a search for Funds Management accounting data import tasks.
3. Click **Import** for the appropriate task.
4. Click **Browse** and select the appropriate CSV file.
5. Click **Run** to enable the task.

Related Information

Sample CSV File Formats for FMD Templates [page 50]

Sample CSV File Formats for FMD Templates

The sample formats of the CSV files illustrate the column header definitions for the data import tasks used by the Funds Management feature.

The following table lists the simplified data import tasks and the corresponding CSV files for the Funds Management account assignment fields:

Data Import task	Name of CSV file
Import Flex Master Data EarmarkedFundsDocument	EarMarkExport.csv
Import Flex Master Data EarmarkedFundsLineItem	EarmarkedFundsDocumentExport.csv
Import Flex Master Data Fund	FundExport.csv
Import Flex Master Data BudgetPeriod	BudgetPeriodExport.csv
Import Flex Master Data FundsCenter	FundsCenterExport.csv
Import Flex Master Data CommitmentItem	CommitExport.csv
Import Flex Master Data FunctionalArea	FuncAreaExport.csv
Import Flex Master Data Grant	GranteeExport.csv
Import Flex Master Data FMArea	FMAreaExport.csv

Import Flex Master Data EarmarkedFundsDocument

The **Import Flex Master Data EarmarkedFundsDocument** data import task defines the **Earmarked Funds Document** field and related information and reads from the `EarMarkExport.csv` file.

Following is an example of the `EarMarkExport.csv` file:

Sample Code

```
UTF-8,,,,,,,
UniqueName,Name,CompanyCode,PurchasingUnit,cus_FMArea,cus_FMValidfromdate,cus_FMV
alidtodate,cus_Description
10,Item Nozzle,,,Atlanta,1/1/2015,1/1/2016,Nozzle for Rocket IVX
11,Item Oxidizer,,,Toronto,1/1/2015,1/1/2016,Oxidizer for Rocket IVX
12,Propulsion Pumps,,,columbia,1/1/2015,1/1/2016,Pumps for Rocket IVX
```

Import Flex Master Data EarmarkedFundsLineItem

The **Import Flex Master Data EarmarkedFundsLineItem** data import task defines the **Earmarked Funds Line Item #** field and related information and reads from the `EarmarkedFundsDocumentExport.csv` file.

Following is an example of the `EarmarkedFundsDocumentExport.csv` file:

Sample Code

```
UTF-8
UniqueName,Name,CompanyCode,PurchasingUnit,cus_DocumentItem,cus_ItemText,Amount,U
niqueName,ApproxAmountInBaseCurrency,ConversionDate,UniqueName
10,Roket Nozzle,,,10,Fund For Nozzle,2000,,,,
11,Oxidizer,,,11,Fund for Oxidizer,2001,,,,
12,Propulsion pumps,,,12,Fund for propulsion Systems,2002,,,,
```

Import Flex Master Data Fund

The **Import Flex Master Data Fund** data import task defines the **Fund** field and related information and reads from the `FundExport.csv` file.

Following is an example of the `FundExport.csv` file:

Sample Code

```
"UTF-8"
UniqueName,Name,CompanyCode,PurchasingUnit,cus_FundAREA,cus_ValidStartDate,cus_Va
lidEndDate,cus_Description
10,Space Discovery,,,Atlanta,01/01/2014,01/01/2016,Fund for Space Discovery
11,Space Station,,,Toronto,01/01/2015,01/01/2017,Fund for Space Station
12,Mars project,,,Columbia,01/01/2013,01/01/2018,Fund for Mars project
```

Import Flex Master Data BudgetPeriod

The **Import Flex Master Data BudgetPeriod** data import task defines the **Budget Period** field and related information and reads from the `BudgetPeriodExport.csv` file.

Following is an example of the `BudgetPeriodExport.csv` file:

Sample Code

```
"UTF-8"
UniqueName,Name,CompanyCode,PurchasingUnit,cus_BudgetPeriod,cus_FMValidfromdate,c
us_FMValidtodate,cus_Description
10,FiscalYear2015-2016,,,FIY15,01/01/2015,01/01/2016,FiscalYear2015-2016
11,FiscalYear2016-2017,,,FIY16,01/01/2016,01/01/2017,FiscalYear2016-2017
```

Import Flex Master Data FundsCenter

The **Import Flex Master Data FundsCenter** data import task defines the **Funds Center** field and related information and reads from the `FundsCenterExport.csv` file.

Following is an example of the `FundsCenterExport.csv` file:

Sample Code

```
"UTF-8"
UniqueName,Name,CompanyCode,PurchasingUnit,cus_FMArea,cus_FundsCenter,cus_FMValid
fromdate,cus_FMValidtodate,cus_Description
10,R&D,,,Atlanta,R&D,01/01/2015,01/01/2016,Research and Development
11,Space Exploration,,,Toronto,Space Exploration,01/01/2015,01/01/2016,Space
Exploration
12,Propulsion Systems,,,Columbia,Propulsion
Systems,01/01/2015,01/01/2016,Propulsion Systems
```

Import Flex Master Data CommitmentItem

The **Import Flex Master Data CommitmentItem** data import task defines the **Commitment** field and related information and reads from the `CommitExport.csv` file.

Following is an example of the `CommitExport.csv` file:

Sample Code

```
"UTF-8"
UniqueName,Name,CompanyCode,PurchasingUnit,cus_FMArea,cus_CommitmentItem,cus_FMVa
lidfromdate,cus_FMValidtodate,cus_SuperiorCommitmentItem,cus_Descriptionline1,cus
_Descriptionline2,cus_Descriptionline3
10,R&D,,,Atlanta,R&D,01/01/2015,01/01/2016,ReasearchOnMars,Research&Development,M
ars Project,Ongoing
11,SpaceShuttle,,,Toronto,SpaceShuttle,01/01/2015,01/01/2016,SpaceShuttle,SpaceSh
uttle,Shuttle Project,Final Stage
```

Import Flex Master Data FunctionalArea

The **Import Flex Master Data FunctionalArea** data import task defines the **Functional Area** field and related information and reads from the `FuncAreaExport.csv` file.

Following is an example of the `FuncAreaExport.csv` file:

Sample Code

```
"UTF-8"
UniqueName,Name,CompanyCode,PurchasingUnit,cus_FMValidfromdate,cus_FMValidtodate,
cus_Functionalareatext
10,Engineering,,,01/01/2015,01/01/2016,Engineering
11,Applied Science,,,01/01/2015,01/01/2016,Applied Science
12,Opearations,,,01/01/2015,01/01/2016,Opearations
```

Import Flex Master Data Grant

The **Import Flex Master Data Grant** data import task defines the **Grant** field and related information and reads from the `GranteeExport.csv` file.

Following is an example of the `GranteeExport.csv` file:

Sample Code

```
"UTF-8"
UniqueName,Name,CompanyCode,PurchasingUnit,cus_Sponsor,cus_Validfromdate,cus_Vali
dtodate,cus_GrantName,cus_GrantDescription,cus_GrantType,cus_GrantTypeDescription
10,COFAR,,,Financial Reform Dept,01/01/2015,01/01/2016,COFAR,Council on
Financial Assistance Reform,Finance,Nasa's Exisiting Fund
11,PMS,,,Payment Managemt System,01/01/2015,01/01/2016,PMS,Payment Management
System,Payment,USA Govt Allocation
```

Import Flex Master Data FMArea

The **Import Flex Master Data FMArea** data import task defines the **FM Area** field and related information and reads from the `FMAreaExport.csv` file.

Following is an example of the `FMAreaExport.csv` file:

Sample Code

```
"UTF-8"
UniqueName,Name,CompanyCode,PurchasingUnit,cus_FMAreatext,cus_FMAreaCurrency
10,Atlanta,,,Atlanta,USD
11,Toronto,,,Toronto,USD
12,Columbia,,,Columbia,USD
```

Branding and Conditions

Terms and conditions describe all of the provisions, agreements, obligations, and rights pertaining to a purchase order.

When you define terms and conditions for purchase orders:

- Purchase orders displayed in the user interface include a link to terms and conditions text.
- Terms and conditions are included in purchase orders that are printed or sent to SAP Business Network.

Defining terms and conditions text is optional.

> ### Note
>
> The data that you add as terms and conditions must be in plain text, including the line breaks.

Import Purchase Order Conditions Task [page 54]

Configuration of Purchasing Unit-Specific Terms and Conditions for Purchase Order Print [page 55]

Customizing Purchasing Unit-Based Branding of Purchase Order Prints [page 56]

Customization of Purchase Order Print Header [page 57]

Import Purchase Order Conditions Task

The **Import Purchase Order Conditions** data import task defines terms and conditions text for purchase orders. It reads data from the CSV file named `POConditions.csv`.

Following is an example of `POConditions.csv`:

```
UTF8
Title,Text
Terms and Conditions of Purchase,"All purchases made by SAP Ariba are subject
to SAP Ariba's Terms and Conditions of Purchase in effect at the date this
purchase order ('Order') is issued to you, which can be found at the following
uniform resource locator: http://www.ariba.com/online_PO_TOU.cfm. To request a hard
copy of these Terms and Conditions of Purchase, please contact the SAP Ariba
Purchasing Department. SAP Ariba explicitly disclaims and rejects all other terms
and conditions of any other party associated with SAP Ariba's purchases."
```

> ### Note
>
> When you import new terms and conditions, existing purchase orders still refer to the previous terms and conditions.

> ### Note
>
> The terms and conditions that you define here is information that is common across all locations for your company. If you want to have multiple and different terms and conditions specific to different locations, it is

recommended that you define these terms and conditions separately on your company's official website and add the appropriate URLs as references in the terms and `POconditions .csv` file.

Configuration of Purchasing Unit-Specific Terms and Conditions for Purchase Order Print

If you use purchasing units at your site, you can configure the terms and conditions that appear on the printed purchase orders to be specific to the purchasing units that the POs are associated.

To configure terms and conditions that are specific to purchasing units, you must specify valid values in the `PurchasingUnit` field in the `POConditions.csv` file and run the Import Purchase Order Conditions data import task.

Besides the Terms and Conditions text, you can also choose to configure the following information that is specific to purchasing units:

- Legal Conditions
- Invoice Conditions

To do this, specify appropriate information in the LegalText and InvoicingText fields that correspond to the required purchasing units in the `POConditions.csv` file.

For example:

```
UTF8
Version,Text,Title,InvoicingText,LegalText,ProcurementUnit
1,"TnC_text_here_SG","SG_Conditions","inv_test_hereSG","legal_text_hereSG","SNG02"
```

In this example, after the import task runs, if a user prints a purchase order that is associated with the purchasing unit SNG02, that printed purchase order will display the terms and conditions, legal conditions, and invoice conditions text that you have specified for SNG02.

> **Note**
>
> You can configure the buying solution to display common terms and conditions for all or some purchasing units in your company. To do this, add a row in the `POConditions.csv` file that specifies the default legal text with the purchasing unit value set as `ALL`. If the purchasing unit in the purchase order is not found in the `POConditions.csv` file, the default terms and conditions are displayed for the purchase order.

When new terms are added, the buying solution versions these terms, to be used when required. Adding new terms does not change the terms that are already associated with purchase orders.

Related Information

Purchasing Unit Administration Guide

Customizing Purchasing Unit-Based Branding of Purchase Order Prints

You can choose to customize the purchase order print layout to display logos that are specific to the purchasing units to which the purchase orders are associated.

Prerequisites

If the logo is specified, the purchase order header is displayed below the logo and above the purchase order name.

For information about purchasing units, see Purchasing Unit Administration Guide.

You specify the name of the logo file in the `LogoFileName` field in `ProcurementUnit.csv` file. You must specify the exact file name with the file extension in lower case (for example, `foo.gif` or `foo.jpeg`), but without the location (path) details. This field is case sensitive.

For example:

```
UTF-8
Name,Description,Level,Parent,UniqueName,LogoFileName
"US005_Name","US005 FLORIDA OPERATIONS",,"All","US005",FL_OP.gif
```

> **Note**
>
> This `LogoFileName` value must match exactly with the logo file name that you have supplied to SAP Ariba Customer Support.

Procedure

1. Supply SAP Ariba Customer Support with the appropriate images for the corresponding purchasing units.

   > **Note**
   >
   > The images/logos that you send must be in JPEG or GIF formats only.

2. After SAP Ariba Customer Support publishes the required brands, update the `ProcurementUnit.csv` and `POConditions.csv` with valid values, and run the import tasks (**Import Purchasing Units** and **Import Purchase Order Conditions** respectively).

Customization of Purchase Order Print Header

You can choose to display a customized header on your purchase order print header. This customized header is independent of the purchasing unit logo that you may (or may not) have configured.

To customize the purchase order print header, contact your account representative and reach out to the SAP Services team. Provide them with the information that you want displayed at the header-level of a printed order. For more information, visit the SAP Store.

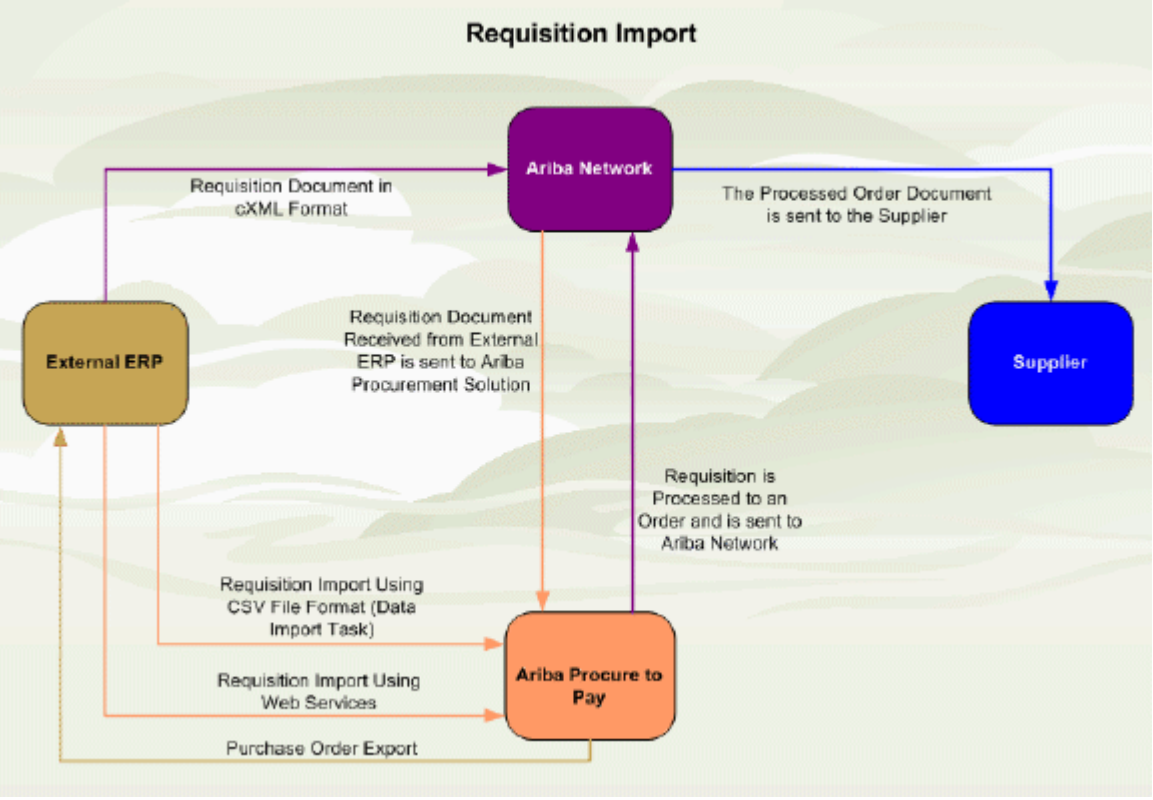
Import of Requisitions

You can import requisitions for material and service items from an external ERP system by using either the **Data Import** task (from Core Administration) or the SAP Ariba data transfer tool. Imported requisitions can include individual line items and kit line items.

For material line items, you send requisition documents in cXML format to the buying solution through SAP Business Network. After the requisition is processed into an order, the purchase order is sent to the appropriate supplier through SAP Business Network.

> **Note**
>
> You can't import new requisitions that have partial catalog items containing parametric data. However, you can update the existing requisitions that have partial catalog items containing parametric data.

The following diagram illustrates the requisition workflow.



Requisition import allows you to:

- Create new requisitions
- Update previously imported requisitions

> **Note**
>
> You cannot perform an update operation on requisitions that have a status of Ordering, Receiving, or Received.

- Cancel/delete previously imported requisitions.
- Facilitates external requisition line items (customer catalog and non-catalog) to be associated with appropriate suppliers, ensuring the best possible price match.

> **Note**
>
> To enable update and cancel/delete operations during requisition import, you must ensure that the CSV files (or XML files, when using the web services channel) are updated to include the required columns.

You can import requisitions with Buyer Part Numbers and Facility values, which can be internally mapped to the appropriate Supplier and Supplier Part Number values, thus ensuring the best price match. The internal mapping is configured using the item master file.

> **Note**
>
> The number of material line items that can be included in requisitions during import is limited to 500. You can't import requisitions that contain more than 500 line items.

Item Master File [page 60]

Item Master Mappings [page 60]

Item Master [page 63]

Use customer catalog prices in SAP Ariba solutions to replace user-defined prices of customer catalog line items in imported requisitions [page 67]

Import of Requisitions with Service Items [page 67]

Import of Requisitions Using Various Integration Channels [page 68]

Import of Requisitions with Kits [page 92]

Import of Requisitions with User Set Fields [page 95]

Import of Requisitions with Account Type Data [page 97]

Import of Contracts with Requisitions [page 97]

Import of Requisitions with Subcontractor Information for Suppliers [page 97]

ERS Flag Value in Imported Requisitions [page 98]

Considerations for Importing Requisitions [page 99]

Import of Amendments [page 103]

APIs to Access Submit Failure Records for Imported Requisitions [page 110]

Related Information

About the Item Master File [page 60]

Item Master File

The Item Master is a CSV file residing in your buying solution that provides a comprehensive mapping of all available items in an external ERP system with appropriate suppliers and supplier part numbers that have been loaded into the buying solution.

You either upload the item master file using data import task, or you create it from the user interface.

The item master file provides you with an option to cross reference externally existing items that are termed as non-catalog in the buying solution.

The item master file includes the following mappings, among others, for each item:

- Buyer Part Number
- Facility
- Supplier
- Supplier Part Number
- Supplier Part Auxiliary ID
- Price
- Rank

> **Note**
>
> Only users belonging to the **Item Master Manager** group can create or load the item master.
>
> If your SAP Ariba system is configured for multi-ERP (a parent site with child sites configured for integration with multiple external ERP systems), the item master must be created or loaded only in the parent site and not the child sites.

Item Master Mappings

The buying solution performs checks against the item master to identify the best possible match for the imported requisition before creating the actual requisition and processing it.

The following checks are performed in order:

1. Identify the Buyer Part Number (BPN) and Facility values in the imported requisition and list the matching entries in the item master. If enabled for the buyer site, the list of matching entries excludes expired customer catalog items.
2. From this list, the system next looks for the supplier match. That is, the buying solution identifies the entries in the item master that match with the supplier value that has been specified in the imported requisition.
3. If there are multiple matches, the buying solution looks into the entry that has the highest rank (lowest integer).
4. If there are multiple entries found with the same rank, the buying solution looks for matching entries with a contract price specified for the item. Among all the matching entries with the contract price, the system selects the entry with the lowest price and applies it to the requisition.

5. If the imported requisition does not include the BPN and Facility field values, but has valid Supplier and Supplier Part Number values that match a customer catalog item in the buying solution, that item is still created as a customer catalog item, and the best matching customer catalog price is selected.
6. If the imported requisition does not include the BPN and Facility field values, and the Supplier and Supplier Part Number values do not match any customer catalog items, the requisition line is created as a non-catalog item in the buying solution.

> **Note**
>
> If an imported requisition has the BPN and Facility values specified, the price specified in the imported requisition is not taken into consideration, even if that price is lower than the best matched customer catalog item price. In such instances, the buying solution always applies the best price that is matched from the item master entry.

Consider the following example to understand how the buying solution processes an imported requisition:

Step 1: The user creates or loads an item master with the following mappings and cross references:

```
     "UTF-8"


UnitOfMeasure,SupplierPartNumber,CommodityCodeValue,Price,CommodityCodeDomain,Rank,C
urrency,Facility,BuyerPartNumber,SupplierPartAuxiliaryID,CommonSupplierID,Descriptio
n

"EA","VGASP2E","432118","99.99","unspsc","1","USD","Atlanta","VGA
Cables",,"sid496","1 to 2 VGA Splitter/Booster Economy HD15F x 3"
"EA","VGASP4","432118","109.99","unspsc","1","USD","Atlanta","VGA
Cables",,"sid496","1 to 4 VGA Splitter/Booster HD15F x 5"
"EA","VGASP2","432118","79.99","unspsc","2","USD","Atlanta","VGA
Cables",,"sid496","1 to 2 VGA Splitter/Booster HD15F x 3"
"EA","VGA","2612","120","unspsc","2","USD","Atlanta","VGA
Cables","VGA_1234","sid515","VGA Cables for Splitter. Non Catalog Item"
```

Step 2: A requisition is imported with the following details:

```


Requisition_Number,Line_Number,Supplier,Supplier_Location,Contact_Id,DeliverTo,Need_
By_Date,CommonCommodity_Code,Domain,Commodity_Code,Quantity,Unit_Price,Unit_Of_Measu
re,Currency,Item_Description,ManPartNo,ItemPartNo,SplitAccountingType,Region,BuyerPa
rtNumber,Facility

"5302",1,11,14,,"Sample
DeliverTo","02/10/2012","51",unspsc,"","500","200","EA","USD","VGA
Cables",SampleManPartNo,SampleItemPartNo,,SE10,"
     VGA Cables
    ","
     Atlanta
    "
```

Step 3: BPN and Facility-based Match. The buying solution looks up the item master to find an appropriate match for this item.

Because the BPN and Facility values are specified in the imported file, the first rule of matching is applied. The buying solution identifies the following entries from the item master that match the BPN and Facility values:

```
          EA","VGASP2E","432118","99.99","unspsc","1","USD","
```

```
    Atlanta
        ","
    VGA Cables
        ",,"sid496","1 to 2 VGA Splitter/Booster Economy HD15F x 3"
```

- `"EA","VGASP4","432118","109.99","unspsc","1","USD","Atlanta","VGA Cables",,"sid496","1 to 4 VGA Splitter/Booster HD15F x 5"`
- `"EA","VGASP2","432118","79.99","unspsc","2","USD","Atlanta","VGA Cables",,"sid496","1 to 2 VGA Splitter/Booster HD15F x 3"`
- `"EA","VGA","2612","120","unspsc","2","USD","Atlanta","VGA Cables","VGA_1234","sid515","VGA Cables for Splitter. Non Catalog Item"`

> **Note**
>
> If enabled for the buyer site, the list of matching entries excludes expired customer catalog items.

Step 4: Supplier-based match. Because multiple matches are found, the buying solution attempts to select the best entry from the item master based on the Supplier value.

The `CommonSupplierID` field in the item master is internally mapped to a unique supplier name and supplier location. In this example, the `CommonSupplierID` - `sid496` is internally mapped to supplier `11` .

Based on this mapping, the following entries are selected:

- `EA","VGASP2E","432118","99.99","unspsc","1","USD","Atlanta","VGA Cables",,"sid496","1 to 2 VGA Splitter/Booster Economy HD15F x 3"`
- `"EA","VGASP4","432118","109.99","unspsc","1","USD","Atlanta","VGA Cables",,"sid496","1 to 4 VGA Splitter/Booster HD15F x 5"`
- `"EA","VGASP2","432118","79.99","unspsc","2","USD","Atlanta","VGA Cables",,"sid496","1 to 2 VGA Splitter/Booster HD15F x 3"`

Step 5: Rank-based match. Because of the multiple matches found from the previous step, the buying solution determines the rank to further identify the best possible match from the listed item master entries. Here, it attempts to match an appropriate entry based on the rank value that has been specified. In this example, the result is the following (rank value of `1`):

- `EA","VGASP2E","432118","99.99","unspsc","1","USD","Atlanta","VGA Cables",,"sid496","1 to 2 VGA Splitter/Booster Economy HD15F x 3"`
- `"EA","VGASP4","432118","109.99","unspsc","1","USD","Atlanta","VGA Cables",,"sid496","1 to 4 VGA Splitter/Booster HD15F x 5"`

Step 6: Contract Price-based match. Continuing with this example, it is seen that there still are two entries that are matched with the same rank value (`1`). This prompts the buying solution to further filter this result to select the best match between the two entries. This is done by applying the contract price-based rule. Here, the system selects the item master entry that has the lowest price and applies that entry to the imported requisition.

In this example, the following entry is selected and applied to the imported requisition:

```
EA","VGASP2E","432118","99.99","unspsc","1","USD","Atlanta","VGA Cables",,"sid496
","1 to 2 VGA Splitter/Booster Economy HD15F x 3"
```

Item Master

You can create an item master from the user interface or by loading an item master using a CSV file data import task.

Related Information

Creating the Item Master from the Administration User Interface [page 63]
Editing the Item Master Data [page 64]
Loading an Item Master Using the CSV File Import Task [page 65]
Sample Item Master CSV File Formats [page 66]

Creating the Item Master from the Administration User Interface

You can create the item master from the Administration user interface if you are a Customer Administrator or a member of the **Item Master Manager** group.

Procedure

1. On the dashboard, click **Manage** > **Core Administration**.
2. Select **Buyer Item Master Manager** > **Buyer Item Master**.
3. Click **Create New**.
4. In the **General** tab, enter the following details:

Field	Description
Adapter Source	Select the appropriate source of truth for this item master data. The Adapter Source value indicates as to which system owns the data and is the source-of-truth for that data. You can select from: • **SAP AribaManaged**: This is the default adapter source. It indicates that the data is currently be managed though the buying solution and not any external system. • **External**: This indicates that the data is owned by an external/third-party system, such as an ERP system or just a CSV file. • **Manually Maintained**: This indicates that the data is to be maintained manually only through the user interface on the buying solution.
Name	Name of the item master that you are creating.

Field	Description
Buyer Part Number	This is a unique internal identifier within your organization.
Facility	This is the identifier of a specific manufacturing plant, company code, or purchasing unit within the company.
Description	Brief description of the item master that you are creating.
Commodity Code	Select a commodity code for the item master.
Unit of Measure	The unit of measure to be used in this item master. For example, kilogram, liter, decibel, gallon, or grade.

5. Click the **Cross-References** tab. In the page that is displayed, click **Add Items**.
6. The **Catalog Search** page opens. Perform a search for the required customer catalog items and select them. Click **Add to Cart** to add the items to the item master.

> ⓘ Note
>
> Items from contract subscriptions cannot be added to the item master. They are not displayed in the customer catalog search results.

   The **Buyer Item Master** page lists both the selected customer catalog and non-catalog items you created from the customer catalog search page, and their cross references.
7. Click the **Description** link from the list of item master data to view details.
8. Click **Save**.

Editing the Item Master Data

After you load or create item master data, you can make changes to it.

Context

The following procedure assumes you are already on the **Buyer Item Master** page.

Procedure

1. Click the **Edit** button that corresponds to the item master that you want to modify.
2. From the **Edit** page, you can:
   - Edit the general item master information
   - Add more items to cross reference with the selected item master
   - Edit the ranking for each item that is cross referenced
   - Copy an existing item and make appropriate changes

- Delete existing items
- Activate or deactivate a cross referenced item

> **Note**
>
> The buying solution includes only active items when it searches the item master for a best possible match with the imported requisition lines.

3. Click **Save** to save the changes and update the selected item master data.

Loading an Item Master Using the CSV File Import Task

You can create an item master from two CSV files you create and import using a CSV file import task. The CSV files you need to create are `BuyerItemMaster.csv` and `BuyerItemMasterXRef.csv`.

Prerequisites

`BuyerItemMaster.csv` and `BuyerItemMasterXRef.csv` must already exist before you can load them.

You must be a Customer Administrator or a member of the **Item Master Manager** group to load an item master.

Procedure

1. On the dashboard, click **Manage** > **Core Administration**.
2. On the **Welcome to Ariba Administrator** page, click **Buyer Item Master Manager** and then click **Data Import/Export** to display the **Import Item Buyer Master** task.
3. Click **Import**.
4. You can perform the following actions:

Operation	Description
Load	Import new item master data and update existing records.
Create	Only import new item master data. If the CSV files contain data that already exists in the buying solution, those details do not get updated.
Update Only	Only update existing item master data.
Deactivate	Delete existing item master data records.

Sample Item Master CSV File Formats

The **Import Item Buyer Master** task imports item master data into the buying solution from two CSV files: `BuyerItemMaster.csv` and `BuyerItemMasterXRef.csv`.

BuyerItemMaster.csv

The `BuyerItemMaster.csv` file defines master item data.

The following is a sample of the `BuyerItemMaster.csv` file:

```
      "UTF-8"


Name,Description,LocaleID,Facility,BuyerPartNumber,CommodityCodeValue,CommodityCodeD
omain,UnitOfMeasure

"VGA Cables","VGA Cables used for Splitter",,"Bangalore","VGA
Cables","2612","unspsc","EA"
```

BuyerItemMasterXRef.csv

The `BuyerItemMasterXRef.csv` file defines cross-reference data for master item data.

The following is a sample of the `BuyerItemMasterXRef.csv` file:

```
      "UTF-8"


UnitOfMeasure,SupplierPartNumber,CommodityCodeValue,Price,CommodityCodeDomain,Rank,C
urrency,Facility,BuyerPartNumber,SupplierPartAuxiliaryID,CommonSupplierID,Descriptio
n

"EA","VGASP2E","432118","99.99","unspsc","1","USD","Bangalore","VGA
Cables",,"sid496","1 to 2 VGA Splitter/Booster Economy HD15F x 3"
"EA","VGASP4","432118","109.99","unspsc","1","USD","Bangalore","VGA
Cables",,"sid496","1 to 4 VGA Splitter/Booster HD15F x 5"
"EA","VGASP2","432118","79.99","unspsc","1","USD","Bangalore","VGA
Cables",,"sid496","1 to 2 VGA Splitter/Booster HD15F x 3"
"EA","VGA","2612","120","unspsc","1","USD","Bangalore","VGA
Cables","VGA_1234","sid515","VGA Cables for Splitter. Non Catalog Item"
```

Use customer catalog prices in SAP Ariba solutions to replace user-defined prices of customer catalog line items in imported requisitions

Buyers can replace user-specified prices of customer catalog line items in imported requisitions with the prices of the corresponding line items in the customer catalogs of SAP Ariba solutions. This helps buyers be compliant with the customer catalog of the SAP Ariba solution and process PRs and POs efficiently by optimizing the effort required for automating requisitions and managing compliance to purchase orders.

The line items in the requisition imported from an external system are first matched with the customer catalog line items in the SAP Ariba solution. If the `Supplier`, `Supplier Part Number`, `Unit of Measure`, and `Supplier Part Auxiliary ID` (if available) field values of a line item available in an imported requisition form an exact match with an item in the customer catalog of the buying solution from SAP Ariba, the item in the imported external requisition is considered a customer catalog line item. For an imported customer catalog line item, the price in the customer catalog of the SAP Ariba solution is used.

If a line item in the imported requisition doesn't find a match in the customer catalog, it is considered a non-customer catalog item. Items not available in the customer catalog of SAP Ariba Buying Solutions are known as non-customer catalog items. A non-customer catalog item uses the price imported from the external system.

> ⓘ Note
>
> If both **Enable contract auto-selection** (`Application.Contract.MasterAgreement.EnableAutoSelect`) and **Use customer catalog price for an imported requisition line item** (`Application.Procure.UseCustomerCatalogPriceForImportedRequisitionLineItem`) parameters are enabled in the environment, the former takes precedence.
>
> If both **Update prices of customer catalog line items with price configuration** (`Application.Purchasing.UpdatePricesOfCustomerCatalogLineItemsWithPriceConfiguration`) and **Use customer catalog price for an imported requisition line item** (`Application.Procure.UseCustomerCatalogPriceForImportedRequisitionLineItem`) parameters are enabled in the environment, the latter takes precedence.

To enable this feature, you must set the **Use customer catalog price for an imported requisition line item** (`Application.Procure.UseCustomerCatalogPriceForImportedRequisitionLineItem`) parameter to **Yes** in the **Intelligent Configuration Manager** workspace.

For more information, refer to Use customer catalog price for an imported requisition line item

Import of Requisitions with Service Items

A service item is a line item on a requisition that represents a service to be performed by a supplier. Service items follow the service sheet procedure to describe the goods and services that suppliers deliver to fulfill a service order.

You can import requisitions for service items (service requisitions) if your site is enabled to create service requisitions using one of the following options:

- By flagging the **Is Service Item** field on requisitions to mark a line item as a service item. This option is available for sites integrated with either SAP, Generic, or PeopleSoft ERP.
- By using the **Item Category** option on the **Create Non-Catalog Item** page or the requisition summary page to mark a line item on a requisition as a service item. This option is applicable only for sites integrated with SAP ERP.

In addition to the fields required for material items, a service item includes additional fields to flag a line item on a requisition as a service item and include more information required to describe the service (for example, maximum and expected amounts and start and end dates of the service).

To ensure that the tax and discount values are not added again in change orders for imported service requisitions, ensure that the **Prevent addition of tax and discount again in change orders** (`Application.Purchasing.TaxDiscountNotAddedAgainInChangeOrder`) parameter is enabled. See Prevent addition of tax and discount again in change orders.

> **Note**
>
> In the default configuration, when service requisitions from SAP Fieldglass are imported using web services, the SAP Ariba solution overwrites the value for the **Purchasing Unit** field based on the requester's user profile. You can have your site configured to retain the purchasing unit specified on an imported requisition. For help with this configuration, please have your Designated Support Contact log a service request. An SAP Ariba Customer Support representative will follow up to complete the request.

> **Note**
>
> In the default configuration, when SAP Fieldglass service requisitions are imported using web services, requisitions that could not be imported successfully are retained in the **Composing** state. You can choose to enable the **Delete updated SAP Fieldglass requisitions on import failure** (`Application.Purchasing.DeleteUpdatedSAPFieldglassRequisitionsOnImportFailure`) parameter so that requisitions that could not be imported successfully are deleted.

Related Information

Import of Requisitions Using CSV Files [page 69]
Import of Requisitions Using Web Services [page 76]

Import of Requisitions Using Various Integration Channels

You can import requisitions in three ways: using CSV files, using web services, or using cXML posts.

You cannot, however, use both CSV files and web services to import requisitions from two different source systems into the same site.

Related Information

Import of Requisitions Using CSV Files [page 69]
Import of Requisitions Using Web Services [page 76]
Import of Requisitions Using cXML Posts [page 89]

Import of Requisitions Using CSV Files

The **Import Requisitions** import task uses the `RequisitionHeader.csv`, `RequisitionDetail.csv`, and the `RequistionSplitAccounting.csv` files to import requisitions for material and service items.

The following table describes the CSV files used by the **Import Requisitions** import task:

CSV File	Description
`RequisitionHeader.csv`	Contains header information.
`RequisitionDetail.csv`	Contains line item information.
`RequisitionSplitAccounting.csv`	Contains line item split accounting information.

Depending on the action you are performing, make the following changes to the `Operation` field value in the `RequisitionHeader.csv` file when performing a requisition import:

User Action	Corresponding "Operation" Field Value
Creating new requisition	`"New"`
Canceling an existing requisition	`"Cancel"`
Changing/updating an existing requisition	`"Update"`

Related Information

Import of Requisitions Using CSV Files: Generic ERP Examples [page 69]
Import of Requisitions Using CSV Files: PeopleSoft ERP Examples [page 71]
Import of Requisitions Using CSV Files: SAP ERP Examples [page 73]

Import of Requisitions Using CSV Files: Generic ERP Examples

You can import requisitions using CSV files only after you have created and loaded the master item file.

> **Note**
>
> You must include the following fields in your CSV files and specify appropriate values for them:

CSV File	Fields
`RequisitionHeader.csv`	• `Operation`
`RequisitionDetail.csv`	• `BuyerPartNumber` • `Facility`
`RequisitionSplitAccounting.csv`	No change

Following is an example of the `RequisitionHeader.csv` file:

```
UTF8


Requisition_Number,Requester,RequesterPasswordAdapter,Preparer,PreparerPasswordAdapt
er,Title,
OriginatingSystem,OriginatingSystemReferenceID,ShipTo,DeliverTo,Need_By_Date,Operati
on,IsServiceRequisition

52979,mditka,PasswordAdapter1,mditka,PasswordAdapter1,Requisition1,SG,52979,Update
```

> **Note**
>
> - The `Operation` field is required if you want to perform an update or cancel operation on any of your imported requisitions. If you add the `Operation` field but do not specify any value for this field, the buying solution will consider this operation as a new requisition import.
> - The `IsServiceRequisition` field is required to be set to `True` to flag a line item as a service item. For material line items, this field can be left blank.

Following is an example of the details of material line items in the `RequisitionDetail.csv` file:

```
UTF8


Requisition_Number,Line_Number,Supplier,Supplier_Location,Contact_Id,DeliverTo,Need_
By_Date,CommonCommodity_Code,Domain,Commodity_Code,Quantity,Unit_Price,Unit_Of_Measu
re,Currency,Item_Description,ManPartNo,ItemPartNo,SplitAccountingType,ShipTo,BillTo,
ItemAuxPartId
    ,
     ParentLineNumber,TaxCode,DiscountAmount,ChargeType,ChargeAmount
    ,
     Comment,ExternalComment,BuyerPartNumber,Facility,


     UnitConversion,PriceUnitQuantity,PriceUnit,PricingDescription,


     Account_Type

    "ST01",1,"11","","","","","",unspsc,"","10","","EA","USD","S Generic Req 1 Item
1",,9999,,,,1111,,"TC_USA_Tx-USA_Oh_10001",-10,"",,"1","2","EA","Price for 2 boxes"
    "ST01",2,"73","","","","","",unspsc,"","10","","EA","USD","S Generic
Req 1 Item 2",,10119,,,,001,"TC_USA_Tx-USA_Oh_10001",-10,"",,"comment
text","true","KCGeneral","Pittsburgh",,,,"Expense"
```

Here, 9999, and 10119 (`ItemPartNo`) are the **Kit Part ID**s (or **Catalog Supplier Part IDs**) that were specified during kit creation. If the **Kit Aux. Part ID** was also specified during kit creation, that value must be specified in the `ItemAuxPartId` field.

Following is an example of the details of service line items in the `RequisitionDetail.csv` file:

```
UTF8


Requisition_Number,Line_Number,Supplier,Supplier_Location,Contact_Id,DeliverTo,Need_
By_Date,Commodity_Code,CommonCommodity_Code,Domain,Quantity,Unit_Price,Unit_Of_Measu
re,Currency,Item_Description,ManPartNo,ItemPartNo,SplitAccountingType,Account_Assign
ment,Item_Category,ShipTo,BillTo,Purch_Org,Purchase_Group,ParentLineNumber,ExpectedA
mount,MaximumAmount,RequiresServiceEntry,ServiceEndDate,ServiceStartDate,ServiceChil
d,TaxCode,ChargeType,ChargeAmount,DiscountAmount

133,1,0000000100,0000000100,0000000100,,,,76,unspsc,1,82,EA,USD,SAP Req 1 Item
1,Supplier-PartId-
Test1,ItemPartNo2,,P,D,3100,3001,3000,013,,0000000100,9000,TRUE,3/28/2017,1/27/2017,
,,,,-30
133,2,0000000100,0000000100,0000000100,,,00801,14,unspsc,6,20,EA,USD,SAP Req 1 Item
2,SampleManPartId-Test2,ItemPartNo1,,K,,3300,3001,3000,015,1,,,,,TRUE,,,,-20
133,3,,,,,,,,,,,charge1,,,,,,,,,2,,,,,,,HandlingCharge,50,
133,4,,,,,,,,,,,charge2,,,,,,,,,1,,,,,,,HandlingCharge,40,
```

Following is an example of `RequisitionSplitAccounting.csv`:

```
UTF8


Requisition_Number,Line_Number,NumberInCollection,Percentage,Amount,Currency,Quantit
y,Company,
Business_Unit,Cost_Center,Account,Sub_Account,Product,Project,Region

52979,1,1,40,,,,GD,US100,3220,7520,5008,,,
52979,1,2,60,,,,02,US002,4100,6420,,,210,
```

Import of Requisitions Using CSV Files: PeopleSoft ERP Examples

You can import requisitions using CSV files only after you have created and loaded the item master file.

> **Note**
>
> You must include the following fields in your CSV files and specify appropriate values for them:

CSV File	Fields
`RequisitionHeader.csv`	• `Operation`
`RequisitionDetail.csv`	• `BuyerPartNumber` • `Facility`
`RequisitionSplitAccounting.csv`	No change

Following is an example of the `RequisitionHeader.csv` file:

```
UTF8


Requisition_Number,Requester,RequesterPasswordAdapter,Preparer,PreparerPasswordAdapt
er,Title,
OriginatingSystem,OriginatingSystemReferenceID,BusinessUnit,ShipTo,DeliverTo,Need_By
```

```
_Date,Operation,IsServiceRequisition

ST018,mditka,PasswordAdapter1,mditka,PasswordAdapter1,Requisition1,PSoft,ST018,,,,,U
pdate
```

> **Note**
>
> - The `Operation` field is required if you want to perform an update or cancel operation on any of your imported requisitions. If you add the `Operation` field but do not specify any value for this field, the buying solution will consider this operation as a new requisition import.
> - The `IsServiceRequisition` field is required to be set to `True` to flag a line item as a service item. For material line items, this field can be left blank.

Following is an example of the details of material line items in the `RequisitionDetail.csv` file:

```
     UTF8



Requisition_Number,Line_Number,Supplier,Supplier_Location,SupplierLocation_SetID,Con
tact_Id,DeliverTo,Need_By_Date,Commodity_Code,CommonCommodity_Code,Domain,Quantity,U
nit_Price,Unit_Of_Measure,Currency,Item_Description,ManPartNo,ItemPartNo,SplitAccoun
tingType,BuyerPartNumber,Facility,


     ParentLineNumber,TaxCode,DiscountAmount,ChargeType,ChargeAmount,


     UnitConversion,PriceUnitQuantity,PriceUnit,PricingDescription,


     Account_Type

    "ST019",1,"sid496","sid496-hq","SHARE","496-HQ","Deliver
to 3","09/15/2015","SHARE:00017","14","unspsc","4","876","EA","INR","PSoft
Req 1 Item 1",SampleManPartNo,SampleItemPartNo
1,"_Percentage","KCGeneral","Pittsburgh",,"TC_USA_Tx-
USA_Oh_10001",-10,"",,”1”,”2”,”EA”,”Price for 2 boxes”
    "ST019",2,"sid496","sid496-hq","SHARE","496-HQ","Deliver
to 3","09/15/2015","SHARE:00017","14","unspsc","4","876","EA","INR","PSoft
Req 1 Item 2",SampleManPartNo,SampleItemPartNo
1,"_Percentage",”NorthPoint”,”Chicago”,,"TC_USA_Tx-
USA_Oh_10001",-10,"",,,,,,”Expense”
```

Following is an example of the details of service line items in the `RequisitionDetail.csv` file:

```
UTF8


Requisition_Number,Line_Number,Supplier,Supplier_Location,Contact_Id,DeliverTo,Need_
By_Date,Commodity_Code,CommonCommodity_Code,Domain,Quantity,Unit_Price,Unit_Of_Measu
re,Currency,Item_Description,ManPartNo,ItemPartNo,SplitAccountingType,Account_Assign
ment,Item_Category,ShipTo,BillTo,Purch_Org,Purchase_Group,ParentLineNumber,ExpectedA
mount,MaximumAmount,RequiresServiceEntry,ServiceEndDate,ServiceStartDate,ServiceChil
d,TaxCode,ChargeType,ChargeAmount,DiscountAmount

133,1,0000000100,0000000100,0000000100,,,,76,unspsc,1,82,EA,USD,SAP Req 1 Item
1,Supplier-PartId-
Test1,ItemPartNo2,,P,D,3100,3001,3000,013,,0000000100,9000,TRUE,3/28/2017,1/27/2017,
,,,,-30
133,2,0000000100,0000000100,0000000100,,,00801,14,unspsc,6,20,EA,USD,SAP Req 1 Item
2,SampleManPartId-Test2,ItemPartNo1,,K,,3300,3001,3000,015,1,,,,,TRUE,,,,-20
133,3,,,,,,,,,,,,charge1,,,,,,,,,2,,,,,,,,HandlingCharge,50,
```

```
133,4,,,,,,,,,,,,charge2,,,,,,,,,1,,,,,,,,HandlingCharge,40,
```

Following is an example of the `RequisitionSplitAccounting.csv` file:

```
UTF8


Requisition_Number,Line_Number,NumberInCollection,Percentage,Amount,Currency,Quantit
y,Account,
GL_Business_Unit,Location_Code,Department,Product,Statistics_Code

ST018,1,1,70,,,,SHARE:110200,DEU01,SHARE:BUY05,SHARE:13000,SHARE:BUILDING
ST018,1,2,30,,,,SPN01:131000,SPN01,SHARE:YUKON,SHARE:41000,SHARE:NETWORK
```

Import of Requisitions Using CSV Files: SAP ERP Examples

You can import requisitions using CSV files only after you have created and loaded the master item file.

> **Note**
>
> You must include the following fields in your CSV files and specify appropriate values for them:

CSV File	Fields
`RequisitionHeader.csv`	• `Operation`
`RequisitionDetail.csv`	• `BuyerPartNumber` • `Facility`
`RequisitionSplitAccounting.csv`	No change

Following is an example of the `RequisitionHeader.csv` file:

```
    UTF8


Requisition_Number,Requester,RequesterPasswordAdapter,Preparer,PreparerPasswordAdapt
er,Title,OriginatingSystem,OriginatingSystemReferenceID,Operation,IsServiceRequisiti
on

1,jham,PasswordAdapter1,jham,PasswordAdapter1,SAP req import with split by
percentage,SAP,1,New
```

> **Note**
>
> - The `Operation` field is required if you want to perform an update or cancel operation on any of your imported requisitions. If you add the `Operation` field but do not specify any value for this field, the buying solution will consider this operation as a new requisition import.
> - The `IsServiceRequisition` field is required to be set to `True` to flag a line item as a service item. For material line items, this field can be left blank.

Following is an example of the details of material line items in the `RequisitionDetail.csv` file:

```
     UTF8


Requisition_Number,Line_Number,Supplier,Supplier_Location,Contact_Id,DeliverTo,Need_
By_Date,Commodity_Code,CommonCommodity_Code,Domain,Quantity,Unit_Price,Unit_Of_Measu
re,Currency,Item_Description,ManPartNo,ItemPartNo,SplitAccountingType,BuyerPartNumbe
r,Facility,

     ParentLineNumber,TaxCode,DiscountAmount,ChargeType,ChargeAmount,

     UnitConversion,

     PriceUnitQuantity,PriceUnit,PricingDescription,Account_Type

            "1",1,"0000000100","0000000100","0000000100","Sample
DeliverTo","09/15/2015","00107","44102904","unspsc","2","1000","EA","USD","SAP Req
1 Item
1",SampleManPartNo,SampleItemPartNo,"_Percentage",EasternStores,Maranello,,"TC_USA_T
x-USA_Oh_10001",-10,"",,”1”,”2”,”EA”,”Price for 2 boxes”,”Expense”
```

> **Note**
>
> Both the `RequiresServiceEntry` and the `Item_Category` fields are mandatory to import service requisitions in SAP ERP-integrated sites.

Following is an example of the details of service line items in the `RequisitionDetail.csv` file:

```
     UTF8


Requisition_Number,Line_Number,OriginatingSystemLineNumber,Supplier,Supplier_Locatio
n,Contact_Id,DeliverTo,Need_By_Date,CommonCommodity_Code,Domain,Commodity_Code,Quant
ity,Unit_Price,Unit_Of_Measure,Currency,Item_Description,ManPartNo,ItemPartNo,SplitA
ccountingType,Item_Category,Region,BuyerPartNumber,Facility,ParentLineNumber,Expecte
dAmount,MaximumAmount,RequiresServiceEntry,ServiceEndDate,ServiceStartDate,TaxCode,D
iscountAmount,ChargeType,ChargeAmount

"1122",1,25,11,14,63,"Sample
DeliverTo","02/10/2030","51",unspsc,"","1","200","EA","USD","VGA
Cables",SampleManPartNo,SampleItemPartNo,,SE10,"SSP16446","City",,150,20000,True,2/2
8/2017,1/27/2017,"TC5","-100","HandlingCharge",50
"1122",2,25,11,14,63,"Sample
DeliverTo","02/10/2030","51",unspsc,"","20","75","EA","USD","VGA
Cables",SampleManPartNo,SampleItemPartNo,,SE10,"SSP16446","City",1,,,,,"TC5","-100"
,"HandlingCharge",50
"1122",3,25,11,14,63,"Sample
DeliverTo","02/10/2030","1412",unspsc,"","10","150","EA","USD","Adapter DB9M/
MD8F","abc179","NEWHEIRARCHYabc179",,SE10,"SSP16446","City",1,,,,,,"TC5","-100","Han
dlingCharge",50
```

Following is an example of the `RequisitionSplitAccounting.csv` file:

```
UTF8
```

```
Requisition_Number,Line_Number,NumberInCollection,Percentage,Quantity,CompanyCode,Co
stCenter,
GeneralLedger,WBSElement,Asset,InternalOrder,SubNumber,IsWorkOrder

1000252979,1,1,30,,3000,0000001200,0000404000,,,000000100015,true
1000252979,1,2,70,,3000,0000002200,0000404000,3-2200/21,000000002047,000000701677,0,
true
```

Import of Requisitions with Tax and Charge Details

If your site supports taxes and charges on requisitions and purchase orders, you can import requisitions with their tax and charge details by specifying the information in the `RequisitionDetail.csv` file.

The following are the prerequisites for successfully importing requisitions with their tax and charge data:

- Tax codes and charge types used in the `RequisitionDetail.csv` file must be defined in the site.
- For correct calculation of taxes and charges, the tax codes and charge types defined in your buying solution must match those defined in your ERP system.

When you import accounting details for requisition line items, make sure that you don't import any accounting details for charge lines.

> **Note**
>
> If your site does not support taxes and charges on requisitions and purchase orders and you try to import requisitions with tax, charge, or discount data, an error is displayed and the requisition import fails.

You can specify tax and discount details for a requisition line item in the same line as the line item details. For charges, add a new line to the CSV file, and under `ParentLineNumber` field, mention the `Line_Number` value of the requisition line item on which the charge is applied. For kit items, specify the tax, charge, and discount details for the child items.

In the example given in the table below, a requisition PR_50 has two line items - one for a laptop and the other for a monitor. The tax code TCD119 has been applied on the two line items. A discount of USD 13 has been applied on the laptop while a discount of USD 10 has been applied on the monitor. Handling and shipping charges with amounts USD 100 each have been applied on the laptop, while handling and shipping charges with amounts USD 70 and USD 90 have been applied on the monitor. A tax with tax code TC_USA_Tx-ServiceTax has been applied on all these charges.

Requisition_Number	Line_Number	Item_Description	Parent-LineNumber	TaxCode	DiscountAmount	ChargeType	ChargeAmount
PR_50	1	Laptop		TCD119	-13		
PR_50	2		1	TC_USA_Tx-ServiceTax		Handling	100
PR_50	3		1	TC_USA_Tx-ServiceTax		Shipping	100
PR_50	4	Monitor		TCD119	-10		

Requisition_Number	Line_Number	Item_Description	Parent-LineNumber	TaxCode	DiscountAmount	ChargeType	ChargeAmount
PR_50	5		4	TC_USA_Tx-ServiceTax		Handling	70
PR_50	6		4	TC_USA_Tx-ServiceTax		Shipping	90

Import of Requisitions Using Web Services

You can import requisitions for material and service lines using the **Import Requisition(s)** web services task.

Before you can import requisitions using web services, ensure that the following conditions are true:

- Your site is configured to use web services for data integration.
- Your web services request (requisition details sent in XML format, from an external system) includes the following fields, if you've configured and loaded the item master:
  - `BuyerPartNumber` (requisition line level)
  - `Facility` (requisition line level)
  - `Need-by Date` (requisition line level)
  - `Operation` (requisition header level)

  In addition, the following fields are required for service items:
  - `ExpectedAmount`
  - `MaxAmount`
  - `ServiceStartDate`
  - `ServiceEndDate`
  - `RequiresServiceEntry`
  - `Item_Category` (Applicable for SAP ERP-integrated sites where creation of service requisitions based on Item Category is enabled.)
- The `NumberInCollection` field value in the requisition line item details must start with 1. This field contains the line number of the item in the requisition that is being imported.

> **Note**
>
> Enable the **Update requisition status in response file for Import Requisition(s) task** (`Application.Purchasing.UpdateRequisitionStatusInResponseFileForImportRequisitionsTask`) parameter to update the requisition status in the response file when the **Import Requisition(s)** web services task is run for importing multiple requisitions associated with the same contract. For more information, see Update requisition status in response file for Import Requisition(s) task.

A snippet of the XML file that includes these fields is shown as follows:

```
<urn:OriginatingSystemLineNumber>1</urn:OriginatingSystemLineNumber>
                    <urn:BuyerPartNumber>VGA Cables</urn:BuyerPartNumber>
                    <urn:Facility>Bangalore</urn:Facility>
<Extrinsic name="Need-by Date">2013-01-20T00:00:00-08:00</Extrinsic>
```

```
                    </urn:item>
                </urn:LineItems>
                <urn:Name>EOE-WS-Feb21th-cnoll3</urn:Name>
                <urn:OriginatingSystem>SG</urn:OriginatingSystem>
                <urn:OriginatingSystemReferenceID>EOE-WS-Feb21th-cnoll3</
urn:OriginatingSystemReferenceID>
                <urn:Preparer>
                   <urn:PasswordAdapter>PasswordAdapter1</urn:PasswordAdapter>
                   <urn:UniqueName>cnoll</urn:UniqueName>
                </urn:Preparer>
                <urn:Requester>
                   <urn:PasswordAdapter>PasswordAdapter1</urn:PasswordAdapter>
                   <urn:UniqueName>cnoll</urn:UniqueName>
                </urn:Requester>
                <urn:UniqueName>ReqUniqueName-1000</urn:UniqueName>
                <urn:Operation>New</urn:Operation>
             </urn:item>
          </urn:Requisition_RequisitionImportPull_Item>
       </urn:RequisitionImportPullRequest>
    </soapenv:Body>
</soapenv:Envelope>
```

The `OriginatingSystemReferenceID` must be unique if it is used for importing requisitions.

If the `Supplier`, `Supplier Part Number`, `Unit of Measure`, and `Supplier Part Auxiliary ID` (if available) field values of a line item available in an imported requisition form an exact match with an item in the customer catalog of the buying solution from SAP Ariba, the item with the external requisition is considered a customer catalog line item.

To import a requisition containing a customer catalog line item, the price of the line item must match the mapped price in the customer catalog of the buying solution from SAP Ariba. If the prices do not match, the import operation fails with an error message, and the status of the requisition changes to **Composing**. You must correct the line item price manually before resubmitting the import requisition for approval.

To avoid failure of the import operation of a requisition with a customer catalog line item enabled with price configuration, and the price differing from the price of the item in the customer catalog of the buying solution, you can use the **Update prices of customer catalog line items with price configuration** (`Application.Purchasing.UpdatePricesOfCustomerCatalogLineItemsWithPriceConfiguration`) parameter. For more information, see Update prices of customer catalog line items with price configuration. After you enable this parameter, the following information message is displayed when buyers import an external requisition with a customer catalog line item enabled with price configuration and the price differing from the price of the item in the customer catalog of the buying solution:

Some of the imported customer catalog line items with price configuration might have been modified.

To validate the price of imported requisitions containing only material items, the **Validate price for line items in imported requisitions** (`Application.Procure.ValidatePriceForLineItemsInImportedRequisition`) parameter must be enabled. For information about the parameter, see Validate price for line items in imported requisitions.

> **Note**
>
> The validation message might not be displayed for requisitions imported from SAP Fieldglass and for imported requisitions that contain the following types of items:
>
> - Service items
> - Non-catalog items
> - Partial items

- Items associated with a contract

There are some considerations that you must be aware of when importing requisitions. For more information, see Considerations for Importing Requisitions [page 99].

Enhanced Requisition Import Using Web Services

Requisition updates can be imported using web services task (**Import Requisition(s)** or **Import Requisitions Asynchronously**) irrespective of whether the requisition was created or updated in the past using the SAP Ariba solution user interface or the import web services task.

When a requisition in **Ordered** state is updated using the web services task, the **History** tab includes a record that summarizes the changes. A new purchase requisition with `V2` gets created.

> **Note**
>
> Ensure that you send the correct active version of the requisition when you're trying to update an existing purchase requisition. If you try to update an inactive version of an existing requisition, the latest active version of the requisition gets updated. If you try to use the `Update` operation for a requisition that does not exist, an error message is displayed.

Prerequisites

- Ensure that your administrator has enabled the **Enable enhanced requisition import using web services** (`Application.Procure.EnableEnhancedRequisitionImportUsingWebServices`) parameter. See Enable enhanced requisition import using web services.
- In the web services request XML file, the following fields are also required in addition to the `BuyerPartNumber`, `Facility`, `Need-by Date`, and `Operation`:
  - At the requisition line level, `OriginatingSystemLineNumber` or `NumberInCollection`
  - At the requisition header level, `OriginatingSystemReferenceID` or `UniqueName` (purchase requisition ID); `OriginatingSystemReferenceID` is required for all updates if the requisition was originally created in the SAP Ariba solution through the import web services task; only `UniqueName` is required for all updates if the requisition was originally created in the SAP Ariba solution through the user interface.

Restrictions

- The number of line items that can be included in requisitions during import is limited to 500. You can't import requisitions that contain more than 500 line items.
- You can't import more than 500 requisitions at a time using the import web service task.

> **Note**
>
> To import requisitions using the web services task (**Import Requisition(s)** or **Import Requisitions Asynchronously**) when both the **CDN-10259: Renumber NumberInCollection based on OriginatingSystemLineNumber** feature and the feature for adding taxes, charges, and discounts to line items (`Application.Procure.EnableTaxesChargesAndDiscount`) are enabled for your site, ensure that your administrator enables the **Update imported requisitions with tax, charge, and discount lines having same OriginatingSystemLineNumber as parent item** (`Application.Procure.UpdateImportedRequisitionsWithTLCHavingSameOSLNAsParentItem`)

parameter. See Update imported requisitions with tax, charge, and discount lines having same OriginatingSystemLineNumber as parent item.

Error Codes in Web Service Responses for Failed Requisition Import

If **CP-7612** and `Application.Procure.IncludeErrorCodeInWebServiceErrorResponse` parameter (set by SAP Ariba Support) are enabled for your site, web service responses for requisitions that fail to be imported from external ERP systems include unique error codes in addition to error messages.

The following table lists the error codes and the corresponding error messages that are displayed in the web service response for requisitions that fail to be imported from an external ERP system:

Error code	Error message
2000	Requisition was not created in the SAP Ariba solution because requisition `{requisition ID}` with OriginatingSystemReferenceId `{OriginatingSystemReferenceId}` already exists.
2001	Requisition `{requisition ID}` with OriginatingSystemReferenceId `{OriginatingSystemReferenceId}` could not be processed in the SAP Ariba solution because the current status of either the requisition or the associated orders does not support the import operation.
2002	Requisition `{requisition ID}` with OriginatingSystemReferenceId `{OriginatingSystemReferenceId}` could not be cancelled in the SAP Ariba solution due to unexpected errors.
2003	Requisition `{requisition ID}` with OriginatingSystemReferenceId `{OriginatingSystemReferenceId}` cannot be cancelled because either the user does not have the required permissions or a new version of this requisition exists with errors generated in the external system.
2004	Could not perform the update operation for requisition `{requisition ID}` with OriginatingSystemReferenceId `{OriginatingSystemReferenceId}` as this requisition is incorrectly associated with one or more orders in the SAP Ariba solution.
2005	Could not import requisition `{requisition ID}` with OriginatingSystemReferenceId `{OriginatingSystemReferenceId}` because tax code is invalid.

Error code	Error message
2006	Could not import requisition `{requisition ID}` with OriginatingSystemReferenceId `{OriginatingSystemReferenceId}` because charge type is invalid.
2007	Could not import requisition `{requisition ID}` with `{OriginatingSystemRefId}` because discount cannot be greater than `{price * quantity}`.
2008	Could not import requisition `{requisition ID}` with `{OriginatingSystemRefId}` because discount must be a negative number
2009	Could not import requisition `{requisition ID}` with OriginatingSystemReferenceId `{OriginatingSystemReferenceId}` because charge amount must be a positive number
2010	Could not import requisition `{requisition ID}` with OriginatingSystemReferenceId `{OriginatingSystemReferenceId}` because charge amount is not specified.
2011	Could not import requisition `{requisition ID}` with OriginatingSystemReferenceId `{OriginatingSystemReferenceId}` because it contains discount, charge, or tax details. Requisitions cannot have discount, charge, or tax details if the parameter `Procure.EnableTaxesChargesAndDiscount` is disabled.
2012	Could not import requisition because contract id `{contract ID}` with `{OriginatingSystemRefId}` is invalid.
2013	Could not import requisition because contract id `{contract ID}` is not matching for selected line `{requisition line number}` with requisition OriginatingSystemReferenceId `{OriginatingSystemReferenceId}`.
2014	Could not import requisition because contract with contract id `{contract ID}` with requisition's `{OriginatingSystemRefId}` is not active.

Error code	Error message
2015	Could not import requisition `{requisition ID}` with OriginatingSystemReferenceId `{OriginatingSystemReferenceId}` because service lines in the requisition failed validation.
2016	Could not import requisition `{requisition ID}` with OriginatingSystemReferenceId `{OriginatingSystemReferenceId}` because vendor (supplier) information for one or more parent service lines is missing.
2017	Could not import requisition `{requisition ID}` with OriginatingSystemReferenceId `{OriginatingSystemReferenceId}` because max amount and expected amount for unplanned service lines are missing.
2018	`{Requisition ID}` (OriginatingSystemReferenceID: `{OriginatingSystemReferenceId}`) has one or more active amendments associated with it. Requisition could not be created in the SAP Ariba solution.
2019	Could not import requisition `{requisition ID}` with OriginatingSystemReferenceId {OriginatingSystemReferenceId} because it contains WorkOrderID but `Application.Purchasing.WorkOrderReqRfqSupport` is disabled.
2020	Service requisition is not supported in Excel import of requisition.
2021	Preparer of requisition `{requisition ID}` does not match the current user.
2022	The contract limit on the service line of the requisition `{requisition ID}` is greater than max amount.
2023	The other limit on the service line of the requisition `{requisition ID}` is greater than max amount.
2026	Simultaneous Amendments feature is not enabled for your site. Ensure that the Simultaneous Amendment and Import Simultaneous Amendment Web Service features are enabled.
2027	Could not import the amendment because either preparer or requester information is missing.

Error code	Error message
2028	Amend operation could not be performed. This operation is allowed only for requisitions that are in the **Ordered** status.
2030	One or more lines are being processed in another amendment.
2031	Could not process the amend operation because the site configuration parameter to allow creation of amendments for header-level fields is not enabled.
2032	Values for certain header-level fields, such as Purchasing Unit or Business Unit should not be modified when amendments are imported.
2033	Amendments with missing line items cannot be imported. Include at least one line item for which an amendment is to be created.
2034	Amendment `{requisition ID}` with OriginatingSystemReferenceID `{OriginatingSystemReferenceId}` and OriginatingSystemAmendmentID `{OriginatingSystemAmendmentID}` could not be cancelled in the SAP Ariba solution due to unexpected errors.
2035	Value for `{Mandatory Field Name}` is missing.
2036	Could not find the requisition matching the ID `{OriginatingSystemReferenceID}`.
2037	Could not cancel amendment `{OriginatingSystemAmendmentID}` since it is in **Merged** state.
2038	Could not cancel amendment `{OriginatingSystemAmendmentID}` since it is in **Submitted** state.
2039	MRP Run requisitions cannot be edited in SAP ERP after they are imported in the SAP Ariba solution. You can edit the requisition only in the SAP Ariba solution.
2040	Could not import retroactive amendment `{requisition ID}` with OriginatingSystemReferenceId `{OriginatingSystemReferenceId}`. Preparer is not authorized to create retroactive amendments.

Error code	Error message
2041	Could not import requisition amendment `{requisition ID}` with OriginatingSystemReferenceId `{OriginatingSystemReferenceId}` because the retroactive effective date is invalid. The date must be on or between the initial ordered date and the current date.
2042	The import failed as the ERSAllowed flag for the imported line item(s) `{Lineitem Numbers}` differs from the previous version in the buying solution.
2043	The import failed for requisition `{requisition ID}` with OriginatingSystemReferenceId `{OriginatingSystemReferenceId}` because the subcontractor flag of line item `{Lineitem Number}` differs from the previous version in the buying solution.
2045	Requisition `{requisition ID}` cannot be updated using the import task as the associated purchase order is closed for all actions such as change order receiving and invoicing.
2046	`{Requisition ID}` cannot be imported as the number of line items exceeds the limit configured for your site. The maximum limit for line items in a requisition during import is `{Maximum Lineitem Count Configured}`.
2047	Requisition `{requisition ID}` cannot be imported as it contains attachments. You cannot import requisitions with attachments using CSV or Excel files.
2048	Could not import requisition `{requisition ID}` with OriginatingSystemReferenceId `{OriginatingSystemReferenceId}` because service line has zero price.
2049	Requisition `{requisition ID}` cannot be updated using the import task as it is in the **Canceling** or **Canceled** state.
2050	Could not perform Update operation on requisition `{requisition ID}` as it does not exist.
2051	Invalid keyword - `{keyword}`. Depending on your task, use one of the predefined keywords - New, Update, Cancel, Amend.
2052	PR-ID is system generated, you cannot pass the same in case of new operation.
2053	UniqueName or OriginatingSystemReferenceID is mandatory for update, amend, and cancel operations.

Related Information

Retain line-level comments and attachments in updated requisition for Import Requisition(s) web services task

Import of Requisitions with Attachments Using Web Services

You can import requisitions that contain attachments using the **Import Requisition(s)** web services task.

While importing requisitions,

- You can include attachments at the header-level or at the line-level.
- You can add new attachments, update existing attachments, or delete previous attachments.
- You can attach multiple attachments.

While importing a requisition, you can optionally indicate whether the attachment is external or internal using the `ExternalAttachment` field. If the field value is set to `true`, the attachment is considered as external. If set to `false`, the attachment is considered as internal. After the successful import, the **Visible to Supplier on Order** check box appears selected for the attachments that are marked as external. These external attachments can be viewed by suppliers.

The maximum size of the attachments and the type of attachments that can be sent depend on your site configuration. The values set for the `Application.Approvable.MaxAttachmentSize` parameter (set by SAP Ariba) and the `Application.Approvable.AllowedAttachmentExtensions` parameter (set by SAP Ariba) determine the validity of the attachments.

> **Note**
>
> When you run the **Import Requisition(s)** web services task for a set of requisitions, we recommend that you attach less than 10 attachments to a requisition and ensure that the cumulative size of all the attachments across requisitions is less than 100 MB. To avoid performance issues, it's recommended that you import 1 requisition at a time.

A snippet of the XML file that includes the attachment fields is shown:

```
<urn:Attachments>
<urn:item>
    <urn:Attachment>
        <urn:ContentType>text/plain</urn:ContentType>
        <urn:Filename>HeaderAttach.txt</urn:Filename>
    </urn:Attachment>
    <urn:Date>2021-06-16T07:39:46Z</urn:Date>
    <urn:ExternalAttachment>false</urn:ExternalAttachment>
    <urn:MappedFilename>cid:HeaderAttach.txt</urn:MappedFilename>
    <urn:User>
        <urn:PasswordAdapter>PasswordAdapter1</urn:PasswordAdapter>
        <urn:UniqueName>smith</urn:UniqueName>
    </urn:User>
</urn:item>
<urn:item>
    <urn:Attachment>
        <urn:ContentType>text/plain</urn:ContentType>
        <urn:Filename>LineAttach.txt</urn:Filename>
    </urn:Attachment>
    <urn:Date>2021-06-16T07:39:46Z</urn:Date>
    <urn:ExternalAttachment>true</urn:ExternalAttachment>
```

```
    <urn:LineNumber>1</urn:LineNumber>
    <urn:MappedFilename>cid:LineAttach.txt</urn:MappedFilename>
    <urn:User>
        <urn:PasswordAdapter>PasswordAdapter1</urn:PasswordAdapter>
        <urn:UniqueName>smith</urn:UniqueName>
    </urn:User>
</urn:item>
</urn:Attachments>
```

Prerequisites

- Your administrator must enable the following parameters:
  - **Enable attachment support in data import web services** (`Application.Messaging.Channels.WS.SupportAttachmentsInWSImports`)
  - **Import requisitions with attachments using web services** (`Application.Procure.ImportRequisitionsWithAttachmentsUsingWebServices`). See Import requisitions with attachments using web services.
  - **Import requisitions using OriginatingSystemLineNumber with attachments using web service** (`Application.Procure.ImportRequisitionsUsingOSLNWithAttachmentsUsingWebServices`). This parameter must be enabled if you plan to use the `OriginatingSystemLineNumber` while importing requisitions with attachments. See Import requisitions using OriginatingSystemLineNumber with attachments using web service.
- A member of the **Integration Admin** group must review and accept the data definition changes for this feature.

> **Note**
>
> Disabling the feature after accepting the data definition changes will cause requisition import failures. You must revert the data definition changes manually.

Restrictions

- You can't import requisitions with attachments using web services if your site is integrated to your external ERP using SAP Integration Suite, managed gateway for spend management and SAP Business Network.
- When the feature is enabled for your site, the fields related to attachments are visible when you import requisitions using CSV or Excel files. Even though the fields are visible, you can't import requisitions with attachments using CSV or Excel files.
- You can't import requisitions with attachments if both the `CDN-10259: Renumber NumberInCollection based on OriginatingSystemLineNumber` feature, and the feature for adding taxes, charges, and discounts to line items (`Application.Procure.EnableTaxesChargesAndDiscount`) parameter are enabled for your site.
- If the `CDN-10259: Renumber NumberInCollection based on OriginatingSystemLineNumber` feature is enabled for your site, you can use only the `OriginatingSystemLineNumber` while importing requisitions with attachments. You can't use `NumberInCollection`.

Import of Requisitions Asynchronously for Sites Integrated with SAP ERP Using SAP Integration Suite, managed gateway for spend management and SAP Business Network

Requisitions from sites integrated with SAP ERP using SAP Integration Suite, managed gateway for spend management and SAP Business Network can now be imported asynchronously using the new **Import Requisitions Asynchronously** web services task.

These requisitions include autogenerated requisitions, such as requisitions created automatically during the planning run for Materials Requirement Planning (MRP) in SAP ERP. Importing requisitions asynchronously improves the reliability of the import process for large number of requisitions and prevents timeout problems, when compared to importing requisitions synchronously using other import channels.

Prerequisites [page 86]

Restrictions [page 86]

Web Services Tasks [page 87]

Workflow for Importing Requisitions from SAP ERP Asynchronously [page 88]

Prerequisites

- This feature is applicable for sites integrated with SAP ERP using SAP Integration Suite, managed gateway for spend management and SAP Business Network.
- To enable this feature, have your Designated Support Contact (DSC) submit a case. SAP Support will follow up to complete the case.
- Imported requisitions do not include tax and charge lines. Users must ensure that they add the appropriate taxes and charges before they submit the requisitions for approval.

Restrictions

- It is recommended that users should not make changes to requisitions in SAP ERP after those requisitions are imported in the SAP Ariba solution. In the case of MRP-run requisitions that are in the **Approved** or later states in the SAP Ariba solution, errors are returned when users attempt to make changes to the corresponding requisitions in SAP ERP.

  > ⚠ Caution
  >
  > Data entered on an updated requisition in SAP ERP might be lost after the associated requisition imported in the SAP Ariba solution is submitted for approval. This scenario can occur typically when a user updates the requisition in SAP ERP before the associated requisition imported in the SAP Ariba solution is processed. After a user submits the imported requisition for approval, the requisition undergoes budget checks in SAP ERP and upon successful completion of budget checks, this requisition overwrites the requisition in SAP ERP.

- When a change order process is initiated based on an MRP-run requisition in the buying solution, held line items might be deleted from the original requisition in SAP ERP when the preparer or requester edits or

withdraws the requisition. In this scenario, duplicate requisitions might be created for the deleted held line items in SAP ERP during the next MRP-run process.

> **Note**
>
> It is recommended that preparers and requesters must not edit or withdraw an MRP-run requisition during the change order process.

- For MRP-run requisitions that are in the **Approved** (or later) states in the SAP Ariba solution, changes cannot be made to the line items on the corresponding requisition in SAP ERP.
- This feature does not support requisitions consisting of service items.

Web Services Tasks

Import Requisitions Asynchronously

The new **Import Requisitions Asynchronously** web services task imports requisitions from SAP ERP using SAP Integration Suite, managed gateway for spend management and SAP Business Network.

The following is a sample of the acknowledgment sent to SAP ERP after requisitions are imported to the SAP Ariba solution:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:Ariba:Buyer:vrealm_123">
   <soapenv:Header>
      <urn:Headers>
         <!--You may enter the following 2 items in any order-->
         <!--Optional:-->
         <urn:variant>?</urn:variant>
         <!--Optional:-->
         <urn:partition>?</urn:partition>
      </urn:Headers>
   </soapenv:Header>
   <soapenv:Body>
      <urn:RequisitionAsyncImportPullStatusReply>
         <urn:status>Ok/urn:status>
      </urn:RequisitionAsyncImportPullStatusReply>
   </soapenv:Body>
</soapenv:Envelope>
```

Export Response for Import Requisitions Task Status Asynchronously

The new **Export Response for Import Requisitions Task Status Asynchronously** web services task exports the success or failure response for each imported requisition depending on whether or not the requisition was successfully created in the SAP Ariba solution.

The SAP Ariba solution sends this response to SAP ERP after the **Import Requisitions Asynchronously** web services task imports requisitions from SAP ERP.

The following is a sample of the response sent to SAP ERP:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance">
   <soap:Header>
      <Headers xmlns="urn:Ariba:Buyer:vrealm_123">
         <variant>vrealm_123</variant>
         <partition>prealm_123</partition>
      </Headers>
   </soap:Header>
   <soap:Body>
      <RequisitionAsyncImportPullStatusRequest partition="prealm_123"
variant="vrealm_123" xmlns="urn:Ariba:Buyer:vrealm_123">
         <RequisitionAsyncImportPullStatus_RequisitionAsyncImportPullResponse_Item>
            <item>
               <EntityReferenceID>peRI-ws-25Oct-1b</EntityReferenceID>
               <Requisition>
                  <ERPRequisitionID/>
                  <OriginatingSystem>SAP</OriginatingSystem>
                  <OriginatingSystemAmendmentID/>
                  <StatusString>Composing</StatusString>
                  <UniqueName>PR121</UniqueName>
               </Requisition>
            </item>
         </RequisitionAsyncImportPullStatus_RequisitionAsyncImportPullResponse_Item>
      </RequisitionAsyncImportPullStatusRequest>
   </soap:Body>
</soap:Envelope>
```

Workflow for Importing Requisitions from SAP ERP Asynchronously

The following is an overview of the high-level steps involved in importing requisitions asynchronously from SAP ERP:

1. The **Import Requisitions Asynchronously** web services task imports purchase requisitions from SAP ERP via SAP Integration Suite, managed gateway for spend management and SAP Business Network.
2. The buying solution performs the following tasks:
    1. Queues the request consisting of data related to the imported requisitions and sends an acknowledgment to SAP ERP.

       > **Note**
       >
       > If a requisition is imported successfully and an acknowledgement is sent to the ERP, don't attempt to import the same requisition again.

    2. Processes the data for the requisitions and creates each requisition in a sequence.
       If the request could not be processed, the **Process Queued task for Async Import Requisition Pull Request** scheduled task retries processing the request. The default number of retries is `21`, which is the value set in the `Application.Purchasing.RetryCountToProcessFailedAsyncRequisitionImportRequest` parameter. See Set number of times to process failed async requisition import request.
3. Depending on the processing status, the **Export Response for Import Requisitions Task Status Asynchronously** sends a success or failure response specific to each requisition to the ERP system.
   If the success or failure response could not be sent due to errors, the **Resend failed confirmation response for Import Requisitions Asynchronously event** scheduled task retries sending the response to the ERP system.

Import of Requisitions Using cXML Posts

Before you can import requisitions using cXML Posts, ensure that your site is configured to use XML for data integration.

The cXML channel applies only to SAP Business Network and not from the ERP system to your SAP Ariba solution.

Four fields (**Buyer Part Number**, **Facility**, **Need By Date**, and **Type**) must appear in the "`PurchaseRequisitionRequest`" cXML file format.

The following cXML examples illustrate the header and item detail sections along with the new fields that must be added when importing new requisitions, when updating an imported requisition, and when canceling or deleting an imported requisition.

Example: Importing a New Requisition

At the header level:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE cXML SYSTEM "http://svcdev6.ariba.com/schemas/cXML/1.2.026/cXML.dtd">
<cXML payloadID="testreq" timestamp="2014-06-13T12:10:00" xml:lang="en-US">
    <Header>
        <From>
            <!-- The buying marketplace and member organization. -->
            <Credential domain="NetworkID">
                <Identity>AN71000002012</Identity>
            </Credential>
        </From>
        <To>
            <Credential domain="NetworkID">
                <Identity>AN71000002012</Identity>
            </Credential>
        </To>
        <Sender>
            <!-- This document has passed through Ariba CSN to the supplier. -->
            <Credential domain="NetworkID">
            <Identity>AN01000000001</Identity>
                <SharedSecret>welcome1a</SharedSecret>
            </Credential>
            <UserAgent>Buyer ERP</UserAgent>
        </Sender>
    </Header>

    <Request>
        <PurchaseRequisitionRequest>
            <PurchaseRequisition>
                <PurchaseRequisitionHeader requisitionID="PR123"
requisitionDate="2015-02-26T00:00:00-08:00" type="new">
                    <ShipTo>
                        <Address addressID="3000">
                            <Name xml:lang="en">New York</Name>
                            <PostalAddress>
                                <DeliverTo>Joe Smith</DeliverTo>
                                <Street>691 Broadway</Street>
                                <City>New York</City>
                                <State>NY</State>
                                <PostalCode>10001</PostalCode>
                                <Country isoCountryCode="US">USA</Country>
                            </PostalAddress>
                        </Address>
                    </ShipTo>
                    <BillTo>
                        <Address addressID="US006">
                            <Name xml:lang="en">New York</Name>
                            <PostalAddress>
```

```
                                <Street>691 Broadway</Street>
                                <City>New York</City>
                                <State>NY</State>
                                <PostalCode>10001</PostalCode>
                                <Country isoCountryCode="US">USA</Country>
                            </PostalAddress>
                        </Address>
                    </BillTo>
                    <Contact role="preparer">
                        <Name xml:lang="en-US">Chad Noll</Name>
                        <PostalAddress>
                            <Street>jUnitDummy</Street>
                            <City>Sunnyvale</City>
                            <State>CA</State>
                            <PostalCode>94089</PostalCode>
                            <Country isoCountryCode="US">United States</Country>
                        </PostalAddress>
                        <Email>cnoll@ariba.com</Email>
                    </Contact>
                    <Contact role="requester">
                        <Name xml:lang="en-US">Chad Noll</Name>
                        <PostalAddress>
                            <Street>jUnitDummy</Street>
                            <City>Sunnyvale</City>
                            <State>CA</State>
                            <PostalCode>94089</PostalCode>
                            <Country isoCountryCode="US">United States</Country>
                        </PostalAddress>
                        <Email>cnoll@ariba.com</Email>
                    </
Contact>
                </PurchaseRequisitionHeader>
```

At the line level:

```
<ItemIn quantity="10.000" lineNumber="00001">
                    <ItemID>
                        <SupplierPartID>MON923 6</SupplierPartID>
                    </ItemID>
                    <ItemDetail>
                        <UnitPrice>
                            <Money currency="USD">100.00</Money>
                        </UnitPrice>
                        <Description xml:lang="en">Amiga Monitor Cable DB9M/DB23F </
Description>
                        <UnitOfMeasure>EA</UnitOfMeasure>
                        <Classification domain="UNSPSC">43211800</Classification>
                        <Extrinsic name="AccountCategory">K</Extrinsic>
                        <Extrinsic name="PurchaseOrg">3000</Extrinsic>
                        <Extrinsic name="PurchaseGroup">100</Extrinsic>
                        <Extrinsic name="BuyerPartNumber">SSP16446-cXML</Extrinsic>
                        <Extrinsic name="Facility">Bangalore</Extrinsic>
                    </ItemDetail>
                    <SupplierList>
                        <Supplier>
                            <Name xml:lang="en">JCN Technologies</Name>
                            <SupplierID domain="NetworkID">AN70000000004</
SupplierID>
                        </Supplier>
                    </SupplierList>
                    <Distribution>
                        <Accounting name="Default">
                            <AccountingSegment id="100">
                                <Name xml:lang="en">Percentage</Name>
                                <Description xml:lang="en">Percentage</Description>
                            </AccountingSegment>
                            <AccountingSegment id="02">
```

```
                                <Name xml:lang="en">Company</Name>
                                <Description xml:lang="en">ID</Description>
                            </AccountingSegment>
                            <AccountingSegment id="5000">
                                <Name xml:lang="en">CostCenter</Name>
                                <Description xml:lang="en">ID</Description>
                            </AccountingSegment>
                            <AccountingSegment id="US002">
                                <Name xml:lang="en">BusinessUnit</Name>
                                <Description xml:lang="en">ID</Description>
                            </AccountingSegment>
                            <AccountingSegment id="8100">
                                <Name xml:lang="en">Account</Name>
                                <Description xml:lang="en">ID</Description>
                            </AccountingSegment>
                            <AccountingSegment id="5009">
                                <Name xml:lang="en">SubAccount</Name>
                                <Description xml:lang="en">ID</Description>
                            </AccountingSegment>
                        </Accounting>
                        <Charge>
                            <Money currency="USD">20000.00</Money>
                        </Charge>
                    </Distribution>
                </ItemIn>
            </PurchaseRequisition>
        </PurchaseRequisitionRequest>
    </Request>
</cXML>
```

Example: Updating an Imported Requisition

At the header level:

```
</Header>

     <Request>


             <PurchaseRequisitionRequest>


                 <PurchaseRequisition>

                <PurchaseRequisitionHeader requisitionID="PR123"
requisitionDate="2012-02-26T00:00:00-08:00"
     type="update"
    >
```

At the line level:

```
<ItemDetail>
                <UnitPrice><Money currency="USD">200.00</Money></UnitPrice>
                <Description xml:lang="pt-br">VGA Cables</Description>
                <UnitOfMeasure>Box</UnitOfMeasure>
                <Classification domain="UNSPSC">10101804</Classification>
                <Extrinsic name="Account Type">K</Extrinsic>

     <Extrinsic name="BuyerPartNumber">VGA Cables</Extrinsic>


                     <Extrinsic name="Facility">Sunnyvale</Extrinsic>

<Extrinsic name="Need-by Date">2013-01-20T00:00:00-08:00</Extrinsic>
            </ItemDetail>
```

Example: Canceling or Deleting an Imported Requisition

At the header level:

```
</Header>

     <Request>


             <PurchaseRequisitionRequest>


                 <PurchaseRequisition>

                <PurchaseRequisitionHeader requisitionID="PR123"
requisitionDate="2012-02-26T00:00:00-08:00"
     type="delete"
    >
```

At the line level:

```
<ItemDetail>
                <UnitPrice><Money currency="USD">200.00</Money></UnitPrice>
                <Description xml:lang="pt-br">VGA Cables</Description>
                <UnitOfMeasure>Box</UnitOfMeasure>
                <Classification domain="UNSPSC">10101804</Classification>
                <Extrinsic name="Account Type">K</Extrinsic>

     <Extrinsic name="BuyerPartNumber">VGA Cables</Extrinsic>


                     <Extrinsic name="Facility">Sunnyvale</Extrinsic>

                        <Extrinsic name="Need-by Date">2013-01-20T00:00:00-08:00</
Extrinsic>
</ItemDetail>
```

Related Information

cXML Solutions Guide

Import of Requisitions with Kits

Consider a business scenario where your company has to procure a pre-defined set of items frequently for its new hires. This may include the following:

- A laptop (specific brand with the required configurations)
- 15 inch LCD monitor
- Battery pack
- Docking station
- Optical mouse

- Multimedia keyboard

Using the option of importing requisitions with kits, you can create a kit with all these items (a one time activity) in the SAP Ariba solution, and add this kit (with the same **Kit Part ID**) during requisition creation in the external ERP system). This task is easier and consistent when compared to a scenario where you have to otherwise add each item as a separate line, every time you need to procure these items.

When creating kits, the **Kit Aux. Part ID** field is optional, but if you specify a value in this field, then you must specify also this value in the `ItemAuxPartId` column of the `RequisitionDetail.csv` file.

When importing kits, the buying solution looks for an exact match between the kit and the corresponding values specified in the `RequisitionDetail.csv` file. Consider an example where you have specified a **Supplier** and/or **Kit Aux. Part ID** for the kit, but did not add these values in the `RequisitionDetail.csv` file. In this scenario, the import is successful, but the requisition is created in the buying solution with the kit items imported as non-catalog items.

You can also import requisition with kits that have advanced pricing details. You must specify the values for the advanced pricing details in the `RequistionDetail.csv`.

After the kit is added to the requisition, the requisition can be imported into the SAP Ariba solution, where you can proceed with the approval and subsequent order fulfillment process.

To import requisitions using the SAP Ariba data transfer tool: Archive the requisition CSV files in a ZIP file (named `Import Requisitions.zip`) and run a transactional data upload operation and specify “Import Requisitions” in the `event` option.

Related Information

Customer Catalog Administration Guide for Buyers

Importing Requisitions with Kits

You can import requisitions with kits if you are an Administrator or Catalog Manager.

Procedure

1. Create the required kit in the buying solution.
2. On the dashboard, click **Core Administration** > **Catalog Manager**.
3. On the **Catalog Manager** page, click **Catalog Kits**.
4. On the **Catalog Kits** page, click **Create New**.
5. After creating the kit, specify the **Kit Part ID** value in the `ItemPartNo` column of the **RequisitionDetail.csv** file.

> **Note**
>
> If you have specified a **Kit Aux. Part ID** and/or **Supplier** value for the kit, ensure that you include these values in the `RequisitionDetail.csv` file as well.

6. Run the `Import Requisitions` import task.

Results

> **Note**
>
> The requisitions that are created in the buying solution after the import follow the same workflow and approval process as requisitions created manually in the user interface of the buying solution.

Exporting Kits to an External ERP System

You can export all kit information from your buying solution into the external ERP systems, thus ensuring all kits created in your external ERP systems are synchronized and current.

Context

The following limitations apply exporting kits:

- Total kit price is not exported. You can calculate the total price in an external system by taking the individual kit component prices into account.
- Only active kits are exported.
- Custom fields in customer catalogs (if part of the kit items) are not exported.

Procedure

1. On the dashboard, click **Core Administration** > **Site Manager**.
2. On the **Site Manager** page, click **Data Import/Export**.
3. Select the **Export** tab.
4. Search for **Export Kits**.
5. Click **Export**.

Results

Export Kits exports the kit information in a ZIP file - `KitExport.zip`. This ZIP file contains the following two CSV files:

- `Kit _Export.csv`
- `KitItem_Export.csv`

Example data: Kit_Export.csv and KitItem_Export.csv

The `KitExport.zip` file produced by **ExportKits** includes `Kit_Export.csv` and `KitItem.csv`.

Kit_Export.csv

The following is a sample of the `Kit_Export.csv` file:

```
"UTF-8"
Kit ID,Status,Kit Aux. Part ID,Description,Short Description,Time Updated,Kit Part
ID,ETA (Days),Expiration Date
"KIT1","Active","test kit 1","test kit 1","test kit 1","Tue Dec 13 11:07:36
GMT+05:30 2011","test kit 1","3","Wed Dec 14 00:00:00 GMT+05:30 2011"
"KIT2","Active","123456","tetdec","test1","Tue Dec 13 11:57:47 GMT+05:30
2011","1234","3","Thu Dec 29 00:00:00 GMT+05:30 2011"
```

KitItem_Export.csv

The following is a sample of the `KitItem_Export.csv` file:

```
"UTF-8"
Kit.UniqueName,Quantity,Supplier,Currency,Manufacturer,PriceUnitQuantity,SupplierAux
.PartID,UnitConversion,ItemDescription,UnitOfMeasure,SupplierPartID,Mfg.PartID,Price
,Required,KitID,PriceUnit,PricingDescription
"KIT1","40","JCN Technologies","USD","NEWHEIRARCHY",”10”,,”0.2”,"VGA Monitor
HD15M/5-BNC","Ea","NEWHEIRARCHYabc237","abc237","39.95","Yes","KIT1",”2”, “For 2
boxes”
```

Import of Requisitions with User Set Fields

In its default behavior, when an imported requisition is changed or edited, the defaulting mechanism built into the buying solution automatically overwrites the existing, imported, values of all related fields in that requisition with the corresponding values existing in the system.

If you do not want the buying solution to override the imported requisition values on requisition change or edit, you must ensure that you are importing the requisitions with the fields marked as “user set.” This would mean that the

buying solution treats all imported requisition field values as set by the user and retains the associated field values even when the imported requisition is changed or edited.

For example, in the default configuration, if a user/approver edits the quantity value of an imported requisition having the item category as "service," the defaulting mechanism in the buying solution automatically replaces all existing values for that requisition with the corresponding values that are available in the system.

If your buying solution is configured to import requisitions with the fields marked as "user set," automatic overriding of imported values by the defaulting mechanism does not happen and the buying solution retains the values that were originally imported. Contact SAP Ariba Customer Support to enable this feature.

> **Note**
>
> If there are certain fields in the imported requisitions that are left blank (for example, optional accounting combination fields), such fields get overridden by the defaulting mechanism even after the feature has been enabled.

To ensure the defaulting mechanism does not override such field values even after the feature has been enabled, you must specify appropriate values in the CSV file when importing requisitions.

If there are inconsistencies in the data that you are importing, the buying solution does NOT trigger the defaulting mechanism to replace incorrect values even after the fields have been marked as "user set." Examples of inconsistencies in imported data:

- Supplier and Supplier Location information is incorrect
- Accounting details are incorrect
- Commodity Code mapping (partitioned commodity code to common commodity code) is incorrect

In such scenarios, the buying solution displays a warning message stating the imported data is incorrect and retains the requisition in the **Composing** state till you manually resolve these inconsistencies and submit the requisition again.

> **Note**
>
> When importing requisitions, if you have specified unassociated accounting field values for Account Assignment, then the buying solution nullies such field values when creating the requisition. This is done to ensure that there are no validation errors and the imported requisitions are not stuck in the **Composing** state.

For example, when importing, if you specified Account Assignment to be **K (Cost Center)** and then specified the following accounting values:

- **GL Account** - valid for K (Cost Center)
- **Cost Center** - valid for K (Cost Center)
- **Project/WBS** - invalid [or not associated with K Cost Center)]

In this scenario, the buying solution automatically deletes the unassociated accounting field values (in this example, **Project/WBS**) and creates the requisition without any validation errors.

Import of Requisitions with Account Type Data

You can import requisitions with the appropriate Account Type data. This eliminates the manual process of checking every imported line item for the correct account type.

If the imported `RequisitionDetail.csv` file includes account type data, the buying solution validates it to ensure that the imported account type is valid for the corresponding commodity code.

When account type value is included in the imported requisition data:

1. The buying solution validates the account type values. To perform the validation, it compares the imported account type values with the data in the `ERPCommodityCodeMap.csv` file.
2. If the account type specified in the imported requisition data matches one of the commodity code entries in the `ERPCommodityCodeMap.csv` file, the requisition is created and processed to the **Submitted** state for approval.
3. If the account type specified in the imported requisition data is not a valid account type for the specified commodity code, or if the account type value is not specified (left blank) in the imported requisition line, then the requisition is created and retained in the **Composing** state. A notification is sent to the preparer.
4. The preparer must make the required changes to the account type information and resubmit the requisition.

Import of Contracts with Requisitions

You can import contracts along with the requisitions by specifying the contract ID. The imported contract is associated with the imported requisition.

To be able to import contracts when importing requisitions, the corresponding changes made to the requisition import data definitions must be accepted. Customer administrators belonging to the **Integration Admin** group can review and accept data definition changes. These changes are available for both the file channel and the web services channel. You can specify a contract ID in the `ContractId` field in the `RequisitionDetail.csv` file for the file channel, or in the `ContractIdStaging` tag for each line item, for the web services channel.

Import of Requisitions with Subcontractor Information for Suppliers

If enabled on your site, you can include the subcontractor value for the supplier when you import requisition details data (`RequisitionDetail.csv`) from external systems. You can now include a field (`SubContractor`) that indicates whether the supplier is a subcontractor.

The field must be set to one of the following values:

- `1`: If the supplier is a subcontractor
- `-1`: If the supplier is not a subcontractor
- `0` or null value: If the subcontractor value is not specified

ERS Flag Value in Imported Requisitions

If you have enabled the evaluated receipt settlement (ERS) feature for your site, you can additionally configure your site to retain the ERS flag value while importing requisitions.

With the ERS feature enabled, the supplier data that is imported into the buying solution from SAP Ariba includes a flag that indicates whether a supplier is configured for ERS. In the default configuration, the flag value set in the supplier data superseded the value that was set in the imported purchase requisitions. If the flag value in the imported requisition is different from the value in the supplier data, the value in the requisition is replaced by the supplier data value. You can change this default behavior if you want to retain the flag value in the imported requisition irrespective of the value in the supplier data.

Prerequisites

Your administrator must enable the following parameters:

- **Generate invoices for evaluated receipt settlement** (`Application.Procure.EnableERS`). For information about the parameter, see Generate invoices for evaluated receipt settlement.
- **Enable amendments for long-standing orders** (`Application.Purchasing.AllowSimultaneousAmendmentsOnRequisitions`). For information about the parameter, see Enable amendments for long-standing orders.
- **Prevent defaulting ERS value from suppliers** (`Application.Procure.PreventDefaultingERSValueFromSuppliers`). For information about the parameter, see Prevent defaulting ERS value from suppliers.

Restrictions

When you import a changed purchase order or an amendment to it, if the ERS flag value differs from the existing version in the buying solution, the import fails with an error message. You must ensure that the imported value matches the value in the existing version.

> **Note**
>
> When you import a requisition with an ERS flag value that differs from the value in the associated contract, the value in the imported requisition is retained.

Considerations for Importing Requisitions

Importing requisition data is a complex operation that requires your careful attention.

When importing requisition information, you should be aware of the following considerations:

- The status of an incoming requisition depends on whether the import is successful as follows:
  - If the requisition has no errors, it has the status **Submitted**.
  - If the requisition is incomplete or incorrect, it has the status **Composing**.
- If errors occur when a requisition is imported, the buying solution sends an email notification message to the preparer. The preparer must change the requisition manually and resubmit the document for approval.
- An imported requisition line item is treated as a customer catalog item if the Supplier, Supplier Part Number, and Unit of Measure field values form an exact match with an item in the customer catalog. The Supplier Part Auxiliary ID field value, if available, is also used for matching purposes.
  If a price is specified on the imported line item, it is used. Otherwise, the price in the customer catalog item description is used. All other imported requisition line items are treated as non-catalog items. All suppliers must be pre-registered.
- For sites configured for budget checks in external ERP system:
  - If an imported requisition has funds allocated in the ERP system and also includes details for the `ERP Reference Number` and `ERPLineItemNumber` fields, the requisition does not undergo budget checks in the ERP system when it is imported to the buying solution. However, if the associated order for the requisition is changed, the requisition undergoes validations for budgetary attributes in the ERP system.
  - If any imported requisitions are stuck in the **Submitting** state due to the pending sync call of the real time budget check, run the **Reschedule Requisition Submit For Imported Requisitions** scheduled task. Requisitions in the **Submitting** state cannot be closed, edited, or copied.
- (For SAP-integrated sites configured for Funds Management accounting functionality) The buying solution does not add Funds Management account assignment fields on requisitions that are imported from external systems. Ensure that you add the appropriate Funds Management account assignment fields on the requisitions before importing the requisitions to the buying solution. For information about the format and contents of the Funds Management account assignment field CSV files, see the Data Dictionary.
- If an imported requisition is associated with a contract, and a matching contract is found in the buying solution, the requisition is created with the matching contract.
  If multiple contracts are found matching in the buying solution, the contract with the best (lowest) price match is selected and applied to the requisition.
  If a matching contract is not found in the buying solution, the requisition is created with the best matching price from the customer catalog.
- If the `Supplier` field value has not been specified in the `RequisitionDetail.csv` file during import, the buying solution replaces the `Supplier Location` field value with a null value (though the `Supplier Location` value is specified).

  > **Note**
  >
  > This behavior is true even when the **User Set** feature has been enabled.

- Requisitions that have no associated line items in `RequisitonDetail.csv` file have the status **Composing**. Line items that do not have an associated requisition in `RequisitionHeader.csv` are ignored.
- The time when you import a requisition is considered the requisition creation date.
- In the user interface, the **History** tab indicates when the requisition was imported.

- The requisition import feature is designed only to create requisitions, not to complete update actions to existing requisitions, nor to delete existing requisitions.
- When running the **Import Requisitions** data import task in Ariba Administrator, you must select the Load data import operation. Selecting any other data import operation will result in inconsistent system behavior.
- As a best practice, we recommend you to ensure requisition files that you are importing have complete accounting data.
  Your site can be configured to automatically populate accounting fields that are blank in imported requisitions with the default values determined by the buying solution. To configure this behavior, contact SAP Ariba Customer Support.

  > ### Note
  >
  > When updating accounting information by exporting and importing Excel spreadsheets, the original field values prior to the export (even if they were blank) are retained in the buying solution if the associated values are blank in your import file.
  >
  > However, if you add new split lines in the spreadsheet, then the accounting information for the blank fields in these lines are defaulted from the first line. If the corresponding fields values in the first line are also blank, then the blank fields in the new split lines are populated with the values determined by the account defaulting mechanism.

- Accounting information is defaulted based on the commodity code mappings specified in `ERPCommodityCodeMap.csv`. If a commodity code does not appear in `ERPCommodityCodeMap.csv`, accounting information is defaulted based on the user profile of the user specified in the `Requester` or `Preparer` field in `RequisitionHeader.csv`.
- If the imported `RequisitionDetail.csv` file includes account type data, the buying solution validates it to ensure that the imported account type is valid for the corresponding commodity code.

  > ### Note
  >
  > If the requisitions have split accountings, the accounting information does not get defaulted when you import such requisition files. This is because corresponding `RequisitionSplitAccounting.csv` files already have some accounting data in them and this data becomes the accounting information for the imported requisition files.
  >
  > If the imported requisitions and the corresponding `RequisitionSplitAccounting.csv` do not have any accounting data in them, then the buying solution uses the accounting information available in the User Profile of the signed-in user.

- You can ask SAP Ariba Customer Support to enable auto-receiving for imported requisitions. (By default, auto-receiving is disabled for imported requisitions). Auto-receiving sets the number of days after which an item will be automatically received. One day is added to the lead time for an item (as specified in the product description). For example, if the lead time for an item is specified as three days, then auto-receiving occurs approximately four days after the item is fully ordered (the lead time plus one day). Non-catalog items have a lead time of zero days. When auto-receiving is enabled for imported requisitions, it is effective regardless of the receiving rules specified for your site.
- If your business requires the imported requisitions with kits to be validated against budgets, contact SAP Ariba Customer Support to have this option configured.
- Your site can be configured to replace the null value specified for requisition preparer with the specified requestor value, when an imported requisition is updated using import requisition tasks. Contact SAP Ariba Support to enable the `Application.Purchasing.DefaultNullPreparerWithRequestorForImportRequisition` parameter.

Defaulting of Line Item Price When Buyer Part Number and Facility Do Not Match

When an imported requisition does not have a matching `Buyer Part Number` or `Facility` value, then the requisition line item price is defaulted in different ways depending on your method of import.

Defaulting of Line Item Price When Importing Using cXML

If the requisition line has a matching `SupplierPartID` and `SupplierID` value, but the line item price (`UnitPrice` value) is zero (0) or null, then the item price is defaulted from the matching customer catalog item.

Defaulting of Line Item Price When Importing Using CSV or Web Services

If the requisition line has a matching `Supplier` and `Supplier_Location` value, but the line item price (UnitPrice value) is null, then the item price is defaulted from the matching customer catalog item.

If the requisition line has a matching `Supplier` and `Supplier_Location` value, but the line item price (`UnitPrice` value) is zero (0), then this value is retained and the requisition is kept in the **Composing** status. You must manually change this value and submit the requisition.

Restrictions on Requisition Import

There are a number of limitations on requisition import about which you should be aware.

Your buying solution does not support the import of the **Delay Purchase Until** field.

You can import one comment each at the header and the line item level. That is, if you are importing comments at the line item level, the same comment is imported for all line items of the kit. This behavior is also applicable to all other line item fields, for example, **Split Accounting**, and **Quantity**.

Two fields have been introduced in the CSV files to facilitate the import of comments:

- `Comment`: The comment string.
- `ExternalComment`: A Boolean value that determines whether or not this comment is available for export to an external ERP.

If you want the comments to be imported at the requisition header level, add these two fields to the `RequisitionHeader.csv` file. If the comments are to be imported at the kit line item level, add the fields to the `RequisitionDetail.csv` file.

All other fields in a requisition that a user can change or edit (the same way if the user was manually creating a requisition from the user interface on the buying solution) are available for import.

> **Note**
>
> When importing requisitions with kits, the following fields do not get imported into the buying solution from the CSV files. These field values get populated in the user interface based on the existing defaulting mechanism.

- `CommodityCode`
- `PartitionedCommodityCode`
- `Price`
- `Description`
- `UOM`

Replacing user-defined prices of customer catalog line items in imported requisitions with customer catalog prices in SAP Ariba solutions is not supported for the import of the following:

- Service items. This feature applies only to material items.
- SAP Fieldglass requisitions

Related Information

About Importing Requisitions Using Various Integration Channels [page 68]

Restrictions When SupplierPartID and SupplierAuxPartID Combinations Are the Same

If there are multiple customer catalog items (individual items as well as kit items) in a requisition that share the same `SupplierPartID` and `SupplierAuxPartID` combinations, then the buying solution cannot make a distinction among those entries.

The `SupplierPartID` and `SupplierAuxPartID` combination must always be unique. If they are not, then the buying solution either imports only one of items, or imports the same line item multiple times.

Restrictions When Requisition Lines Are Split by Amount

If an imported requisition includes lines that are split by amount, and a matching entry is found in the item master that has a better price match, then the buying solution replaces the existing line amount with that of the matching entry in the item master.

When requisition lines are split by amount, the split lines are not updated, which can result in a mismatch of line item amounts and the total split line amount.

For example, if a requisition line (amounting to 100.00 USD) contains the following splits by amount:

- Split 1: 30.00 USD
- Split 2: 30.00 USD

- Split 3: 40.00 USD

The buying solution performs checks against the item master to identify the best possible match for the imported requisition item. During this check, if a best price match of 90.00 USD is found for the requisition line item, the original line item amount of 100.00 USD is replaced with 90.00 USD and the requisition is created.

In this example, line item splits by amount are not considered during the best price match, resulting in an amount mismatch where the line item amount is 90.00 USD and the split lines total amount is still 100.00 USD.

Requisitions where lines are split by amount remain in the **Composing** state until the amount mismatch is resolved manually and the split line amounts are corrected.

Import of Amendments

An amendment is an approvable document in which buyers add ordered items from a requisition and manage the change process for those items independently of the main requisition or other amendment documents which are in process for other lines on the same requisition. For more information about creating and managing amendments, see Purchasing Guide for Procurement Professionals.

You can import amendments based on imported requisitions and process change orders in the buying solution. Using imported amendments, you can add new line items, edit header and line-level fields, and delete line items from the associated requisition. You can also cancel an amendment that is in the **Composing** state.

The **Import Requisition(s)** web services task for importing requisitions can be used to import amendments.

The status of an imported amendment depends on whether the import is successful as follows:

- If the amendment has no errors, it has the status **Submitted**.
- If the amendment is incomplete or incorrect, it has the status **Composing**.

Similar to amendments submitted in the buying solution, imported amendments in the **Submitted** state undergo the approval process. After an amendment is fully approved, the changes in the amendment are merged with the latest version of the requisition and change orders are sent to suppliers.

When importing amendments, you should be aware of the following considerations:

- If an imported amendment is associated with a contract, and a matching contract is found in the buying solution, the amendment is created with the matching contract.
- The time when you import an amendment is considered the amendment creation date.
- In the user interface, the **History** tab indicates when the amendment was imported.

Related Information

Prerequisites for Importing Amendments [page 104]
Import of Amendments Using Web Services [page 104]
Error Messages for Failed Amendment Import [page 108]

Prerequisites for Importing Amendments

- Your site must be enabled to allow users to create amendments based on ordered requisitions.
  To enable this feature, have your Designated Support Contact (DSC) submit a case. SAP Support will follow up to complete the case.
- Your site must be configured to use web services for data integration.
- Customer administrators must enable the following self-service parameter to allow users to create amendments for header-level fields on requisitions:
  `Application.Purchasing.EditAmendmentHeader`
- Before amendments are imported, the associated requisitions must be imported from the external ERP system.
- In the default configuration, amendments cannot be imported if the associated purchase orders are later than the **Ordered** state (for example, **Receiving**, **Received**, or **Invoiced** states). To allow imported amendments for purchase orders later than the **Ordered** state, have your Designated Support Contact log a service request. An SAP Ariba Customer Support representative will follow up to complete the request.
- (Optional) If error codes for failed amendment import must be included in the web services response, contact SAP Ariba Support for help with the configuration.

> **Note**
>
> Amendments can be imported only by using web services. Other data import methods, such as CSV file channel and cXML posts are not supported.

Import of Amendments Using Web Services

You can import amendments using the **Import Requisition(s)** web services task.

Header-Level Information for the Import Requisition(s) Web Services Task

The following table lists the fields or attributes at the header-level of the XML request for the **Import Requisition(s)** web services task and also indicates whether they are mandatory for the request to be sent successfully to the external system:

Field/Attribute name	Description	Mandatory/Not mandatory
`operation`	The existing `operation` attribute at the header level of the XML request used by this task now supports a new value named `Amend`. This value specifies that the document type is an amendment. The `operation` attribute also supports the `Cancel` value. This operation	Mandatory

Field/Attribute name	Description	Mandatory/Not mandatory
	allows you to cancel previously imported amendments. Imported amendments can be canceled before they are fully approved and changes in those amendments are merged with the corresponding requisition.	
`OriginatingSystemReferenceID`	Refers to the ID assigned to the associated requisition in the external system.	Mandatory
`OriginatingSystemAmendmentID`	Refers to the ID assigned to the amendment document in the external system.	Mandatory

Line-Level Information for the Import Requisition(s) Web Services Task

The XML request for the **Import Requisition(s)** web services task includes the following fields or attributes at the line level to specify the details for the line items on an amendment:

- `operation` attribute: The XML request includes the `operation` attribute for each line item on the amendment. This attribute is mandatory at the line level and accepts one of the following values to indicate the type of operation for the line item:
  - `New`: Specifies a new line item that must be added to the associated requisition.
    Example:

    ```
    <urn:NumberInCollection>1</urn:NumberInCollection>
                        <urn:OriginatingSystemLineNumber>1</
    urn:OriginatingSystemLineNumber>
                  <urn:Operation>New</urn:Operation>
    ```

    If no value is specified for this attribute for a newly added line item, the buying solution considers the operation as `New`.
  - `Delete`
    Example:

    ```
    <urn:NumberInCollection>1</urn:NumberInCollection>
                        <urn:OriginatingSystemLineNumber>1</
    urn:OriginatingSystemLineNumber>
                  <urn:Operation>Delete</urn:Operation>
    : Specifies that the line item must be
                                        deleted from the associated requisition.
    ```
  - `Update`: Specifies that values for one or more fields for the line item have been updated.
    Example:

    ```
    <urn:NumberInCollection>1</urn:NumberInCollection>
                        <urn:OriginatingSystemLineNumber>1</
    urn:OriginatingSystemLineNumber>
                  <urn:Operation>Update</urn:Operation>
    ```

If no value is specified for the `operation` attribute for line item on the corresponding requisition, the buying solution considers the operation as `Update`.

- `OriginatingSystemAmendmentID`: Refers to the ID assigned to the amendment document in the external system.
- `OriginatingSystemReferenceID`: Refers to the ID assigned to the associated requisition in the external system. For example, ST018.
- `UniqueName`: Refers to the unique ID of the amendment document. For example, PR45-A1.
- `OriginatingSystemLineNumber`: Refers to the line number assigned to an item on an amendment undergoing the amendment process. This attribute is mandatory if the value for the `operation` attribute at the line level is `New`, `Delete`, or `Update`.
- `NumberInCollection`: Specifies the line number of the item on the associated requisition for which the amendment is being imported. This attribute is mandatory if the value for the `operation` attribute at the line level is `Update`.

Sample Web Service Request

The following is a sample XML request for an amendment imported using the **Import Requisition(s)** web services task:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:Ariba:Buyer:vrealm_1">
   <soapenv:Header>
      <urn:Headers>
         <!--You may enter the following 2 items in any order-->
         <!--Optional:-->
         <urn:variant>vrealm_1</urn:variant>
         <!--Optional:-->
         <urn:partition>vrealm_1</urn:partition>
      </urn:Headers>
   </soapenv:Header>
   <soapenv:Body>
      <urn:RequisitionImportPullRequest partition="prealm_1" variant="vrealm_1">
         <urn:Requisition_RequisitionImportPull_Item>
            <urn:item>
               <urn:DefaultLineItem>
                  <urn:DeliverTo>Jack</urn:DeliverTo>
                  <urn:NeedBy>2019-08-21T09:30:47Z</urn:NeedBy>
               </urn:DefaultLineItem>
               <urn:ImportedHeaderCommentStaging>HeaderlevelComment</
urn:ImportedHeaderCommentStaging>
               <urn:ImportedHeaderExternalCommentStaging>true</
urn:ImportedHeaderExternalCommentStaging>
               <urn:LineItems>
                  <urn:item>
                     <urn:BillingAddress>
                        <urn:UniqueName>3001</urn:UniqueName>
                     </urn:BillingAddress>
                     <urn:CommodityCode>
                        <urn:UniqueName>05007</urn:UniqueName>
                     </urn:CommodityCode>
                     <urn:Description>
                        <urn:CommonCommodityCode>
                           <urn:Domain>unspsc</urn:Domain>
                           <urn:UniqueName>3210</urn:UniqueName>
                        </urn:CommonCommodityCode>
                        <urn:Description>SG Req 1 Item 1</urn:Description>
                        <urn:ManPartNumber>Supplier-PartId-Test1</urn:ManPartNumber>
```

```
                        <urn:Price>
                           <urn:Amount>300</urn:Amount>
                           <urn:Currency>
                              <urn:UniqueName>USD</urn:UniqueName>
                           </urn:Currency>
                        </urn:Price>
                        <urn:SupplierPartAuxiliaryID/>
                        <urn:SupplierPartNumber/>
                        <urn:UnitOfMeasure>
                           <urn:UniqueName>EA</urn:UniqueName>
                        </urn:UnitOfMeasure>
                     </urn:Description>
                     <urn:ImportedAccountTypeStaging>Expense</
urn:ImportedAccountTypeStaging>
                     <urn:ImportedAccountingsStaging>
                        <urn:SplitAccountings>
                           <urn:item>
                              <urn:Account>
                                 <urn:UniqueName>7752</urn:UniqueName>
                              </urn:Account>
                              <urn:BusinessUnit>
                                 <urn:UniqueName>US002</urn:UniqueName>
                              </urn:BusinessUnit>
                              <urn:Company>
                                 <urn:UniqueName>02</urn:UniqueName>
                              </urn:Company>
                              <urn:CostCenter>
                                 <urn:UniqueName>4000</urn:UniqueName>
                              </urn:CostCenter>
                              <urn:NumberInCollection>1</urn:NumberInCollection>
                              <urn:Percentage>100.0000000000</urn:Percentage>
                              <urn:ProcurementUnit/>
                              <urn:Quantity>5</urn:Quantity>
                           </urn:item>
                        </urn:SplitAccountings>
                        <urn:Type>
                           <urn:UniqueName>_Percentage</urn:UniqueName>
                        </urn:Type>
                     </urn:ImportedAccountingsStaging>
                     <urn:ImportedDeliverToStaging>Jack</
urn:ImportedDeliverToStaging>
                     <urn:ImportedLineCommentStaging>LineComment</
urn:ImportedLineCommentStaging>
                     <urn:NumberInCollection>1</urn:NumberInCollection>
                     <urn:OriginatingSystemLineNumber>1</
urn:OriginatingSystemLineNumber>
                     <urn:Operation>Update</urn:Operation>
                     <urn:Quantity>5</urn:Quantity>
                     <urn:ShipTo>
                        <urn:UniqueName>3000</urn:UniqueName>
                     </urn:ShipTo>
                     <urn:Supplier>
                        <urn:UniqueName>sid496<OriginatingSystem>SG</urn:>
                     </urn:Supplier>
                     <urn:SupplierLocation>
                        <urn:ContactID>14</urn:ContactID>
                        <urn:UniqueName>14</urn:UniqueName>
                     </urn:SupplierLocation>
                  </urn:item>
               </urn:LineItems>
               <urn:Name>FK_PR_WS_IMP</urn:Name>
               <urn:OriginatingSystem>
               <urn:OriginatingSystemReferenceID>ST018
               </urn:OriginatingSystemReferenceID>
               <urn:Requester>
                  <urn:PasswordAdapter>PasswordAdapter1</urn:PasswordAdapter>
                  <urn:UniqueName>cnoll</urn:UniqueName>
               </urn:Requester>
```

```
               <urn:OriginatingSystemAmendmentID>ST018-v1</
urn:OriginatingSystemAmendmentID>
               <urn:UniqueName></urn:UniqueName>
               <urn:Operation>Amend</urn:Operation>
            </urn:item>
         </urn:Requisition_RequisitionImportPull_Item>
      </urn:RequisitionImportPullRequest>
   </soapenv:Body>
</soapenv:Envelope>
```

The following is a sample XML response for an imported amendment that is successfully submitted in the buying solution:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance">
   <soap:Header>
      <Headers xmlns="urn:Ariba:Buyer:vrealm_1">
         <variant>vrealm_1</variant>
         <partition>prealm_1</partition>
      </Headers>
   </soap:Header>
   <soap:Body>
      <RequisitionImportPullReply variant="vrealm_1" partition="prealm_1"
xmlns="urn:Ariba:Buyer:vrealm_1">
         <Requisition_RequisitionIdExport_Item>
            <item>
               <OriginatingSystem>SG</OriginatingSystem>
               <OriginatingSystemReferenceID>ST018</OriginatingSystemReferenceID>
               <urn:OriginatingSystemAmendmentID>ST018-v1</
urn:OriginatingSystemAmendmentID>
               <StatusString>Submitted</StatusString>
               <UniqueName>PR10-A1</UniqueName>
            </item>
         </Requisition_RequisitionIdExport_Item>
      </RequisitionImportPullReply>
   </soap:Body>
</soap:Envelope>
```

The following attributes are mandatory in the XML response:

- `OriginatingSystem`: Specifies the ID of the external system from which the amendment is imported.
- `OriginatingSystemReferenceID`: Refers to the ID assigned to the associated requisition in the external system.
- `OriginatingSystemAmendmentID`: Refers to the ID assigned to the amendment in the external system.
- `UniqueName`: Refers to the unique ID of the amendment document in the buying solution. For example, PR45-A1.
- `StatusString`: Specifies the status of the amendment. If an amendment import is successful, the status is **Submitted**.

Error Messages for Failed Amendment Import

The following table lists the error messages that are displayed in the **Import Requisitions(s)** web services response when an amendment fails to be imported.

Error code	Description
2026	Simultaneous amendments feature is not enabled for your site. Ensure that Simultaneous Amendment and Import Simultaneous Web Service features are enabled.
2031	Could not process the amend operation because the site configuration parameter to allow creation of amendments for header-level fields is not enabled.
2032	Values for certain header-level fields, such as Purchasing Unit or Business Unit should not be modified when amendments are imported.
2033	Amendments with missing line items cannot be imported. Include at least one line item for which an amendment is to be created.
2034	Amendment `{0}` with `OriginatingSystemReferenceID {1}` and `OriginatingSystemAmendmentID {2}` could not be canceled in the SAP Ariba solution due to unexpected errors. `OriginatingSystemReferenceID` refers to the ID of the requisition associated with the amendment. `OriginatingSystemAmendmentID {2}` refers to the ID of the amendment in the external system.
2035	Value for `{0}` is missing. `{0}` is the placeholder for a mandatory field (for example, amendment ID) on an imported amendment.
2036	Could not find the requisition matching the ID `{0}`. `{0}` is the placeholder for the associated requisition for which the amendment is being imported.
2037	Could not cancel amendment `{0}` since it is in Merged state. `{0}` is the placeholder for the amendment that is being imported.
2038	Could not cancel amendment `{0}` since it is in the submitted state. `{0}` is the placeholder for the associated requisition for which the amendment is being imported.

APIs to Access Submit Failure Records for Imported Requisitions

When users import requisitions from external systems into the buying solutions, they can access the errors resulting from validations of standard and custom fields through APIs such as the Operational Reporting for Procurement API.

The errors could be at the line item level or at the header level. Users can view and fix the errors, and then import the requisitions again.

> **Note**
>
> - If the total number of characters in the error messages for a requisition exceeds 50000, then, the records retrieval might take longer.
> - The error messages can be retrieved for requisitions that are imported after the **Use APIs to access submit failure records for imported requisitions** (`Application.Procure.UseAPIsToAccessSubmitFailureRecordsForImportedRequisitions`) parameter is enabled. The errors that occurred during previous requisition imports can't be retrieved. For example, if requisition PR123 was imported with errors before the parameter was enabled, the messages for those validation errors can't be retrieved. If the PR123 is updated and imported again after the parameter is enabled, and validation errors occur, then, the error messages related to the second import can be retrieved.

To retrieve the data using the Operational Reporting for Procurement API, users must first create a reporting view template that specifies the reporting fields to be included in the API response and the selection filters to select the relevant records for inclusion.

- In the list of reporting fields, users must include the `SubmitEventDetails` field that will contain the error messages for a requisition.
- In the filter parameters, users can use the `createdDateFrom`, the `createdDateTo`, and the `status` parameters. The `status` parameter must be set to `Composing`. If you set the `status` parameter to any other value, the `SubmitEventDetails` field returns an empty value.

Prerequisites

- Your administrator must register your organization to use the SAP Ariba Developer Portal. See Registering Your Organization to Use the SAP Ariba Developer Portal.
- Your administrator must enable the **Use APIs to access submit failure records for imported requisitions** (`Application.Procure.UseAPIsToAccessSubmitFailureRecordsForImportedRequisitions`) parameter. See Use APIs to access submit failure records for imported requisitions.

Sample Request to Create a Custom View Template

```
{
```

```
    "viewTemplateName": "RequistionSupplierTemplate_SubmitFailure",
    "type": "custom",
    "status": "published",
    "documentType": "Requisition",
    "selectAttributes": [
        "UniqueName",
        "Name",
        "Preparer",
        "Requester",
        "OriginatingSystem",
        "OriginatingSystemReferenceID",
        "CompanyCode",
        "SubmitEventDetails"
    ],
    "filterExpressions": [
        {
            "name": "createdDateTo",
            "field": "TimeCreated",
            "op": "<=",
            "defaultValue": "2022-05-10T00:00:00Z"
        },
        {
            "name": "createdDateFrom",
            "field": "TimeCreated",
            "op": ">",
            "defaultValue": "2022-05-09T00:00:00Z"
        },
        {
            "name": "StatusInList",
            "field": "StatusString",
            "op": "IN",
            "defaultValue": [
                "Composing"
            ]
        }
    ]
}
```

Sample Response

```
{
"Records": [
{
    "Requester":
    {
        "UniqueName": "cnoll",
        "PasswordAdapter": "PasswordAdapter1"
    },
    "OriginatingSystem": "ERP1651805054801",
    "UniqueName": "PR776",
    "OriginatingSystemReferenceID": "ERP1651805054801",
    "Preparer":
    {
        "UniqueName": "cnoll",
        "PasswordAdapter": "PasswordAdapter1"
    },
    "Name": null,
    "SubmitEventDetails": [
    {
        "EventMessage": ": This requisition has been retained in the
composing status because some mandatory field values are missing. Specify
appropriate values in the missing fields and submit the requisition.\n\nHere
is the error information:\n\nRequisition :PR776 ([ariba.purchasing.core.Requisition
```

```
[BaseId 23519500991 AAGQASgV53xq! asywj3z.1qfhk0]])\n\nLineItem # \tError Fields
\n\nName\t\t.\n",
        "UniqueName": "PR776",
        "EventType": "RequisitionImportSubmitFailureRecord"
    }]
}]
```


Related Information

Operational Reporting API for Procurement

Rules for Demand Aggregation

Demand aggregation enables buyer organizations to consolidate line items from different purchase requisitions into aggregated requisitions.

You can configure the following types of rules to manage the process of demand aggregation:

- Demand holding rules: These are used to identify requisition line items to be held for aggregation and auto-aggregation.
- Assignment rules: These are used to assign the held line items to purchasing agents.
- Aggregation rules: These are used to group the held line items marked for auto-aggregation. A scheduled task converts these groups into aggregated requisitions.

Demand Holding Rules [page 113]

Assignment Rules for Demand Aggregation [page 114]

Demand Aggregation Rules [page 115]

Import Demand Holding Rules Task [page 116]

Import Assignment Rules for Demand Aggregation Task [page 118]

Import Demand Aggregation Rules Task [page 121]

Running the Auto-Aggregate Held Line Items Scheduled Task [page 123]

Modifying Demand Holding, Assignment, and Aggregation Rules [page 124]

Deactivating Demand Holding, Assignment, and Aggregation Rules [page 125]

Demand Aggregation for MRP-run Requisitions in Sites Enabled for Demand Aggregation and External Budget Checking [page 125]

Demand Holding Rules

Demand holding rules are defined in a CSV file by specifying values for fields, such as commodity code, supplier, threshold amount, and item type (catalog or non-catalog). This file is then imported using the **Import Demand Holding Rules** task. Each rule can use one or more fields and the rules file can have multiple rules.

Demand holding rules are used to identify requisition line items to be held for aggregation and for auto-aggregation. These rules are invoked when a requisition is submitted and when the requisition is fully approved.

The buying solution compares a line item's fields' values with those defined for the holding rules and holds the line for aggregation if it meets the conditions specified.

Example

Field	Value
CommodityCodeDomain	unspsc
CommodityCodeValue	432118
HasContract	No
LineType	Customer catalog
LineLevelThreshold	200
LineLevelThresholdCurrency	USD

If you configure a demand holding rule using the details specified in the table in this example, a line item will be held for aggregation if it fulfills these conditions:

- the item is a customer catalog item with commodity code 432118
- the item does not have any contracts associated with it
- the item's amount is 200USD or more

Importing rules for demand aggregation

Members of the **Customer Administrator** group can import rules for demand aggregation using the **Site Manager** workspace in Ariba Administrator.

Assignment Rules for Demand Aggregation

Assignment rules are defined in a CSV file by specifying values for fields such as company code, purchasing unit, and commodity code. This file is then imported using the **Import Assignment Rules for Demand Aggregation** task. Each rule can use one or more fields, and the rules file can have multiple rules.

Assignment rules are used to assign held line items to purchasing agents. Assignment rules are invoked when a requisition is submitted and when it gets fully approved.

Your buying solution compares the line item's fields' values with those defined for the rules and assigns the line item to the specified purchasing agent if the values match.

For example, you can assign all held line items for IT Equipment from a specific purchasing unit to a purchasing agent, by specifying the values of commodity code and purchasing unit in the CSV file.

Example

Field	Value
CommodityCodeDomain	unspsc
CommodityCodeValue	432118
HasContract	No
LineType	Customer catalog
PurchasingUnit	US002
AssignTo	hlong

If you configure an assignment rule like the one shown in the table in this example, a line item will be assigned to the user `hlong` for aggregation, if it is a customer catalog item with commodity code 432118, it does not have any contracts associated with it, and if it belongs to the `US002` purchasing unit.

Demand Aggregation Rules

Demand aggregation rules are defined in a CSV file by selecting the fields based on which you want to group the held line items. Specify “Yes” or “No” under the fields in the rules CSV file to select or deselect them.

Demand aggregation rules are used to group held line items marked for auto-aggregation. A scheduled task creates and submits aggregated requisitions for these groups.

You can define only one aggregation rule for a purchasing unit, company code, or busines unit. The rules file can have multiple rules.

Held line items marked for auto-aggregation are grouped first based on their purchasing unit, then based on their assigned purchasing agent and then based on the fields selected in the aggregation rule.

Example

Field	Value
PurchasingUnit	IN001
Supplier	Yes
ShipTo	Yes

In this example, the aggregation rule for purchasing unit IN001 has the values for supplier and ship-to location fields as “Yes”. In this case, the held line items from IN001 that are marked for auto-aggregation are first grouped

based on their assigned purchasing agents and then based on their supplier and ship-to locations. That is, two held line items are added to the same aggregated requisition only if their assigned purchasing agent, supplier, and ship-to location are same.

Import Demand Holding Rules **Task**

The **Import Demand Holding Rules** task enables you to load, update, and deactivate demand holding rules. These rules are used to identify requisition line items to be held for aggregation and auto-aggregation.

Demand holding rules are defined in a CSV file by specifying values for fields such as commodity code, supplier, threshold amount, and type of line item (customer catalog or non-catalog).

For sites integrated with SAP or PeopleSoft ERP systems, to hold requisition line items based on their company codes or business units, you can do either of the following:

- Specify the purchasing unit in the `PurchasingUnit` field and also define the mapping between purchasing units and company codes or business units.
- Configure an extensible field to the company code or business unit field and then specify the company code or business unit in the configured field.

Import Demand Holding Rules: **sample file**

The following table provides descriptions and sample values for fields in the `DemandHoldingRule.csv` file used to import demand holding rules:

Field Name	Required	Description	Sample Value
`UniqueName`	Yes	String. The user-specified name of a rule. Each rule must have a unique name.	DHR001
`DefinitionLine`	No	Boolean. Specifies whether a row of values forms a definition line. This is used to add custom fields to the CSV file. A rules file must have only one definition line. It is recommended to have the definition line immediately after the header line of the CSV file.	Yes/No

Field Name	Required	Description	Sample Value
`CommodityCodeDomain`	No	String. The domain of the commodity code. For example, 'unspsc' denotes United Nations Standard Products and Services Code.	unspsc
`CommodityCodeValue`	No	String. The commodity code of the line item	432118
`Ranking`	Yes	Integer. This is used to set the priority of the rules defined. The rule with the lowest ranking has the highest priority.	1
`HasContract`	No	Boolean. Specifies whether a line item has a contract associated with it	Yes/No
`LineType`	No	String. The line type of the line item. The value of this field can be`Catalog` or `Non Catalog`.	Customer catalog
`Supplier`	No	String. The supplier ID of the supplier of the line item	11
`PurchasingUnit`	No	String. The purchasing unit of the line item. For sites integrated with SAP or PeopleSoft ERP systems, for this field to work, purchasing units must be mapped to company codes or business units respectively.	US003
`AutoAggregation`	No	Boolean. Specifies whether the line item should be marked for auto-aggregation	Yes/No
`Description`	No	String. The description of the holding rule	Demand holding rule for ITEquipment for US003 purchasing unit
`LineLevelThreshold`	No	BigDecimal. Line items having amount equal to or more than this value are held for aggregation.	100
`LineLevelThresholdCurrency`	No	String. The currency of the threshold value	USD

Field Name	Required	Description	Sample Value
`ConditionValue`	No	String. These fields can be used to add additional conditions to the demand holding rules. To use these fields, specify the required dotted field paths as the field values in the definition line, and specify the value of the field in the rule. For example, specify `Requisition.LineItemsCount` in the definition line for ConditionValue1, and specify 1 as the field value in the rule. You can define up to 10 custom fields for each rule.	10

Import Assignment Rules for Demand Aggregation Task

The **Import Assignment Rules for Demand Aggregation** task enables you to load, update, and deactivate assignment rules for demand aggregation. These rules are used to assign held line items to purchasing agents.

Assignment rules for demand aggregation are defined in a CSV file by specifying values for fields such as company code, purchasing unit, and commodity code.

For sites integrated with SAP or PeopleSoft ERP systems, to assign requisition line items based on their company codes or business units, you can do either of the following:

- Specify the purchasing unit in the `PurchasingUnit` field and also define the mapping between purchasing units and company codes or business units.
- Configure an extensible field to the company code or business unit field and then specify the company code or business unit in the configured field.

Import Assignment Rules for Demand Aggregation: **sample file**

The following table provides descriptions and sample values for fields in the `DemandAssignmentRule.csv` file used to import assignment rules for demand aggregation:

Field Name	Required	Description	Sample Value
`UniqueName`	Yes	String. The user-specified name of a rule. Each rule must have a unique name	DAR001
`DefinitionLine`	No	Boolean. Specifies whether a row of values forms a definition line. This is used to add custom fields to the CSV file. A rules file must have only one definition line. It is recommended to have the definition line immediately after the header line of the CSV file.	Yes/No
`CommodityCodeDomain`	No	String. The domain of the commodity code. For example, 'unspsc' denotes United Nations Standard Products and Services Code.	unspsc
`CommodityCodeValue`	No	String. The commodity code of the line item	432118
`Ranking`	Yes	Integer. This value is used to set the priority of the rules defined. The rule with the lowest ranking has the highest priority.	1
`HasContract`	No	Boolean. Specifies whether a line item has a contract associated with it.	Yes/No
`LineType`	No	String. Line type of the line item. The value of this field can be `Catalog` or `Non Catalog`.	Catalog
`Supplier`	No	String. The supplier ID of the line item	11

Field Name	Required	Description	Sample Value
`PurchasingUnit`	No	String. The purchasing unit of the line item. For sites integrated with SAP or PeopleSoft ERP systems, for this field to work, purchasing units must be mapped to company codes or business units respectively.	US003
`AssignTo`	No	String. The user ID of the purchasing agent to whom the held line item should be assigned if it satisfies the conditions of the rule	hlong
`AssignToPasswordAdapter`	No	String. The password adapter of the purchasing agent mentioned in the AssignTo field. The value specified here must match a value defined in System.PasswordAdapters.	PasswordAdapter1
`Description`	No	String. The description of the assignment rule	Assignment rule for ITEquipment for US003 purchasing unit
`ConditionValue`	No	String. You can use these fields to add upto 10 custom conditions for assignment rules. To use these fields, specify the required dotted field paths as the field values in the definition line, and specify the value of the field in the rule. For example, specify `Requisition.LineItemsCount` in the definition line for ConditionValue1, and specify 1 as the field value in the rule.	10

Import Demand Aggregation Rules **Task**

The **Import Demand Aggregation Rules** task enables you to load, update, and deactivate demand aggregation rules. These rules are used to group held line items marked for auto-aggregation.

Demand aggregation rules are defined in a CSV file by selecting the fields based on which you want to group the held line items. You need to specify "Yes" or "No" as the field values to select or deselect the fields. You can define only one aggregation rule for a purchasing unit, company code, or business unit.

Import Demand Aggregation Rules**: sample file**

The following table provides descriptions and sample values for fields in the `DemandAggregationRule.csv` file used to import demand aggregation rules:

Field Name	Required	Description	Sample Value
`UniqueName`	Yes	String. The user-specified name of a rule. Each rule must have a unique name.	DAggR001
`DefinitionLine`	No	Boolean. Specifies whether a row of values forms a definition line. This is used to add custom fields to the CSV file. A rules file must have only one definition line. It is recommended to have the definition line immediately after the header line of the CSV file.	Yes/No
Generic ERP system:`PurchasingUnit` SAP ERP system: `CompanyCode` PeopleSoft ERP system: `BusinessUnit`	No	String. The purchasing unit, company code, or business unit of the line item. The field name varies based on the ERP system used. One purchasing unit, company code, or business unit can have only one aggregation rule.	US003
`Description`	No	String. The description of the demand aggregation rule	Aggregation rule for held line items in US003

Field Name	Required	Description	Sample Value
`ShipTo`	No	Boolean. Specifies whether the held line items should be grouped based on their shipping address	Yes/No
`DeliverTo`	No	Boolean. Specifies whether the held line items should be grouped based on their delivery location	Yes/No
`Supplier`	No	Boolean. Specifies whether the held line items should be grouped based on their supplier	Yes/No
`SupplierLocation`	No	Boolean. Specifies whether the held line items should be grouped based on their supplier location	Yes/No
`ConditionValue`	No	Boolean. These fields can be used to add additional grouping criteria to the rules. To use these fields, specify the dotted field path in the definition line. Specify whether you want to use the field as a grouping criterion, by specifying **`Yes`** or **`No`** as the field value in the rule. For example, to group the held line items based on their supplier's location, specify `SupplierLocation.Country.Name` in the definition line for ConditionValue1, and specify **`Yes`** as the field value in the rule. You can define up to 2 custom fields	`SupplierLocation.Country.Name`

Running the Auto-Aggregate Held Line Items Scheduled Task

The **Auto-Aggregate Held Line Items** scheduled task automatically creates and submits aggregated requisitions for held line items marked for auto-aggregation.

Context

The scheduled task runs based on the schedule configured using the parameter `Application.Purchasing.AutoAggregateHeldLines.Schedule`. By default, the scheduled task runs every Monday at 2 a.m.

You can use the `Application.Purchasing.AutoAggregateHeldLines.ScheduledExtendedMinutes` parameter (contact SAP Ariba Support to configure) to set the duration in minutes to run the **Auto-Aggregate Held Line Items** scheduled task and create aggregated requisitions. For example, if the duration is set as 50, and the time in the `Application.Purchasing.AutoAggregateHeldLines.Schedule` parameter is set as 10 pm, the **Auto-Aggregate Held Line Items** scheduled task runs and creates aggregated requisitions for held line items from 10 pm to 10:50 pm. The scheduled task does not process the remaining held line items after 10:50 pm. If the parameter is set to -1, the scheduled task creates the aggregated requisitions anytime in the hour after the scheduled time. For example, between 10 pm to 10:59 pm. The default value of this parameter is 50.

> **Note**
>
> By default, the **Validate field values in auto-aggregated requisitions** (`Application.Purchasing.ValidateFieldValuesInAutoAggregatedRequisitions`) parameter is enabled. This ensures that the invalid field values are validated in auto-aggregated requisitions when the scheduled task runs. If field values are invalid, the requisition is not submitted and an error message is displayed. For more information, see Validate field values in auto-aggregated requisitions.

Procedure

1. On the dashboard, click **Manage** > **Core Administration**.
2. Click **Advanced**.
3. Select **Site Manager** > **Scheduled Tasks**.
4. Search for the **Auto-Aggregate Held Line Items** scheduled task using the search options.
5. Click **Run**.

Results

The **Auto-Aggregate Held Line Items** scheduled task creates and submits aggregated requisitions for held line items in the **Purchasing Workbench** that are marked for auto-aggregation.

Modifying Demand Holding, Assignment, and Aggregation Rules

You can modify demand holding, assignment, and aggregation rules using their unique name.

Procedure

1. On the **Data Import/Export** page in Ariba Administrator, on the **Export** tab, lookup the rule and click **Export** to export the rules to a CSV file.
2. Save the file on your computer's local drive.
3. Modify the rules as required. Ensure that you do not change the unique name of the rules.

   > **Note**
   >
   > To remove a condition from the `DemandHoldingRule.csv` or `DemandAssignmentRule.csv` files, specify `null` as the field value instead of keeping the field blank.

4. Save the file.
5. On the **Data Import/Export** page, on the **Import** tab, locate the import task for the rule and click **Import**.
6. Select the **Load** import operation and select the modified rules file.
7. Click **Run** to import the rules.

Results

The rules are updated based on their unique names. Errors, if any, are indicated by an exclamation mark displayed next to the status of the import task. You can click the status to view the error details.

Deactivating Demand Holding, Assignment, and Aggregation Rules

You can deactivate demand holding, assignment, and aggregation rules using their unique name.

Procedure

1. On the **Data Import/Export** page in Ariba Administrator, on the **Export** tab, click **Export** to export the rules file and save the file on your computer's local drive.
2. In the rules file, retain only the rules that you wish to deactivate and remove all other rules. Ensure that you do not change the unique names of the rules.
3. Save the file.
4. On the **Import** tab, locate the import task for the rule.
5. Click **Import**.
6. Select the **Deactivate** import operation and select the rules file that contains the rules to be deactivated.
7. Click **Run**.

Results

The rules are deactivated based on their unique names. Errors, if any, are indicated by an exclamation mark displayed next to the status of the import task. You can click the status to view error details.

Demand Aggregation for MRP-run Requisitions in Sites Enabled for Demand Aggregation and External Budget Checking

In the default setting in sites integrated with SAP ERP using SAP Integration Suite, managed gateway for spend management and SAP Business Network, if the sites are enabled for both demand aggregation and budget checking functionalities, line items that qualify for demand holding rules (held line items) are also sent to SAP ERP when users submit requisitions for approval. During the budget check process for a submitted requisition, the held line items are deleted from the corresponding requisition in SAP ERP. For MRP-run requisitions imported from SAP ERP, the MRP run process in SAP ERP triggers a cyclic update in which multiple requisitions are created for the same set of held line items that are deleted.

If enabled for your site, the held line items on MRP-run requisitions are deleted from SAP ERP only after an aggregated requisition is created in the SAP Ariba solution. This prevents creation of multiple requisitions during the MRP-run process. The SAP Ariba solution initiates budget checks on the aggregated requisition after the line items held for demand aggregation are deleted in SAP ERP. For help with this configuration, ask your Designated Support Contact to log a service request and an SAP Ariba Customer Support representative will contact you.

Prerequisites [page 126]

Restrictions [page 126]

Web Services Tasks [page 127]

Workflow for Managing Demand Aggregation and External Budget Checks for MRP-run Requisitions [page 128]

Prerequisites

- This feature is applicable for sites integrated with SAP ERP using SAP Integration Suite, managed gateway for spend management and SAP Business Network.
- To enable this feature, have your Designated Support Contact (DSC) submit a case. SAP Support will follow up to complete the case.
- Imported requisitions do not include tax and charge lines. Users must ensure that they add the appropriate taxes and charges before they submit the requisitions for approval.

Restrictions

- It is recommended that users should not make changes to requisitions in SAP ERP after those requisitions are imported in the SAP Ariba solution. In the case of MRP-run requisitions that are in the **Approved** or later states in the SAP Ariba solution, errors are returned when users attempt to make changes to the corresponding requisitions in SAP ERP.

  > ⚠ Caution
  >
  > Data entered on an updated requisition in SAP ERP might be lost after the associated requisition imported in the SAP Ariba solution is submitted for approval. This scenario can occur typically when a user updates the requisition in SAP ERP before the associated requisition imported in the SAP Ariba solution is processed. After a user submits the imported requisition for approval, the requisition undergoes budget checks in SAP ERP and upon successful completion of budget checks, this requisition overwrites the requisition in SAP ERP.

- When a change order process is initiated based on an MRP-run requisition in the buying solution, held line items might be deleted from the original requisition in SAP ERP when the preparer or requester edits or withdraws the requisition. In this scenario, duplicate requisitions might be created for the deleted held line items in SAP ERP during the next MRP-run process.

  > ⓘ Note
  >
  > It is recommended that preparers and requesters must not edit or withdraw an MRP-run requisition during the change order process.

- Line items marked as held line items in the SAP Ariba solution might continue to exist on the original requisitions in SAP ERP until an explicit deletion request is made to the ERP system. Possible causes for this scenario are:
  - An aggregated requisition has not been created.

- The customer catalog associated with the held line item has expired because of which the item could not be added to an aggregated requisition.

- For MRP-run requisitions that are in the **Approved** (or later) states in the SAP Ariba solution, changes cannot be made to the line items on the corresponding requisition in SAP ERP.
- This feature does not support requisitions consisting of service items.

Web Services Tasks

Import Requisitions Asynchronously

See Import Requisitions Asynchronously [page 87].

Export Response for Import Requisitions Task Status Asynchronously

See Export Response for Import Requisitions Task Status Asynchronously [page 87].

Export Deletion Request for Held Line Items on MRP Run Requisitions Asynchronously

The **Export Deletion Request for Held Line Items on MRP Run Requisitions Asynchronously** web services task exports the update request to SAP ERP to delete held line items from MRP-run requisitions before the aggregated requisition for those items can be successfully submitted.

Import Deletion Response for Held Line Items on MRP Run Requisitions

The **Import Deletion Response for Held Line Items on MRP Run Requisitions** web services task imports the success or failure response when held line items on MRP-run requisitions are successfully deleted in SAP ERP.

> **Note**
>
> Held line items on MRP-run requisitions are exported to SAP for deletion using the **Export Response for Import Requisitions Task Status Asynchronously** web services task.

The following is a sample response when held line items are deleted from the corresponding requisition in SAP ERP:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:Ariba:Buyer:vrealm_123">
   <soapenv:Header>
      <urn:Headers>
         <!--You may enter the following 2 items in any order-->
         <!--Optional:-->
```

```
         <urn:variant>?</urn:variant>
         <!--Optional:-->
         <urn:partition>?</urn:partition>
      </urn:Headers>
   </soapenv:Header>
   <soapenv:Body>
      <urn:AggregatedRequisitionRevertBudgetAsyncExportReply>
         <urn:status>Ok</urn:status>
      </urn:AggregatedRequisitionRevertBudgetAsyncExportReply>
   </soapenv:Body>
</soapenv:Envelope>
```

Export Submitted Requisitions for Budget Checks Using SAP Integration Suite, managed gateway for spend management and SAP Business Network

The **Export Submitted Requisitions for Budget Checks Using SAP Ariba Cloud Integration Gateway** exports submitted requisitions to SAP ERP for budget checks. This task can be used to export MRP-run requisitions which consist of line items eligible for demand aggregation (held line items). Exporting requisitions by using this task prevents deletion of held line items when budgets are reserved in SAP ERP.

Workflow for Managing Demand Aggregation and External Budget Checks for MRP-run Requisitions

The following is an overview of the high-level steps involved in creating aggregated requisitions for line items held for demand aggregation from MRP-run requisitions and performing budget checks for the aggregated requisition in SAP ERP:

1. The **Import Requisitions Asynchronously** web services task imports MRP-run requisitions from SAP ERP via SAP Integration Suite, managed gateway for spend management and SAP Business Network.
   The SAP Ariba solution creates purchase requisitions based on the MRP-run requisitions.
2. A user submits a requisition consisting of line items that are eligible for demand aggregation (held line items).
3. The SAP Ariba solution performs the following tasks:
   - Initiates budget checks for all line items on the requisition by using the **Export Submitted Requisitions for Budget Checks Using SAP Ariba Cloud Integration Gateway** web services task.
     SAP ERP saves a copy of the requisition. During the budget check process, held line items are not deleted from the requisition
   - Consolidates the held line items from the requisition and adds them to the Purchasing Workbench.
4. A purchasing agent accesses the Purchasing Workbench to add the held line items to an aggregated requisition and submits the requisition for approval.
5. The **Export Deletion Request for Held Line Items on MRP Run Requisitions Asynchronously** web services task sends a request to delete the line items from the original requisitions in SAP ERP.
   The status of the requisition changes from **Composing** to **Submitting**.
6. SAP ERP deletes the held line items from the MRP-run requisitions and saves a copy of the aggregated requisition.
7. The **Import Deletion Response for Held Line Items on MRP Run Requisitions** task imports the deletion confirmation response from SAP ERP.
   The status of the aggregated requisition changes to **Submitted**.

> **Note**
>
> If the held line items could not be deleted in SAP ERP, the status of the requisition reverts to **Composing**. The line items that failed to be deleted include the details of the error. In this case, the user can either retry exporting the deletion request or delete the failed items from the aggregated requisition and export the deletion request again.

8. The SAP Ariba solution initiates budget checks for the aggregated requisition using the **Export Submitted Requisitions for Budget Checks Using SAP Ariba Cloud Integration Gateway** web services task.
9. After successful completion of budget checks, the status of the aggregated requisition changes to one of the following:
   - **Submitted**, if the aggregated requisition passed budget checks in SAP ERP.
   - **Composing**, if the aggregated requisition failed budget checks in SAP ERP.

Collaborative Requisitions

A collaborative requisition (SAP Ariba Buying and Invoicing and SAP Ariba Buying) or shopping cart (SAP Ariba Catalog) is one in which a buyer can negotiate with one or more suppliers before completing the purchasing process.

- SAP Ariba Buying and Invoicing
- SAP Ariba Buying
- SAP Ariba Catalog

A requisition or shopping cart becomes collaborative in any one of the following scenarios:

- Dynamic collaboration: The requestor chooses to collaborate dynamically with the supplier to negotiate a better price or to procure an item that suits their needs better.
- N Bids and a Buy: Collaboration is enforced on a requisition as one or more line items in a requisition satisfy the conditions specified in the N Bids and a Buy policy.

For a collaborative shopping cart or requisition during the collaboration portion of the approval flow, a collaboration request is created for each invited supplier on that item, and then sent to them on SAP Business Network. A supplier views the collaboration request in their **Inbox** on SAP Business Network, reviews the details, and responds either by declining the request or by submitting a proposal, which contains the details of how they will fill the request. For example, if the line item is for a contractor to work as a temporary administrative assistant, the supplier's proposal might contain the names of proposed candidates, their resumes, and hourly rates.

After suppliers submit their proposals, users can review them on the **Collaboration** tab of the requisition in the user interface. Users can further negotiate a proposal or send questions to the suppliers using the message screen on the **Collaboration** tab. Suppliers can send messages back and respond with additional or amended proposals.

Dynamic Collaboration [page 131]

N Bids and a Buy [page 131]

Data Import Tasks for Collaboration Information [page 132]

Quick Enablement Collaboration Request [page 133]

Related Information

Collaborating with Suppliers

Dynamic Collaboration

Dynamic collaboration enables users to collaborate with multiple suppliers when procuring both customer catalog and non-catalog goods items including punchout items.

The option to choose to collaborate with multiple suppliers when editing a line item will be visible only to users belonging to the **Dynamic Collaboration User** group.

> ⓘ Note
>
> Dynamic collaboration is only applicable to goods items.

Related Information

Collaborating with Suppliers

N Bids and a Buy

In some organizations, if a requisition or shopping cart meets certain policy terms, it is required that the purchasing agents collaborate with a minimum number of suppliers and receive their bids before completing the purchasing process.

To enforce this, you define an N Bids and a Buy policy where you specify the minimum number of suppliers that need to be invited for collaboration, the fields that this policy is based on and the terms of the policy.

You can specify a policy based on a number of fields available in a requisition or shopping cart, for example:

- Requisition Amount
- Purchasing Unit
- SET ID (PeopleSoft only)
- Company Code
- Common Commodity Code
- Partitioned Commodity Code
- Line Item Type (Customer catalog / Services / Non catalog items)

You can also specify if violations of the policy are allowed. For example, you can specify if purchasing agents can still submit requisitions or check out shopping carts if they do not invite the minimum number of suppliers or not.

> ⓘ Note
>
> The N Bids and a Buy policy can be enforced when procuring both goods and services items.

In a requisition or shopping cart containing multiple line items, you can specify whether the line items must be grouped based on the policy attributes. When line items are grouped, the policy is applied on the total amount of the grouped line items.

You configure the N Bids and a Buy policy attributes in a policy lookup table (a CSV file). The lookup table contains data related to the fields that the policy is based on, the terms of the policy and the group that is responsible for enforcing the policy.

> **Note**
>
> The N Bids and a Buy policy is not applicable to change orders.

Related Information

Approval Process Management Guide

Data Import Tasks for Collaboration Information

There are two data import tasks for collaboration information: `Import Collaboration Reasons` and `Import Bidding Types`.

The following table lists each data import task and the CSV files that define collaboration information. Tasks appear in the recommended data import order.

Data Import Task	CSV Files Imported
Import Collaboration Reasons	`CollaborationReasons.csv`
Import Bidding Types	`BiddingType.csv`

You manually enter data in the CSV files.

Related Information

About Import Collaboration Reasons [page 132]
About Import Bidding Types [page 133]

Import Collaboration Reasons Task

The `Import Collaboration Reasons` data import task defines the reasons buyers and suppliers can give for various collaboration actions, such as rejecting, declining, and canceling. This task reads data from the CSV file named `CollaborationReasons.csv`.

Following is an example of `CollaborationReasons.csv`:

```
Cp1252
```

```
     UniqueName,DisplayName,Type,ProcureCategoryDriver

AutoReject,Automatically rejected,AutoReject,All
AutoCancel,Automatically canceled,AutoCancel,All
ProposalOther,Other,RejectProposal,All
ProposalMoney,Proposal too expensive,RejectProposal,All
ProposalStructure,Proposal terms are not favorable,RejectProposal,All
```

You can customize this file and load the updated reasons into your configuration. You can also add, edit, and delete reasons, with the exception of `AutoReject` and `AutoCancel`, which must remain in the file with their `UniqueName` values unchanged.

> **Note**
>
> You can use the `Import Collaboration Reason Translations` data import task to import translations for collaboration reasons.

Import Bidding Types

The `Import Bidding Types` data import task defines system-level bidding types that all users can use during the collaborative open bidding process. This task reads data from the CSV file named `BiddingType.csv`.

Following is an example of `BiddingType.csv`:

```
Cp1252


UniqueName,Name,ShowLowestBidAmount,ShowLowestBidPrice,ShowLowestBidSupplier,ShowBid
Rank,
ShowBidSupplierRank,HideBiddingInfoBeforeFirstBid

"Closed","Closed",false,false,false,false,false,false
"Amount","Open - Amount",true,true,false,false,false,false
"Rank","Open - Amount, Rank",true,true,false,true,false,false
"Supplier","Open - Amount, Rank, Supplier",true,true,true,true,true,false
```

> **Note**
>
> You can use the `Import Bidding Type Translations` data import task to import translations for bidding types.

Quick Enablement Collaboration Request

The quick enablement collaboration request feature enables buyers to collaborate with suppliers who are not members of SAP Business Network and do not possess SAP Business Network IDs (ANID) or suppliers whose ANIDs are not available in the buying solution.

This feature is similar to the quick enablement purchase order feature.

The steps involved in the quick enablement collaboration request process are as follows:

1. A user adds a supplier whose ANID is not available in the buying solution to the collaborative requisition.
2. The collaborative requisition is sent to SAP Business Network during the collaboration phase.
3. SAP Business Network creates a private account for this supplier and generates an SAP Business Network private ID.
4. The supplier receives an email invitation from SAP Business Network and clicks the link provided in the email to start the registration process.
5. If the supplier is not yet registered, they can accept the invitation by completing the registration process.
6. If the supplier is already registered, they can accept the invitation and combine their new relationship request with their existing account.
7. Once registered, an ANID is created and the supplier gains immediate access to the account.
8. The supplier can then punchin to the buying solution to view and respond to the collaboration request.

> **Note**
>
> Private IDs are unique per supplier location.You can view private IDs for supplier locations by using the **Suppliers** task in the Supplier Manager workspace in Ariba Administrator.

Enabling Quick Enablement Collaboration Request

Only suppliers that exist in the buying solution can be added to the invited suppliers list when creating a collaboration requisition.

If you want to invite suppliers who are not members of SAP Business Network, contact your system administrator to add these suppliers to your customer site and set their preferred ordering method to Email as SAP Business Network routes the collaboration requests only to those suppliers whose preferred ordering method is Email.

Before you invite suppliers who are not enabled on SAP Business Network for collaboration, you must customize the welcome letter that is sent to them along with the collaboration request.

You must also create a custom PDF with the process that these suppliers need to follow to respond to the collaboration request.

> **Note**
>
> To customize the welcome letter, navigate to **Supplier Enablement** and select **Configure** in the SAP Business Network buyer account. To create the custom PDF, contact your account representative and reach out to the SAP Services team. For more information, visit the SAP Store

Quick Enablement Collaboration Request Workflow

When adding suppliers to the invited suppliers list in collaborative requisitions, the requestor can add suppliers who are not on SAP Business Network. When requisitions are submitted, a collaboration request is created for each invited supplier during the collaboration phase, and sent to them on SAP Business Network.

For suppliers that do not have SAP Business Network accounts, the workflow is as follows:

1. The buying solution generates an SAP Business Network private ID.
   Private IDs are unique per supplier location. Administrators can view private IDs for supplier locations by using the Suppliers task in the Supplier Manager workspace in Ariba Administrator.
2. The collaboration request is then sent to SAP Business Network.
3. SAP Business Network then generates an ANID for the supplier and routes the collaboration request to the supplier which includes an invitation to sign in and complete the registration process.

After the supplier logs in and activates the SAP Business Network account, the supplier can punchin to the buying solution from SAP Ariba to view and respond to the collaboration request.

Collaboration Request Email Notifications

When you choose to collaborate with suppliers who are not registered on SAP Business Network, the collaboration request is routed to those suppliers through SAP Business Network by an email notification which includes an invitation to sign in and complete the registration process.

During the collaboration phase of the requisition, email notifications are sent to the suppliers and the buyers respectively when suppliers send proposals and the buyer responds to them.

Contact your Customer Success Manager or Account Manager to customize the welcome letter that is sent to Quick Enablement suppliers and to create the custom PDF with the process that these suppliers need to follow to respond to the collaboration request.

Predefined Notes in a Requisition

The pre-defined notes functionality allow notes to be saved internally in the document manager service for use in transactions. You can add and manage notes relevant to procurement transactions, and reuse them when required. These notes will be added by default when a user creates a requisition, and can be used to communicate additional information to suppliers in the purchase order. Pre-defined notes can be uploaded in any language and the notes will be added to the requisition in that language, if they meet the defaulting criteria. This is useful in communicating instructions or guidelines to suppliers in their language.

This functionality can be turned on by setting the parameter **Enable the document manager to save and use pre-defined notes** (`Application.Procure.DocumentService.Enabled`). This adds the **Document Manager** option to the **Procure-to-Pay Manager** workspace in Ariba Administrator.

Customer administrators responsible to maintain the pre-defined notes can download an empty CSV file, populate it with pre-defined notes and their defaulting conditions, and then upload this CSV file to add the notes to the document manager. They can also download all the pre-defined notes available for their site.

Notes can be configured to default based on a combination of one or more of the following values:

- Commodity Code
- Partitioned Commodity Code
- Supplier
- Ship To
- Procurement Unit
- Business Unit (for PeopleSoft variants only)
- SAPPlant (for SAP variants only)
- Item Category (for SAP variants only)
- Company Code (for SAP variants only)

For example, if a particular note is configured to apply to a requisition for a specific commodity code and a specific supplier, then this note will be added by default when a requisition is created with that commodity code and for that supplier.

Customer administrators can also set up rules to default notes in a requisition. In addition to the 5 inbuilt fields that are supported, up to 3 custom fields can be configured for use in rules for defaulting notes. The custom fields must be of type text or string. Customers can request SAP Ariba to create these custom fields and configure them using the appropriate template.

In addition to the defaulted notes, users can search for pre-defined notes and choose any note to add manually to the requisition. They can also delete any note that has been added by default. Adding and deleting notes is only possible when the requisition has not been approved or ordered. When the requisition is approved and a purchase order is created, the notes will be shown in the order as non-editable text. These notes will be sent to SAP Business Network as well as customers' back-end systems in the purchase order. It will also be included in the new version of an order, when the order is copied or changed.

How notes are defaulted in a requisition

Notes are added to requisitions by default based on matching criteria configured by customer administrators in the rules file. The rules can be assigned a numerical value to indicate priority, and rules will be applied to default notes according to this priority. For example, a rule with priority 1 will be applied first. If no notes are available for the corresponding matching criteria, a rule with priority 2 will be applied, and so on.

> **Note**
>
> Two rules cannot have the same priority number.

To ensure that notes are added by default for all requisitions matching one of the criteria fields specified in the rule, a wildcard value can be entered in the other fields. You can enter **ALL** in the other criteria fields or leave them blank to act as wildcard values. For instance, if the rule is defined to default based on matching commodity code and supplier, and a note is defined with values for the commodity code with **ALL** specified for the supplier field, the note is added to all requisitions with a matching commodity code.

Related Information

Adding Notes in the Document Manager [page 138]
Adding Rules in the Document Manager [page 137]

Adding Rules in the Document Manager

Context

Customer administrators must configure rules for defaulting notes. Defaulting will not work if no rules have been configured. Administrators can use the rules CSV file to add and upload rules for defaulting notes in the **Document Manager**.

Procedure

1. In Ariba Administrator, open **Procure-to-Pay Manager** > **Document Manager** .
2. In the **Download all rules configured for this site** section, click **Download Rules** to download any rules CSV file that is available. This file has a name with the following pattern `<sitename>-Rules.csv`. If no rules CSV file has been configured this downloads a blank file with only the column headers.
3. Add the following information in the corresponding column of the CSV file:

1. `ruleName`: Enter the name of the rule.
2. `rulePriority`: Enter a numerical value to indicate the priority for this rule.

3. `fieldNames`: Enter a comma-separated list of fields to be used as the matching condition for this rule. If this condition is met, the corresponding note will be added to the requisition by default. For example, `partitionedCommodityCode,commonCommodityCode,cus_Note`, where `cus_Note` is an example of a custom field configured for use in defaulting notes. The names of the fields to be used can be taken from the notes template file.
4. Repeat the previous step to add more rules.
5. In the **Upload CSV file to create rules** section, click **Choose file**.
6. Choose the updated rules CSV file.
7. Click **Upload Rules** to add the rules.

Adding Notes in the Document Manager

Context

Administrators can use the notes CSV file to add and upload pre-defined notes in the **Document Manager**.

Procedure

1. In Ariba Administrator, open **Procure-to-Pay Manager** > **Document Manager**.
2. In the **Download the notes template to create the external notes CSV file** section, click **Download Notes Template** to download the empty `<sitename>-Header.csv` file.
3. Add the following information in the corresponding column of the CSV file:
1. `Name`: Enter the name of the note.
2. `CommodityCode`: Enter the commodity code if it is to be used as a matching criterion for this note.
3. `PartitionedCommodityCode`: Enter the partitioned commodity code if it is to be used as a matching criterion for this note.
4. `SAPPlant`: Enter the plant value if it is to be used as a matching criterion for this note.
5. `ShipTo`: Enter the Ship To value if it is to be used as a matching criterion for this note.
6. `ProcurementUnit`: Enter the Procurement Unit value if it is to be used as a matching criterion for this note.
7. `BusinessUnit`: Enter the Business Unit value if it is to be used as a matching criterion for this note.
8. `Supplier`: Enter the Supplier value if it is to be used as a matching criterion for this note.
9. `itemCategory`: Enter the Item Category value if it is to be used as a matching criterion for this note.
10. `Companycode`:Enter the Company Code value if it is to be used as a matching criterion for this note.
11. `Text`: Enter the note to be added.
4. Repeat the previous step to add more notes.
5. In the **Upload CSV file to create external notes** section, click **Choose file**.
6. Choose the updated `<sitename>-Header.csv` file.
7. Click **Upload Notes** to add the notes.

Examples: Configuring rules and defining values for the rules to retrieve predefinied notes

This section describes the following scenarios where customer administrators can configure rules and define values for them to retrieve predefined notes and default them to the line item in a purchase requisition:

- Configure a rule in the rules.csv file with matching criteria and the values for the rule in the notes.csv file to retrieve predefined notes [page 139]
- Configure an additional field and its value in the notes.csv file to retrieve predefined notes without changing the corresponding rule in the rules.csv file [page 140]
- Configure an additional rule in the rules.csv file with different matching criteria to retrieve predefined notes [page 140]

Configure a rule in the `rules.csv` file with matching criteria and the values for the rule in the `notes.csv` file to retrieve predefined notes

Customer administrators can configure a rule with matching criteria and priority in the `rules.csv` file. Customer administrators can follow the steps described in the Adding Rules in the Document Manager [page 137] topic to configure the `rules.csv` file.

First, configure the following columns in the `rules.csv` file:

- `ruleName`
- `rulePriority`
- `fieldNames`

Then, name the rule, configure priority of the rule, and define matching criteria for the rule as follows:

`ruleName`	`BasedonPlantAndsupplier`
`rulePriority`	`1`
`fieldNames`	`sapPlant,supplier`

The comma-separated list of fields in the `fieldNames` column acts as the matching criteria for the rule.

After configuring the rule, the `rules.csv` file appears as follows:

```
ruleName","rulePriority","fieldNames"
"BasedonPlantAndsupplier","1","sapPlant,supplier"
```

Now, customer administrators can create the `notes.csv` file and define values for the matching criteria configured in the `rules.csv` file. Customer administrators can follow the steps described in the Adding Notes in the Document Manager [page 138] topic to configure the `notes.csv` file.

The `notes.csv` file can have multiple entries for one rule. In this example, the customer administrators provide the following values to the `sapPlant,supplier` matching criteria in the notes.csv file and create multiple entries:

Values for `sapPlant`	300025, 300026
Values for `supplier`	0000003920, 0000003921

The `notes.csv` file appears as follows:

```
Notes.csv
Name,companyCode,sapPlant,itemCategory,supplier,commonCommodityCode,partitionedCommo
dityCode,shipTo,Text
AZRULE2NOTE001, ,300025, ,0000003920, , , ,”Notes for AZRULE2NOTE001”
AZRULE2NOTE002, ,300026, ,0000003921, , , ,”Notes for AZRULE2NOTE002”
AZRULE2NOTE003, ,300026, ,0000003920, , , ,”Notes for AZRULE2NOTE003”
```

With this configuration, all the predefined notes that comply with the matching criteria in the rule defined in the `rules.csv` file and the values provided to the matching criteria in the `notes.csv` file are defaulted to the line item in the requisition.

Configure an additional field and its value in the `notes.csv` file to retrieve predefined notes without changing the corresponding rule in the `rules.csv` file

Customer administrators can add additional fields in the `notes.csv` file without changing the corresponding rule in the `rules.csv` file. In this example, a new entry with the additional field representing `companyCode` is added to the `notes.csv` file. The updated `notes.csv` file appears as follows:

```
Notes.csv
Name,companyCode,sapPlant,itemCategory,supplier,commonCommodityCode,partitionedCommo
dityCode,shipTo,Text
AZRULE2NOTE001, ,300025, ,0000003920, , , ,”Notes for AZRULE2NOTE001”
AZRULE2NOTE002, ,300026, ,0000003921, , , ,”Notes for AZRULE2NOTE002”
AZRULE2NOTE003, ,300026, ,0000003920, , , ,”Notes for AZRULE2NOTE003”
AZRULE2NOTE004,3000,300025,,0000003920, , , ,”Notes for AZRULE2NOTE004”
```

The new entry still corresponds to the following rule in the `rule.csv`:

```
ruleName","rulePriority","fieldNames"
"BasedonPlantAndsupplier","1","sapPlant,supplier"
```

With this new entry, the predefined notes for the company code 3000 that match the `sapPlant` 300025 and `supplier` 0000003920 default to the line item of the requisition. In this way, values provided to the additional fields are used to retrieve specific predefined notes.

Configure an additional rule in the `rules.csv` file with different matching criteria to retrieve predefined notes

Customer administrators can configure additional rules in the `rules.csv` file. If matching predefined notes were not retrieved for the first rule defined in the `rules.csv` file, the system can try locating the predefined notes that match the second rule in the `rules.csv` file.

> **Note**
>
> The rules in the `rules.csv` file must differ in matching criteria.

The `rules.csv` file can contain more than one rule, and the customer administrator must assign priorities to them. A rule runs based on the priority assigned to it.

In this example, a second rule with priority 2 is configured in the `rules.csv` file, which appears as follows:

```
ruleName","rulePriority","fieldNames"
"BasedonPlantAndsupplier","1","sapPlant,supplier"
“BasedOnCommodityCode”,2,commonCommodityCode
```

> **Note**
>
> The second rule is run only if no predefined notes are retrieved for the first rule.

The matching criteria defined in the second rule is `commonCommodityCode`. Two entries that define the values for the second rule are configured in the `notes.csv` file, which appears as follows:

```
Notes.csv
Name,companyCode,sapPlant,itemCategory,supplier,commonCommodityCode,partitionedCommo
dityCode,shipTo,Text
AZRULE2NOTE001, ,300025, ,0000003920, , , ,”Notes for AZRULE2NOTE001”
AZRULE2NOTE002, ,300026, ,0000003921, , , ,”Notes for AZRULE2NOTE002”
AZRULE2NOTE003, ,300026, ,0000003920, , , ,”Notes for AZRULE2NOTE003”
AZRULE2NOTE004,3000,300025,,0000003920, , , ,”Notes for AZRULE2NOTE004”
“NOTE001”,,53101602,,,”Text for NOTE001”
“NOTE002”,,43,,,”Text for Text for NOTE002”
```

Understanding Order Methods

An ordering method specifies the way SAP Ariba Buying and Invoicing and SAP Ariba Buying process and send orders to suppliers.

There are two instances in the buying solution where ordering methods are configured:

- Order method parameter: This configuration is performed by SAP Ariba Customer Support on request. This configuration checks the order attributes and determines the appropriate order process method (for example, PCard, Ghosted PCard, Direct Order, ERP Order, etc.).
- Preferred order method for each supplier location: You can select a preferred ordering method for each supplier when creating or editing supplier information. This setting determines how a particular supplier is configured to accept orders (for example, electronically or as printed orders).

When a purchase requisition is fully approved, the buying solution converts that requisition into purchase orders. Each requisition can potentially generate one or more orders. For example, if a requisition contains items from different suppliers (or different supplier locations), the buying solution typically generates several orders from that requisition, one to each supplier location.

The buying solution decides how to split line items into orders by testing each line item against a set of possible order methods.

Each order method includes a logic to specify which items can be grouped ("aggregated") together. For example, an order method typically creates more than one order if:

- The line items on the requisition have different billing information.
- The line items on the requisition are from different purchase organizations, inventory organizations, or have other accounting differences.
- The line items on the requisition are from several different supplier locations.

Each supplier location also includes a field that indicates whether that supplier allows line items with different ship-to addresses on a single order. If this field value is set to true, the default configuration always separates items with different shipping addresses onto different orders.

If you want the purchase order split to be based on certain required fields, contact SAP Ariba Customer Support to configure those fields.

The next step in the process is for the buying solution to identify the preferred ordering method that is configured for the corresponding supplier.

A supplier location may be configured to accept orders in any of the following methods:

- **Fax**: Orders are sent to SAP Business Network, along with the supplier's fax number, using a private ID.
- **Email**: Orders are sent to SAP Business Network, along with the supplier's email address, using a private ID.
- **Printing**: If you do not want to send orders to the supplier through SAP Business Network, choose this ordering method. When you specify Printing, users must manually process orders sent to the supplier.
- **No Choice**: If an order method is not selected, the system defaults to this order method. This order method works identical to the **Printing** order method.
- **Electronic Ordering**: Orders are sent to SAP Business Network using the supplier's SAP Business Network ID.

> **Note**
>
> - If your buying solution is configured to process orders to external ERP systems, then the order methods configured at the parameter level take precedence over the preferred order methods that are configured for each supplier location.
> - For routing purchase orders to additional supplier contacts, each supplier location defined for your site must be configured to accept orders using the Email routing method. Emails to additional supplier contacts will not be sent with other routing methods (Fax, Print, or Online).

The preferred ordering method takes precedence when a supplier's preferred ordering method is "**Printing**." Irrespective of whether it is a PCard or direct order, such a supplier accepts orders only in a printed format. In this scenario, the order gets processed electronically until it moves into the Ordering state, but remains in this state until the user marks it as Ordered and manually sends it to the supplier.

> **Note**
>
> - If the order method parameter is configured as **ERPCC**, changes made to the supplier's preferred ordering method affect order creation when the order is changed. When the preferred ordering method is changed after the creation of a purchase order, the change order process triggers the creation of a new order. For example, say the preferred ordering method is set as print and order PO1 is created. After the creation of PO1, the preferred ordering method is changed to email. If PO1 is changed now to add a new line item, a new order PO2 is created for the newly added line and sent to SAP Business Network. Any change to the line items in the original purchase order continues to be updated on PO1 and the next version, PO1_V2, is created.
> - If the **Skip ordering method check for previous supplier location from Carbon Copy to SAP Business Network on supplier location change** (`Application.Procure.SkipOrderMethodCheckForCarbonCopyToSAPBusinessNetwork`) parameter is enabled, SAP Ariba skips the ordering method check for the previous supplier location when a supplier location is changed on a line item during a change order. This parameter applies when the original order uses Carbon Copy, and the updated supplier location is configured to receive purchase orders through the SAP Business Network. For more information, see Skip ordering method check for previous supplier location from Carbon Copy to SAP Business Network on supplier location change.

Order Methods [page 144]

General Order Methods [page 146]

ERP Order Methods [page 149]

Order Numbers [page 150]

Related Information

Common Data Import and Administration for SAP Ariba Procurement Solutions

Order Methods

An order method is a Java class that creates the approvable documents and copies the line items from a requisition to the order.

The set of order methods configured at your site determine how your buying solution must process the orders.

The logic of the order method determines which line items are to be grouped together on an order. The default configuration supplies a set of order methods, which implement typical business rules for splitting requisitions into orders. Contact SAP Ariba Customer Support to configure the order methods based on your business requirements.

Related Information

Available Order Methods [page 144]
Carbon Copy Direct Order (CC:DO) Orders [page 146]

Available Order Methods

The default configuration provides a set of order methods for both SAP Ariba Buying and Invoicing and SAP Ariba Buying. Each method is appropriate for different circumstances or different kinds of orders.

The following table lists the ordering methods available to you based on your site's ERP system:

ERP	Ordering Methods
Generic	• `Ariba Network Dummy Order` • `Ariba SSP Ghost PCard` • `Ariba SSP PCard` • `SSP All Direct Order` • `Generic PO ERP` • `Generic PO ERPCC`
PeopleSoft	• `PS PO ERP` • `PS PO ERPCC` • `Ariba Network Dummy Order` • `Ariba SSP Ghost PCard` • `Ariba SSP PCard` • `SSP All Direct Order`

ERP	Ordering Methods
SAP	• `SAP PO ERP` • `SAP PO ERPCC` • `Ariba Network Dummy Order` • `Ariba SSP Ghost PCard` • `Ariba SSP PCard` • `SSP All Direct Order`

> **Note**
>
> The `Ariba Network Dummy Order` order method is used only for testing purposes. This order method processes orders by setting up a dummy Order Recipient (in a real business scenario, an Order Recipient is an external ERP and/or SAP Business Network) and moves the orders to the **Ordered** state.

> **Note**
>
> The ranking of the order methods (the order in which the methods are listed, when configured) determines the way the buying solution is going to process the orders in the system.

Each time a requisition is fully approved, the buying solution tests the line items on that requisition against the order methods as configured at your site, and in the order in which they are listed. Each order method returns `true` or `false` value, to indicate whether it can process the given line item. The buying solution uses the first order method that returns a `true` value for the line item in question.

For example, if you are integrating with a PeopleSoft ERP (that is, you want your orders processed in an external PeopleSoft ERP system), and your order methods are configured in the following order to:

- Use a ghosted purchasing card, if applicable
- Use the requester's purchasing card, if applicable
- Create both a dispatched order for PeopleSoft and a direct order for SAP Business Network
- Send an order to PeopleSoft

In this example, the buying solution uses a purchasing card order method whenever the requisition involves a purchasing card and its use is approved as per your company's policies.

If the use of a purchasing card is not appropriate or the requisition does not involve a purchasing card, the buying solution then checks for the next available order method, which in this example, is to send a dispatched order for PeopleSoft and also a direct order to SAP Business Network. This value returns `true`, and so the order is sent to both PeopleSoft ERP and SAP Business Network.

Consider a scenario where none of the earlier methods have succeeded, the buying solution then implements the last order method that is listed, and sends the order to the PeopleSoft ERP directly.

> **Note**
>
> Typically, in any configuration, the last order method listed must be a "catch-all" with no restrictions, which is able to handle any line item that has not previously been handled. This is the All Direct Order method.

> **Note**
>
> Your buying solution does not consider any order method that is ranked below the All Direct Order method. This order method processes any kind of order. The All Direct Order method creates the order and sends it to SAP Business Network.

Carbon Copy Direct Order (CC:DO) Orders

If your site is configured to use both an external ERP system and SAP Business Network, you can use an optional feature called CC:DO (carbon copy direct order). The CC:DO feature sends the order to SAP Business Network and a "carbon copy" (CC) to your ERP system.

When you use CC:DO, the SAP Business Network copy is the real order. SAP Business Network owns the order and is responsible for it. On the ERP system, the order is for information only and cannot be changed. Your ERP system administrators can receive against the order or pay it, but they cannot make any changes to the order.

To configure your buying solution and use the CC:DO feature, contact SAP Ariba Customer Support and ensure that this order method is ranked higher than the other order methods. For example, instead of using the order method that sends an order directly to the PeopleSoft ERP, you must use another variation of this order that creates a direct order for SAP Business Network and a dispatched order for PeopleSoft ERP.

General Order Methods

General order methods are those that are not restricted to a specific ERP system. These are appropriate when you want your buying solution to send direct orders or SAP Business Network orders.

Related Information

SAP Ariba Order Method [page 147]
SAP Ariba PCard Master Order Method [page 147]
SAP Ariba PCard Order Method [page 147]
SAP Ariba GhostPCard Order Method [page 148]
SSP All Direct Order Method [page 149]

SAP Ariba Order Method

The SAP Ariba order method is common to all ordering methods. It aggregates line items when a set of conditions is met.

Conditions that must be met include:

- Line items are from the same supplier location
- Line items have the same shipping address
- Line items have the same billing address
- Line items are all customer catalog items, or all ad-hoc
- Line items are punchout items with the same header
- Line items are part of the same contract

> **Note**
>
> For change orders, whether line items are aggregated also depends on the change order restrictions of the supplier location.

SAP Ariba PCard Master Order Method

The SAP Ariba PCard Master order method is intended for configurations in which accounting purchasing cards are associated with all PCard orders.

> **Note**
>
> This order method is not available in the default configuration.

SAP Ariba PCard Order Method

The SAP Ariba PCard order method is intended for configurations where individual requesters have personal purchasing cards.

PCard accepts a line item if the following conditions are met:

- Line items are from the same supplier location
- Line item is from a customer catalog (not ad hoc)

  > **Note**
  >
  > In the default configuration, requisitions with non-catalog items cannot be processed to create PCard orders. To configure the SAP Ariba solution to process a requisition with non-catalog items as a PCard order, contact your Designated Support Contact who will log a service request and an SAP Ariba Customer Support representative will follow up to configure your site accordingly.

- Line items have the same shipping and billing addresses

- Line item is from a supplier location that accepts purchasing cards
- Line items are part of the same contract
- Requester has a purchasing card
- Line items are punchout items with the same header
- Purchasing card number, user name, and expiration date on the purchasing card are valid

PCard orders aggregate line items if they have the same line item split accounting, purchasing card number, transaction number, and transaction date.

> **Note**
>
> PCard details are encrypted. If encryption fails while ordering, a purchase order is created instead of a PCard order.

In the default configuration, your buying solution creates multiple PCard orders for the same supplier location when the associated requisition includes line items with different accounting values. You can choose to create a single PCard order for a supplier location even if the line items on the associated requisition include different accounting values. For help with this configuration, contact your Designated Support Contact who will log a service request and an SAP Ariba Customer Support representative will contact you.

> **Note**
>
> This option is applicable for sites configured to load Level 2 (line-level) charges for PCards.

SAP Ariba GhostPCard Order Method

The SAP Ariba GhostPCard order method is intended for configurations in which purchasing cards are associated with supplier locations rather than individual users.

GhostPcard accepts a line item if the following conditions are met:

- Line item is from a customer catalog
- Line item is from a supplier location that accepts purchasing cards
- Supplier location has a ghosted purchasing card

It aggregates line items if the line items have the same line item split accounting, purchasing card number, transaction number, and transaction date.

> **Note**
>
> The conditions described for the `SAP Ariba Order Method` also apply.

SSP All Direct Order Method

The SSP All Direct order method is intended for use as a "catch-all" method. It handles all requisitions unconditionally and always generates a direct order to the appropriate supplier.

If you use this order method, it is essential that the requisition be completely filled out, with accurate supplier information, supplier location information, and accounting information.

ERP Order Methods

The default configuration provides order methods specific to particular ERP configurations. There are three basic order methods, and three carbon copy direct order (CC:DO) order methods.

The three basic order methods are:

Order Method	Description
`Generic PO ERP`	Creates purchase orders to export to the Oracle Financials.
`PS PO ERP`	Creates purchase orders to export to the PeopleSoft ERP.
`SAP PO ERP`	Creates purchase orders to export to the SAP ERP.

The order methods that facilitate a carbon copy as well as a direct order are:

Order Method	Description
`Generic PO ERPCC`	Creates both a frozen purchase order for Oracle Financials and a direct order for SAP Business Network.
`PS PO ERPCC`	Creates both a dispatched purchase order for PeopleSoft and a direct order for SAP Business Network.
`SAP PO ERPCC`	Creates an SAP Ariba ("AR") document type purchase order for SAP and a direct order for SAP Business Network.

By default, the CC:DO order methods do not accept ad hoc line items. To change this default behavior, contact SAP Ariba Customer Support.

Order Numbers

When the buying solution first creates a purchase order, it assigns the order a `UniqueName`, which is always unique and sequential. `UniqueName` is formatted with a prefix, immediately followed by a sequential number, for example, DO1234. The `UniqueName` is an internal ID created as a placeholder.

After the order is created, it is assigned an `OrderID`, which is initially the same as the `UniqueName`. The `OrderID` is visible to users. It can change after it is initially set, in the following circumstances:

- For orders that are sent to PeopleSoft or SAP (including those sent with CC:DO), the buying solution allows the ERP system to generate the PO number. The buying solution imports that number from the ERP system, and uses it as the `OrderID`.

  > **Note**
  >
  > If the underlying ERP system is Oracle Financials, the buying solution generates the order number rather than importing the number from Oracle. This order number is then exported to Oracle.

- For orders that are matched to a contract, the `OrderID` is changed to reflect the contract name.
- For change orders, each change order is a revision of an existing order. The new version has a new `UniqueName`, but it has the `OrderID` of the original version of the order.
  In such circumstances, the `OrderID` field for orders is not necessarily sequential.

In addition, the buying solution creates temporary approvable documents to use during approvable document search. Those temporary approvable documents are assigned order numbers, but are later discarded.

Order Numbers That Appear to Be Skipped

Sometimes, the way the buying solution assigns order numbers causes an order number to be allocated in such a way that the number appears to be skipped in the numbering scheme. Users searching for these order numbers cannot find matching orders in a search.

This happens because the buying solution assigns an order number to a requisition at specific stages, sometimes before the order is actually created (such as when the first approval is made on a submitted requisition). After an order number is assigned to a requisition, that order number cannot be reused, even if the order itself never existed.

Continuing with the example of an order number being assigned to a requisition as soon as the first approver submits approval, suppose the requisition is then withdrawn and deleted while it still has a status of **Submitted**. The buying solution already used the order number but never created the order, so searching for that order number will not produce a matching order. (Searching for the requisition will not produce a match, either, This is because the requisition was deleted, rendering it inactive.)

In the following example, it appears that order number PO101 was skipped, but it was not; the number was simply consumed before the order was created:

- Requisition PR10 is created, submitted, fully approved, and creates order PO100.
- Requisition PR11 is created, submitted, approved by one approver, and assigned an associated order number, PO101.
- Requisition PR12 is created, submitted, fully approved, and creates order PO102.

- Requisition PR11 is withdrawn and deleted. Order PO101 is nonexistent (it was never created), but PO100 and PO102 exist, so it appears as if PO101 was skipped even though it was not.

Configuration of Incoterms

You can configure your SAP Ariba solution to include Incoterms on requisitions and the associated orders.

Incoterms are a series of predefined commercial terms published by the International Chamber of Commerce (ICC). The Incoterms three-letter trade terms are related to common contractual sales practices. They are intended to clearly communicate the tasks, costs, and risks associated with the transportation and delivery of goods.

For detailed descriptions of the Incoterms rules, see the International Chamber of Commerce website.

Related Information

Import Incoterms Task [page 152]

Import Incoterms Task

The `Import Incoterms` task imports the Incoterms code and the location applicable for the incoterms. This task uses the `IncoTerms.csv` file to import the Incoterms code and location for your site.

The following table describes the fields in the `IncoTerms.csv` file:

Field name	Required	Type and description	Sample value
INCO1	Yes	String. Refers to the Incoterms code or the unique internal identifier for the Incoterms.	DAT
ORTOB	No	Boolean. Specifies whether location is applicable for the Incoterms.	Yes
NAME1	Yes	String. Description for the Incoterms code.	Delivery at terminal

For example:

```
"UTF-8"
INCO1,NAME1,ORTOB
FCA,Free Carrier,
CPT,Carriage Paid To,
CIP,Carriage and Insurance Paid,
DAT,Delivered At Terminal,
```

Payment Terms Filtering Based on the Incoterms Location Field on Requisition Line Items and Invoices for SAP Variants

Payment terms can be filtered based on the value of the Incoterms location in the **Incoterms** field on requisition line items and invoices.

This option is applicable only if your SAP Ariba Buying and Invoicing solution is integrated with SAP Ariba Supplier Management solutions.

This option is controlled by the `Application. Common.EnableIncoTermsLocationDeafulting` site-configuration parameter, which should be set to true to enable the filtering. If this parameter is enabled, payment terms is filtered for a supplier and supplier location combination if **Incoterms** is not null.

Asset Management for SAP-ERP Integrated Sites

Organizations can integrate with the asset management component in SAP ERP to manage their fixed assets through various stages in the lifecycle of those assets.

Integration with the asset management component in SAP ERP can be achieved using either SAP Integration Suite, managed gateway for spend management and SAP Business Network or SAP Ariba APIs. Using this feature, asset managers before approving requisitions can specify asset class and asset line type values for line items configured as assets. When a requisition consisting of asset lines is fully approved, asset master records are created for those line items in SAP ERP. The buying solution then assigns unique asset numbers to those items on the requisition based on the referenced asset master records.

Asset managers can add asset information only for those items that are configured as assets. Further, you must define the asset class values and asset line types for items configured as assets. Perform the following tasks to mark items as assets and sub-assets:

1. Run the `Export Asset Data` task to download the CSV file used by this task.
2. For each company code, ensure that you update the default values for the `ANLN1` and the `ANLN2` fields in the CSV file as follows:
   - Enter `none` for the `ANLN2` field.
   - Enter `Unspecified` for the `ANLN1` field.
     `Unspecified` refers to a temporary or a dummy asset value. With this setting, when users create requisitions, the **Asset Number** field for asset line items automatically default to `Unspecified`. This ensures errors do not occur when users submit requisitions without entering asset values, specifically for sites configured for budget checking in external systems. Dummy asset values are automatically overridden with appropriate asset values referenced in the asset master records when asset managers complete their approvals.

     > **Note**
     >
     > If your ERP system does not accept string values, such as `Unspecified` for asset values, specify a dummy asset value in the appropriate format.

   Example:

   ```
   "UTF-8"
   BUKRS,ANLN1,"TXA50,TXT50,TXA50,TXT50",ANLN2
   "3000","Unspecified","Unspecified","none"
   "1000","Unspecified","Unspecified","none"
   ```
3. Import the updated CSV file for the `Import Assets` task and then run the task.
4. Complete the `Import Asset Classes` and the `Import Asset Line Types` data import tasks to define asset classes and asset line types.
5. Run the `Export ERP Commodity to Commodity Mapping` task to download the CSV file used by this task.
6. Update the CSV file as follows:
   - Ensure that the `AccountCategory` field for asset items is specified as `A`.
   - Specify `Unspecified` for the `Asset` field corresponding to the appropriate commodity codes.

> **Note**
>
> If you have used a different value instead of `Unspecified` in the CSV file used by the `Export Asset Data` task, use the same value for the `Asset` field used by the CSV file used by this task.

Example:

```
UTF-8,,,,,,,,,,,,,,,,,,,,,,,,,,
AccountCategory,AccountType,CompanyCode,ActivityNumber,"Asset,CompanyCode",Asset,
BillTo,CommonId,CompanyCode,"GeneralLedger,CompanyCode",GeneralLedger,InternalOrd
er,ItemCategory,CurrForMinAndMaxAmt,CompanyCode,Network,MaterialGroup,SAPPurchase
Group,ShipTo,WBSElement,CommonIdDomain,SubAsset,DeliverTo,MaxAmt,MinAmt,Preferred
A,Asset,,,3000,Unspecified,,43211507,3000,3000,405000,,M,USD,,,12,,3300,,unspsc,n
one,Arnold Davis,9999999999,150,No
```

7. Run the `Import ERP Commodity to Commodity Mapping` task to import the updated CSV file for commodity code mapping.

Related Information

Purchasing Guide for Procurement Professionals
Common Data Import and Administration for SAP Ariba Procurement Solutions

Export of Purchase Orders

SAP Ariba Buying and Invoicing and SAP Ariba Buying support purchase order export tasks. The purchase order export feature lets you extract purchase order data from your buying solution and exports it to comma-separated value (CSV) files.

Advanced ship notices are part of the purchase order export. Your buying solution creates an advanced ship notice when they receive a ship notice message from SAP Business Network.

> ### Note
>
> The buying solution updates only the advanced ship notice status of the orders (**Shipping** or **Shipped**) as received from SAP Business Network, and includes details such as the Tax Amounts, Comments, or Attachments only at the order header level and not at the line item level.

Purchase orders and advanced ship notices are queued for export each time certain status changes occur. For example, a purchase order might be queued for export when its status changes to **Ordering**, and queued again when its status changes to **Ordered**.

You can ask your SAP Ariba Customer Support to configure the export queue statuses for purchase orders and advanced ship notices for your site.

> ### Note
>
> - You can also choose to include any purchase order header level fields that when changed, trigger a purchase order to the queue for export. To do this, ask your SAP Ariba Customer Support Administrator to include the required purchase order fields in the appropriate parameter.
> - If you requested your SAP Ariba Support run the scheduled task **Auto Close Fully Reconciled PO**, the fully reconciled purchase orders to be closed might re-appear in your daily purchase order export task, which might cause your daily export task to fail because of large data volumes. Therefore, run the closed order export and daily export tasks at different times.

In the default configuration, the **Amount Reconciled** and **Closed** fields are included.

> ### Note
>
> The standard export data discussed in this chapter are enhanced periodically to include more fields. New fields, when added, are always inserted at the end of the record in all exported CSV files. Customers that are implementing further integrations with these exported CSV files must ensure that their integrations are robust enough to ignore these newly added fields, if they are not required.

Export of Attachments and Comments with Purchase Orders [page 157]

Purchase Order Import Integration with ERP Systems [page 158]

Import Purchase Order Close Status Task [page 159]

Workflow of Failed ERP Purchase Order Export [page 160]

Exporting Purchase Orders [page 161]

Export Legal Fields in Purchase Order cXML sent to SAP Business Network [page 161]

Purchase Order Data Files [page 162]

Resend Purchase Orders Stuck in Ordering or Canceling State [page 171]

Export of Attachments and Comments with Purchase Orders

You can export attachments and comments at both the header level and the line item level, if this functionality is enabled for your site. You can ask your SAP Ariba Customer Support Administrator to enable it for you.

Only attachments and comments that are visible to suppliers on SAP Business Network can be sent to the external system. Exporting attachments and comments is supported only on the web services channel.

The Web Services Description Language (WSDL) files for the **Export Purchase Orders** and **Export Change Purchase Orders** data integration tasks have been modified to export the attachments and comments. You must import the updated WSDL file into your Enterprise Application Integration (EAI) tool and update your integration to get the enhanced functionality.

For more information on web services integration, see Administration and Data Maintenance Fundamentals.

For customers using SAP ERP as their external ERP system, adapter support to consume the purchase order attachment and comment export in SAP ERP is available in SAP Ariba Cloud Integration Release 9.0. For customers using PeopleSoft or generic ERP systems, the buying solution publishes a SOAP message with the purchase order attachment and comment details, for both new purchase orders and changed purchase orders. Customers can use an EAI tool to consume the SOAP message and enable integration with their ERP system using web services.

Sending Purchase Order Attachments and Comments in Response to Requests from External Systems

An alternative method to exporting purchase order attachments and comments can be enabled for sharing attachments and comments with external systems. You can send the details of purchase orders, including their attachments and comments, in response to web service requests from external systems.

The external system sends a web service request for the details, along with the unique identifier (PO ID) and the version number of the purchase order in the buying solution from SAP Ariba, for which the details are needed. When the request is received, the **Fetch Purchase Order Details** integration task searches for the purchase order and sends the web service response with the complete purchase order details, including the attachment and comments, to the external system.

If there are multiple versions of a purchase order, the buying solution from SAP Ariba sends details for the purchase order version sent in the request, even if a newer version is available. For example, if there is a newer version of the purchase order in the buying solution, say v2, and the external system requests details for a previous version, say v1, the details will be sent for v1.

To enable the **Fetch Purchase Order Details** web service, you must specify the endpoint URL. The external system publishes a SOAP message corresponding to the task and posts that message to your Enterprise Application Integration (EAI) tool. The EAI tool transforms the message into a format that allows your buying solution from SAP Ariba to consume the message. SAP Ariba provides a new Web Services Description Language (WSDL) file for this.

You must import the WSDL file into your EAI tool to ensure that the generated SOAP messages adhere to the WSDL file.

Purchase Order Import Integration with ERP Systems

There is integration between buying solutions from SAP Ariba and third-party ERP systems, such as SAP and PeopleSoft using web services.

Related Information

Purchase Order Import Integration with SAP ERP [page 158]
Purchase Order Import Integration with PeopleSoft [page 159]

Purchase Order Import Integration with SAP ERP

End-to-end integration is achieved using SAP NetWeaver PI. SAP NetWeaver PI acts as an EAI tool to ensure that SOAP messages (Request and Response) are exchanged between the buying solutions and SAP ERP.

In addition to the existing transactions that were available on file based integration, the following new transactions are available that can be configured to use web services for integration.

- Purchase order
- Change purchase order
- Cancel purchase order

> **Note**
>
> These transactions can also be configured to integrate using files.

> **Note**
>
> When importing canceled and changed purchase order data, it is possible that the imported data contains duplicate entries of the same purchase order. You must ensure the duplicate entries are removed from your ERP system before processing the information.

Purchase Order Import Integration with PeopleSoft

Web services can be configured for integration of purchase orders, change purchase orders, cancel purchase orders, and purchase order ERP header status import. File-based integration transactions for Remittance, Expense Report and Payments can also be configured.

> **Note**
>
> These transactions can also be configured to integrate using files.

> **Note**
>
> When importing canceled and changed purchase order data, it is possible that the imported data contains duplicate entries of the same purchase order. You must ensure the duplicate entries are removed from your ERP system before processing the information.

To integrate your buying solution with a PeopleSoft backend system, the buying solution publishes a SOAP message for the required integration event. You need to use an EAI tool or application that can consume this SOAP message and send an acknowledgment back as an ID or error.

Import Purchase Order Close Status Task

The **Import Purchase Order Close Status** task imports the close status of the ERP purchase orders when the order is fully received and then closed for all actions.

When the `Application.Purchasing.PullCloseOrderStatusAckFromERP` and `Application.Purchasing.PushCloseOrderStatusToERP` parameters are enabled, the **Import Purchase Order Close Status** task reads data from the `PurchaseOrderCloseStatusHeaderImport.csv` and `PurchaseOrderCloseStatusItemImport.csv` files.

Following is an example of `PurchaseOrderCloseStatusItemImport.csv` for `EP17` containing 4 line items.

```
"UTF-8"
POLineNumber,UniqueName
1,EP17
2,EP17
3,EP17
4,EP17
```

When you run the task, ensure that all the line items are included in the `PurchaseOrderCloseStatusItemImport.csv` file. Else, an error message is displayed and purchase order line items are deleted if the `Application.Purchasing.AllowPurchaseOrderLineClose` parameter (set by SAP Ariba Support) is also enabled. For example, if EP17 doesn't have 2 lines in the file corresponding to POLineNumber 2 and 3, the following error message is displayed.

```
You cannot update order EP17 because POLineNumber value is missing for the line items
[2, 3].
```

Workflow of Failed ERP Purchase Order Export

ERP Purchase orders are those that are created in the buying solution and exported to an external ERP system. If failures occur, your buying solution attempts notification and correction.

There are mainly two reasons when a PO export to the external ERP may fail:

- Integration-related failure of the PO export:
  This is when the PO export fails. In such cases, an email is sent to the concerned groups notifying them about the failure. The buying solution then retries to send failed orders till the task is successful.
- Data related failure:
  This is when the export is successful, but the external ERP considers the purchase order as a failure due to data mismatch. In such cases, the ERP system sends a PushFail status message back to the buying solution. To resolve such issues, the buying solution deletes the failed order, and creates a new version of the original requisition and moves it to the **Composing** state.

Assuming your site is configured to process purchase orders as ERP orders, the following system behavior is observed when a PO export fails:

1. You create a requisition and submit it in the buying solution. The buying solution creates an ERP order, triggers an export to the external ERP, and moves this approvable into the **Ordering** state.
   The PO status remains in the **Ordering** state till the ERP system sends back a response to the buying solution.
   - If you are using the File Channel for communications, you must manually export the purchase orders.
   - If you are using the web services channel, all PO exports are performed in real time.
2. The buying solution automatically creates an Order Recipient object (internal to the buying solution, which is not visible on the user interface) for each such order that is exported and moves this object to the **Sending** state.
3. If the external ERP system responds with a PO export successful message, the buying solution moves the order into the Ordered state.
   If the PO export fails, the order remains in the **Ordering** state, but the Order Recipient object is moved to the **Failed** state.
4. Every time there is a failed PO export, the buying solution triggers an email notification that is sent to all users belonging to the **Purchasing Administrator** group.
5. The Failed Orders scheduled task (runs every six hours in the default configuration) then picks up all orders that have failed the PO export (Order Recipient object in **Failed** state) and tries to resend them to the external ERP system.

> **Note**
>
> The scheduled task continues to resend failed orders to the external ERP until successful. An email notification is sent the **Purchasing Administrator** group every time the purchase order export fails. To restrict the number of days the retries are attempted, your administrator can set the **Set number of days to resend failed ERP orders** (`Application.Purchasing.PermanentFailureRetryCountForERPOrder`) parameter. For information about the parameter, see Set number of days to resend failed ERP orders.

Related Information

Understanding Order Methods [page 142]

Exporting Purchase Orders

You can export purchase orders from your procurement solution.

Procedure

1. On the dashboard, click **Manage** > **Core Administration**.
2. On the **Welcome to Ariba Adminisrator** page click **Site Manager**.
3. Click **Data Import/Export** and click the **Export** tab.
4. Search for the **Export Purchase Orders** data export task and click **Export**.
5. Select a date range for the export by choosing dates in the **From** and **To** fields.
6. Click **Run** to export the data.
7. After the data has finished downloading, click **Open** to view the contents of the ZIP file or **Save** to save the file to your local hard drive.

Export Legal Fields in Purchase Order cXML sent to SAP Business Network

If the purchase order cXML that the SAP Ariba buying solution sends to SAP Business Network includes the company code, purchasing organization, and purchasing group information, users in SAP Business Network can search for purchase orders by these fields.

If your administrator has enabled the **Include new legal entity fields in purchase order cXML** (`Application.Procure.IncludeNewFieldsForLegalEntityInPOcXML`) parameter, the company code, purchasing organization, and purchasing group fields are passed in the purchase order cXML that the SAP Ariba buying solution sends to SAP Business Network. For information about the parameter, see Include new legal entity fields in purchase order cXML.

The following example shows a sample purchase order cXML snippet with the new fields that is sent to SAP Business Network.

Sample Code

```
<LegalEntity>
       <IdReference identifier="3000" domain="CompanyCode">
           <Description xml:lang="en_US">IDES&#032;US&#032;INC</Description>
```

```
        </IdReference>
    </LegalEntity>
    <OrganizationalUnit>
         <IdReference identifier="300" domain="PurchasingOrganization">
            <Description xml:lang="en_US">IDES&#032;NY</Description>
        </IdReference>
    </OrganizationalUnit>
    <OrganizationalUnit>
         <IdReference identifier="100" domain="PurchasingGroup">
            <Description xml:lang="en_US">Corporate&#032;Purchsng</Description>
        </IdReference>
    </OrganizationalUnit>
```

Purchase Order Data Files

When you run the `Export Purchase Orders` data export task, it creates a ZIP file consisting of purchase order data in CSV files.

The ZIP file contains the following three purchase order CSV files:

- `PurchaseOrderExport.csv`
- `PurchaseOrderDetailExport.csv`
- `PurchaseOrderSplitExport.csv`

Related Information

Purchase Order Header Data Reference [page 162]
Purchase Order Detail Data [page 164]
Purchase Order Split Accounting Data [page 167]

Purchase Order Header Data Reference

The `PurchaseOrderExport.csv` file contains header information for purchase orders and advanced ship notices.

The following table describes the columns in `PurchaseOrderExport.csv`.

Column	Description	Format
`Id`	The unique reference ID of the purchase order or advanced ship notice. This value is used to match purchase orders and advanced ship notices in the header, detail, and split accounting data files.	String (255)
`EventDate`	The date which this status change occurred.	Date
`EventStatus`	The new status for the purchase order or advanced ship notice.	String (255)

Column	Description	Format
EventType	The type of event (Ordered or Ordering).	String (255)
`Title`	The title of the purchase order.	String (128)
`PONumber`	The purchase order number.	String (255)
`Version`	The version of the purchase order.	Numeric
`ProcurementUnit`	The purchasing unit specified for the purchase order.	String (50)
`OriginatingSystem`	The originating system for the original requisition. This column is populated only by the requisition import feature. For information on the requisition import feature, see Import of Requisitions [page 58].	String (50)
`OriginatingSystemReferenceID`	The originating system reference ID of the original requisition. This column is populated only by the requisition import feature. For information on the requisition import feature, see Import of Requisitions [page 58].	String (50)
`Supplier`	The unique internal identifier of the supplier this purchase order was sent to or the ship notice was received from.	String (50)
`SupplierLocation`	The supplier location this order is being sent to.	String (50)
`PaymentTerms`	The unique internal identifier of the payment terms used for this purchase order. Payment terms specify the time period in which a supplier requests payment and any discounts that the supplier chooses to make available for early payment.	String (50)
`ShipNoticeDate`	The date on which this advanced ship notice was received.	Date
`ShipmentDate`	The shipment date as specified on the advanced ship notice.	Date
`DeliveryDate`	The delivery date as specified on the advanced ship notice.	Date
`ShippingCarrier`	The name of the shipping carrier.	String (80)
`TrackingNumber`	The shipment identifier.	String (255)
`ASNNumber`	The unique internal identifier of the advanced ship notice.	String (255)
`Comments`	Comments, including date and creator information. For example: `3/31/2008 12:08 PM Arnold Davis: comment`	String (2000)

Following is an example of `PurchaseOrderExport.csv`:

```
"UTF-8"


Id,EventDate,EventStatus,EventType,Title,PONumber,Version,ProcurementUnit,Originatin
gSystem,
OriginatingSystemReferenceID,Supplier,SupplierLocation,PaymentTerms,ShipNoticeDate,S
hipmentDate,
DeliveryDate,ASNNumber,ShippingCarrier,TrackingNumber,Comments

         "1","03/26/2008 20:20:50
-0700","Ordering","Ordering","sanitytestingwt-6","PO265","1",
"US002",,,"11","14","PT1",,,,,,,
```

```
          "3","03/26/2008 20:20:59
-0700","Ordered","Ordered","sanitytestingwt-6","PO265","1",
"US002",,,"11","14","PT1",,,,,,,
          "5","03/26/2008 20:21:08
-0700","Ordering","Ordering","sanitytestingwt-7","PO266","1",
"US002",,,"11","14","PT1",,,,,,,
          "6","03/26/2008 20:21:19
-0700","Ordered","Ordered","sanitytestingwt-7","PO266","1",
"US002",,,"11","14","PT1",,,,,,,
          "9","03/26/2008 20:21:30
-0700","Ordering","Ordering","sanitytestingwt-8","PO267","1",
"US002",,,"11","14","PT1",,,,,,,
```

Purchase Order Detail Data

The `PurchaseOrderDetailExport.csv` file contains line-item information for purchase orders and advance ship notices. It also contains details of taxes, charges, and discounts applied to the line items.

The following table describes the columns in the `PurchaseOrderDetailExport.csv` file.

Column	Description	Format
`Id`	The unique reference ID of the purchase order or advanced ship notice. This value is used to match purchase orders and advanced ship notices in the header, detail, and split accounting data files.	String (255)
`PONumber`	The purchase order number.	String (255)
`Version`	The version of the purchase order.	Numeric
`LineNumber`	The unique line number assigned to every material, tax, charge, and discount line item in the purchase order.	Numeric
`Description`	The description of the purchase order line. For standard lines, this is the item description. For tax lines, this is the name of the tax component. For charge lines, this is the charge type. For discount lines, it displays the text `Discount`.	String (2000)
`ExportLineItems_ReqId`	The unique requisition ID from which this PO was created.	String (50)
`RequisitionID`	The unique internal identifier of the requisition from which this purchase order was created.	String (50)
`ShortName`	A short version of the description of this item.	String (50)
`SupplierPartAuxID`	The supplier's auxiliary ID for this item.	String (255)
`SupplierPartNumber`	The supplier's part number for this item.	String (255)
`UnitOfMeasure`	The unit of measure for this item.	String (50)
`CommodityCode`	The system commodity code for this item.	String (100)
`UnitPrice`	The unit price for the item. For standard line items and discount lines, this displays the unit price of the item. For tax lines, this displays the calculated tax amount. For charge lines, this displays the charge amount.	Numeric
`UnitPrice_Currency`	The currency of the unit price for this item.	String (50)

Column	Description	Format
`UnitPrice_InBaseCurrency`	The price for this item, given in the default currency using the current exchange rate.	Numeric
`Quantity`	The quantity of this item.	Numeric
`Price`	The total price for this item. For standard line items, this displays the (`UnitPrice` x `Quantity`) value. For tax lines, this displays the calculated tax amount. For charge lines, this displays the charge amount. For discount lines, this displays the discount as a negative value.	Numeric
`Price_Currency`	The currency of the total price.	String (50)
`Price_InBaseCurrency`	The total price for this line specified in the default currency, using the current exchange rate.	Numeric
`BillingAddress`	The unique internal identifier of the billing address for this item.	String (50)
`ShippingAddress`	The unique internal identifier of the shipping address for this item. **Note** The export file doesn't contain the information entered in the **Street 2** and **Street 3** fields.	String (50)
`DeliverTo`	The person to whom this item should be delivered.	String (100)
`ERPCommodityCode`	The ERP-specific commodity code for this item.	String (100)
`NeedBy`	The need by date for this item.	Date
`NumberShipped`	The number shipped as specified in the advanced ship notice.	Numeric
`Comments`	Comments, including date and creator information. For example: 3/31/2008 12:08 PM Arnold Davis: `comment`	String (2000)
`ServicesLineItemDetails`	This column contains services procurement specific information. This information is grouped based on the following categories: • Labor • Marketing • Consulting • Generic • Print Each category has an individual cXML file that contains information specific to that category.	XML
`ExternalLineNumber`	This column value maps the maps the PR Line Item NumberInCollection to the PO Line Item NumberInCollection.	
`MASK_UNITPRICE`	Indicates whether the unit price of the item is flagged to be hidden from suppliers. When unit price is hidden, the amount value is also hidden from suppliers.	

Column	Description	Format
MASK_AMOUNT	Indicates whether the amount value for the item is hidden from suppliers depending on if the unit price of the item is flagged to be hidden from suppliers.	
PriceUnitQuantity	The quantity corresponding to the unit price of the item specified by the supplier. The unit price is based on the price unit quantity of the item.	Big Decimal
UnitConversion	The ratio used to convert the ordered unit to the price unit. The unit price is based on the price unit quantity and the price unit.	Big Decimal
PricingDescription	Any information for the advanced pricing details.	String (2000)
PriceUnit	The unit of measure for the unit price.	String (50)
ParentLineNumber	The line number of the line item on which the tax, charge, or discount has been applied.	Integer
LineType	The type of line item – Tax, Charge, or Discount. This field is blank for regular line items.	String (50)
LineTypeCategory	The category of the line item. This value is 1 for standard line items, 2 for tax lines, 4 for charge lines with out-of-the-box shipping charge, 8 for charge lines with out-of-the-box handling charge, 16 for charge lines with customer-configured charge types, and 32 for discount lines.	Integer
TaxCode	The tax code applied on the standard or charge line item. This value is blank for discount lines and tax lines.	String (50)
RealRequester	The name of the actual requester specified in the **On behalf Of** field on a requisition.	
TaxPercent	The percentage rate of the tax component for taxes applied as a percentage.	Big Decimal
TaxPerUnit	The per unit rate of the tax component for taxes applied per unit	Big Decimal
IsTaxDeductible	Specifies whether the tax component is deductible	Boolean
TaxableAmount_Amount	The amount on which the tax is applied. The abatement percentage is considered when calculating the amount.	Big Decimal
TaxableAmount_Amount_Currency	The currency in which the taxable amount is displayed.	String (50)
TaxableAmount_Amount_InBaseCurrency	The taxable amount converted to base currency.	Big Decimal

Following is an example of `PurchaseOrderDetailExport.csv`:

```
"UTF-8"
Id,LineNumber,Quantity,RequisitionRequester,ERPCommodityCode,SupplierPartAuxID,PONum
ber,Price,Price_Currency,UnitPrice,SupplierPartNumber,RequisitionID,DeliverTo,Compan
yCode,UnitPrice_Currency,Comments,ShortName,UnitOfMeasure,PurchaseGroup,ShippingAddr
ess,PurchaseOrg,Description,UnitPrice_InBaseCurrency,NumberShipped,CommodityCode,Ext
ernalLineNumber,Version,NeedBy,AccountCategory,Price_InBaseCurrency,BillingAddress,I
temCategory,IsComposite,ParentLineNumber,PriceUnitQuantity,UnitConversion,PricingDes
cription,PriceUnit,TaxableAmount_Amount_Currency,LineTypeCategory,IsTaxDeductible,Li
```

```
neType,TaxPerUnit,TaxCode,RealRequester,TaxableAmount_Amount_InBaseCurrency,TaxPerce
nt,TaxableAmount_Amount,ERSAllowed,MASK_UNITPRICE,MASK_AMOUNT
5046,1,10,adavis,12,,PO10,102.5,USD,10.25,AD961L 1,PR13,Arnold
Davis,3000,USD,,Adapter DB9F/MD6F Logitech
Trackman,EA,100,3300,3000,Adapter DB9F/MD6F Logitech
Trackman,10.25,,432118,1,1,,K,102.5,3000,M,No,0,,,,,,,,,,adavis,,,,No,No,No
5046,2,10,adavis,12,,PO10,59.5,USD,5.95,AD964,PR13,Arnold Davis,3000,USD,,Adapter
DB9M/MD6M Ser. to PS2 Mouse,EA,100,3300,3000,Adapter DB9M/MD6M Ser. to PS2
Mouse,5.95,,432118,2,1,,K,59.5,3000,M,No,0,,,,,,,,,,adavis,,,,No,No,No
5084,1,10,adavis,12,,112239,102.5,USD,10.25,AD961L 1,PR22,Arnold
Davis,3000,USD,,Adapter DB9F/MD6F Logitech
Trackman,EA,100,3300,3000,Adapter DB9F/MD6F Logitech
Trackman,10.25,,432118,1,1,,K,102.5,3000,M,No,0,,,,,,,,,I1,adavis,,,,No,Yes,Yes
5084,2,10,adavis,12,,112239,59.5,USD,5.95,AD964,PR22,Arnold Davis,3000,USD,,Adapter
DB9M/MD6M Ser. to PS2 Mouse,EA,100,3300,3000,Adapter DB9M/MD6M Ser. to PS2
Mouse,5.95,,432118,2,1,,K,59.5,3000,M,No,0,,,,,,,,,I1,adavis,,,,No,No,No
```

Accounting Columns in PurchaseOrderDetailExport.csv File for SAP ERP

In addition to the general columns in `PurchaseOrderDetailExport.csv`, it also includes columns of accounting information specific to SAP ERP.

Accounting information columns specific to SAP include:

Column	Format
`PurchaseOrg`	String (50)
`PurchaseGroup`	String (50)
`ItemCategory`	String (50)
`AccountCategory`	String (50)

Purchase Order Split Accounting Data

The `PurchaseOrderSplitExport.csv` file contains accounting information for purchase orders and advanced ship notices, including split accountings, if present.

The following table describes the columns that are common to PeopleSoft, SAP, and Generic ERP systems.

Column	Description	Format
`Id`	The unique reference ID of the purchase order or advanced ship notice. This value is used to match purchase orders and advanced ship notices in the header, detail, and split accounting data files.	String (255)
`PONumber`	The purchase order number.	String (10)
`Version`	The version of the purchase order.	Numeric
`LineNumber`	The number of this line item on the purchase order.	Numeric

Column	Description	Format
SplitNumber	The number of this split accounting on the purchase order.	Numeric
Quantity	The quantity for this split. This value is calculated if the split is specified by amount or percent.	Numeric
UnitOfMeasure	The unit of measure for the quantity.	String (50)
Amount	The amount for this split. This value is calculated if the split is specified by quantity or percent.	Numeric
Amount_Currency	The currency for the amount.	String (50)
Amount_InBaseCurrency	The amount for this split in the default currency.	Numeric

The following is an example of the PurchaseOrderSplitExport.csv file that shows the additional accounting fields that are included in the exported file for Generic ERP:

```
"UTF-8"


Id,PONumber,Version,LineNumber,SplitNumber,Quantity,UnitOfMeasure,Amount,Amount_Curr
ency,
Amount_InBaseCurrency,Account,Business_Unit,Cost_Center,Region,Sub_Account,Company,P
roject,
Product

"67","PO268","1","1","1","1","EA","50.0000","USD","50.0000","7752","US002","4000",,"
5009",
"02",,
"67","PO268","1","2","1","2","EA","100.0000","USD","100.0000","7752","US002","4000",
,"5009",
"02",,
"67","PO268","1","3","1","3","EA","150.0000","USD","150.0000","7752","US002","4000",
,"5009",
"02",,
"68","PO269","1","1","1","1","EA","50.0000","USD","50.0000","7752","US002","4000",,"
5009",
"02",,
"68","PO269","1","2","1","2","EA","100.0000","USD","100.0000","7752","US002","4000",
,"5009",
"02",,
```

ERP-Specific Split Accounting Columns

In addition to the columns generic to all ERP systems in PurchaseOrderSplitExport.csv, it includes columns for accounting information that is specific to your ERP system.

The following table lists the additional accounting columns for Generic:

Column	Format
Account	String (50)
Company	String (50)
Business_Unit	String (50)

Column	Format
Cost_Center	String (50)
Sub_Account	String (50)
Project	String (50)
Product	String (50)
Region	String (50)

The following table lists the additional accounting columns for PeopleSoft:

Column	Format
Account	String (50)
GLBusinessUnit	String (50)
Location	String (50)
Department	String (50)
Product	String (50)
StatisticsCode	String (50)

The following table lists the additional columns for SAP:

Column	Format
CostCenter	String (50)
GeneralLedger	String (50)
WBSElement	String (50)
Asset	String (50)
InternalOrder	String (50)

Purchase Order Balance Amount Data

Purchase order balance is the amount on a purchase order still remaining when that purchase order has been partially invoiced. This is the amount committed to be spent on the line items in that order minus the amount that has already been invoiced and reconciled for each such line item.

Based on the tolerances that you have configured, there may be header and/or line level exceptions that can occur on the amount that the supplier invoices for a purchase order. These exceptions must be handled or reconciled by authorized users. During the reconciliation process, you can do any of the following:

- Accept the amount invoiced by the supplier
- Dispute it and force the terms of the purchase order on the invoice
- Reject the invoice

After the reconciliation process is complete, the invoice is moved to the Reconciled state and sent out for payment.

To facilitate the purchase order balance amount to be displayed in the exported data, a new field `AmountLessReconciledBySplit` has been added to the Split Accounting class. This field calculates the reconciled amount minus the ordered amount.

> **Note**
>
> This field is not visible in the exported data by default. Contact SAP Ariba Customer Support to customize the Purchase Order Export integration event to be able to see this field in the exported data.

New Fields in Incremental Purchase Order Export

You can have the purchase order export configured such that any field in the purchase order page that undergoes a change is automatically added to the incremental purchase order export transaction data.

To enable this feature, contact SAP Ariba Customer Support and provide the field names that must be added to the corresponding parameter. When a field that is added to this parameter undergoes any change, the parameter triggers a new entry for this changed field to be inserted to the PO export transaction data.

A typical scenario to illustrate this addition would be when a purchase order having multiple splits is reconciled. After reconciliation, if the **PO Balance** field is added to the parameter, a new PO Balance entry (based on the various accounting splits) is inserted into the PO export transaction data.

You can also have your PO export configured to capture data when an order is closed or re-opened.

For example:

Consider a purchase order with amount of $100 is invoiced and ready to be reconciled. This order also has multiple split accounts. Before reconciliation, the PO export shows the PO Balance field value as $100.

This invoice is reconciled for $50 and the balance amount is split among the various accounts. But when an incremental PO export is done, the PO Balance should show the balance at each split accounting level.

As there is no change in the status of the PO itself (though some of the PO field values had changed), the PO export does not insert the changed information in the transaction data.

Optionally, if you want such transactional data to be included in your incremental PO export, you must contact SAP Ariba Customer Support to configure these fields appropriately.

> **Note**
>
> The fields that you want to add to your incremental PO export transactional data must be the purchase order header-level fields only (and not the line level fields).
>
> - The `PO Balance` entry is inserted into the `PurchaseOrderSplitExport.csv` file
> - The `Closed PO Status` column is inserted into the `PurchaseOrderExport.csv` file.

Resend Purchase Orders Stuck in Ordering or Canceling State

Purchase orders can be stuck in the ordering or canceled state due to various reasons while being sent or resent from SAP Ariba buying solutions to SAP Business Network or ERP.

Multiple retries are automatically attempted based on where the order is being sent:

- For orders being sent to SAP Business Network, retries are attempted for a day.
- For orders being sent to ERP, retries are attempted based on the number of days configured in the **Set number of days to resend failed ERP orders** (`Application.Purchasing.PermanentFailureRetryCountForERPOrder`) parameter.

If the retries are taking too long for an order, or if sending the order fails permanently, you can use the **Resend Order** button to manually initiate an attempt to resend. This button is displayed based on the number of hours specified in the **Set hours to enable resending of stuck purchase orders** (`Application.Purchasing.SetHoursToEnableResendingOfStuckPurchaseOrders`) parameter. For example, if the parameter is set to the recommended value 24, the button is displayed 24 hours after the purchase order is created, if the order is still being sent, or if the sending has permanently failed.

> **Note**
>
> - You must belong to the **Purchasing Approvable Resend Administrator** group to view and use the **Resend Order** button.
>   As an administrator, you must exercise caution when assigning users to the **Purchasing Approvable Resend Administrator** group in an environment where the **Restrict data visibility based on purchasing unit responsibilities** (`Application.Approvable.OrganizationalVisibilityPartitioning`) parameter is enabled. This is because the users in the group can search for approvable role assignments and view and resend all orders and receipts stuck in the **Ordering** status. This includes the orders and policies that don't belong to the purchasing units assigned to the users.
> - The **Resend Order** button isn't available for manual orders.
> - This feature is available for resending purchase orders using SAP Integration Suite, managed gateway for spend management and SAP Business Network or SAP Business Network. It is not available for resending purchase orders using CSV files or web services.

Prerequisites

Your administrator must configure the new parameter for the feature, **Set hours to enable resending of stuck purchase orders** (`Application.Purchasing.SetHoursToEnableResendingOfStuckPurchaseOrders`). See Set hours to enable resending of stuck purchase orders.

Related Information

Workflow of Failed ERP Purchase Order Export [page 160]

Resending Stuck Purchase Orders [page 172]
Set number of days to resend failed ERP orders
Set hours to enable resending of stuck purchase orders

Resending Stuck Purchase Orders

You can resend purchase orders that are stuck in the ordering or canceled state due to various reasons while being sent or resent from SAP Ariba buying solutions to SAP Business Network or ERP.

Prerequisites

- You must be assigned to the **Purchasing Approvable Resend Administrator** group to resend stuck purchase orders.
- Before resending the order, ensure that it does not exist on SAP Business Network, on any external ERP system, or on any other integration point. If the order already exists, a duplicate will be created.

Context

Use the **Resend Order** button to manually resend a purchase order that is stuck in the ordering or canceled state. This button is displayed only in the following scenario:

- The hours since the purchase order was created exceed the hours specified in the **Set hours to enable resending of stuck purchase orders** (`Application.Purchasing.SetHoursToEnableResendingOfStuckPurchaseOrders`) parameter.
- The purchase order is still being sent, or if sending has permanently failed.

> **Note**
>
> Clicking the **Resend Order** button only marks the order for resend and it is not resent immediately.

Procedure

1. On the dashboard, click **Manage** > **Purchase Orders**.
2. On the **Purchase Orders** page, enter search terms for the purchase order you want to resend, and click **Search**.
3. Click the **Order ID** link for the purchase order you want to resend.
4. Click **Resend Order**.
5. Add any comments you want to include, then click **OK**.

Related Information

Resend Purchase Orders Stuck in Ordering or Canceling State [page 171]
Set hours to enable resending of stuck purchase orders

Export of Financial Transaction Information

In SAP Ariba Buying and Invoicing and SAP Ariba Buying, the Financial Export feature allows you to export all transactional data to ERP systems to facilitate the creation of accrual and accounting liability entries as per your business requirements.

Transactions are counted when an order is made, the item is delivered, or the service is provided, regardless of when payment is actually made. In other words, expenses are counted when the goods or services are received, and income is counted when the sale occurs.

The financial export feature provides transaction information in CSV file format. The exported CSV file contains a variety of transaction information, such as tracking:

- The date and time when received and the quantity that was received against an order or contract.
- The date and time, and the quantity that was invoiced against an order, contract, or a non-PO invoice.

The exported CSV file also provides information on the object that triggered the transaction, the kind of transaction, along with the data that is required to perform the accrual calculations.

The buying solution provides two methods you can use to export and generate your accruals data:

- Generate Accruals uses the RNI prepackaged report.
- Financial Export exports all transactional data to ERP systems.

> **Note**
>
> Standard export data are enhanced periodically to include more fields. When added, new fields are always inserted at the end of the record in all exported CSV files. Customers that implement further integrations with these exported CSV files must ensure that their integrations are robust enough to ignore these newly added fields if they are not required.

Configuring Financial Export

The financial export feature is not enabled in the default configuration. Contact your SAP Ariba Customer Support to enable this feature on your site.

Financial Information in the Buying Solution [page 175]

Financial Data Extract Event Types [page 175]

Considerations for Exporting Financial Information [page 176]

Exporting Financial Information [page 177]

Loading Financial Extract into an ERP System Database [page 177]

Financial Data in FinancialExtractExport.csv [page 177]

Financial Information in the Buying Solution

The buying solution captures transaction details and posts them to an "export queue" whenever an event occurs that triggers a change to a relevant accumulator on an item in a purchase order, contract, or invoice.

The relevant accumulators are:

- Amount billed
- Amount cleared
- Amount accepted

The events that trigger changes to the relevant accumulators include:

- An item is received (the first approver approves a receipt).
- An invoice is received and an invoice reconciliation document is created.
- An invoice reconciliation document is manually matched to another purchase order or contract, or each line is manually matched to a different line of the purchase order or contract.
- An invoice reconciliation document is rejected or canceled.

Financial Data Extract Event Types

Every transaction that you perform is associated with a corresponding event type. You can view the event name that corresponds to each transaction when you perform a financial extract.

The following event types are supported:

Event Type	Trigger
Change Order	A change order action is performed, both on the old version and the new version of the purchase order.
Credit Memo Match (The credit memo event type is supported only at the line item level. Header level credit memo is not supported.)	• Credit memo gets matched to a PO • Credit memo gets into reconciling state for a non-PO Credit memo • A non-PO credit memo gets matched to a PO • A credit memo line gets manually matched to a PO line of the same PO or different PO
Credit Memo Unmatch	• A non-PO credit memo gets matched to a PO • On an old PO line, an invoice line gets manually matched to a different line of the same PO or a different PO
Credit Memo Reject	An invoice reconciliation gets rejected from a credit memo
Credit Memo Cancel	An invoice reconciliation gets cancelled for a credit memo
Receipt Match	• A PO is received • Editing of a receipt changes the Amount Accepted value
Receipt Reject	A receipt is rejected (denied)
Receipt Correction	A receipt is edited to accommodate negative receiving

Event Type	Trigger
ReceiptAdjusted	When an adjustment is made to the amount in the receipt by any of the approvers. Receipt adjustments are when the amount is accepted, modified, or denied.
Invoice Match	• An invoice gets matched to a PO • A non-PO invoice moves to the reconciling state • A non-PO invoice gets matched to a PO • On new PO line, an invoice line gets manually matched to a different line of the same PO or different PO
Invoice Unmatch	• A non-PO invoice gets matched to a PO • On an old PO line, an invoice line gets manually matched to a different line of the same PO or different PO
Invoice Reject	An invoice reconciliation gets rejected
Invoice Cancel	An invoice reconciliation gets cancelled

Considerations for Exporting Financial Information

When exporting financial information, be aware that not everything is considered for accrual calculation, and that calculation of accrual data is not supported.

Specifically, be aware of the following when exporting financial information:

- Taxes, shipment, handling charges, and discounts are not considered for accrual calculation.
- Purchasing card orders, charges based on accruals, expense report-based accruals, header-level credit memos, partial payments, and items on invoices that do not match an existing purchase order or contract are not included in accrual transactions.
- PO/Contract based invoices can have items in PO/Contract along with the items that are not part of Order/Contract. Accrual transaction will be considered for matched items in Order/Contract. Unmatched items are not considered.
- Calculation of accrual data is not supported. The financial data that is logged can be used for accrual and liability calculations. Programmatic linkage between the `FinancialExtractExport.csv` file and the external ERP system is not provided.

Exporting Financial Information

You can export financial information by running the `Export Financial Extract` data export task in Ariba Administrator. You can also export financial information by using the SAP Ariba data transfer tool.

Procedure

1. On the dashboard, click **Manage** > **Core Administration** .
2. On the **Welcome to Ariba Adminisrator** page click **Site Manager**.
3. Click **Data Import/Export** and click the **Export** tab.
4. Search for the **Export Financial Extract** data export task and click **Export**.
5. Select a date range for the export by choosing dates in the **From** and **To** fields. Ariba Administrator exports all financial information created during this time period. The first time you export the data, the **From** text box is empty.
6. Click **Run** to export the data.
7. After the data has finished downloading, click **Open** to view the contents of `FinancialExtractExport.zip`, or click **Save** to save the file to your local hard drive.

Loading Financial Extract into an ERP System Database

You can use the DB Connector to load accrual information into a supported ERP system database.

Supported databases include:

- Oracle
- DB2
- SQL Server

Financial Data in FinancialExtractExport.csv

The `Export Transaction Accruals` data export task exports accrual information in CSV format to the file `FinancialExtractExport.csv`. This file contains data common to all ERP systems, and there is accounting and split accounting information specific to each supported ERP system.

The following table describes the columns in the `FinancialExtractExport.csv` file that are common to PeopleSoft, SAP, and Generic ERPs.

Column Name	Description	Format
TransactionDate	Date on which transaction was posted.	Date
Supplier	Supplier name associated with the transaction.	String (128)
SupplierID	The unique internal identifier of the supplier.	String (50)
SupplierLocationID	The unique internal identifier of the supplier location.	String (50)
EventType	Event type that caused this transaction.	String (64)
TrxnDocNumber	The unique internal identifier of the transactional document that causes the change in accrual liability of referenced object. It can be a Receipt, an Invoice, or an InvoiceReconciliation document.	String (50)
TrxnDocLineNumber	Line item number on the transactional document.	Integer
TrxnDocLineUnitPrice	Unit price on the transactional document line item.	Big Decimal
TrxnDocLineQuantity	Quantity on the transactional document line item.	Big Decimal
TrxnReceivedAmount	Received amount of this transaction.	Big Decimal
TrxnInvoicedAmount	Invoiced amount of this transaction.	Big Decimal
RefDocCurrency	Currency of the reference document.	String (50)
RefDocNumber	Reference document number that is the context object on which this transaction occurred. This could be an Order or Invoice.	String (64)
RefDocLineNumber	Line item number on the reference document. For example, the line number of one item in the Order or Invoice.	Integer
RefDocSplitLineNumber	Split line number on the reference document.	Integer
RefDocLineUnitPrice	Unit price on the reference document line item.	Big Decimal
RefDocLineQuantity	Quantity on the reference document line item.	Big Decimal
RefDocLineAmount	Amount on the reference document line item.	Big Decimal
RefDocLineCommodityCode	Unique internal identifier of the commodity code on the reference document line item.	String (100)
OriginatingSystem	The name of the external system from which the reference document is imported.	String (50)
OriginatingSystemReferenceID	Unique internal identifier from the originating system.	String (50)
ProcurementUnit	The unique internal identifier of the purchasing unit associated with the reference document.	String (50)

In addition to the columns of common data in `FinancialExtractExport.csv`, it also includes columns for accounting information. These columns are specific to your ERP system. For more information about the columns specific to the ERP system, see:

- FinancialExtractExport.csv: Accounting Columns for PeopleSoft ERP [page 179]
- FinancialExtractExport.csv: Accounting Columns for SAP ERP [page 179]
- FinancialExtractExport.csv: Accounting Columns for Generic ERP [page 180]

FinancialExtractExport.csv: Accounting Columns for SAP ERP

A table lists the accounting columns in `FinancialExtractExport.csv` for SAP ERP.

Column	Format
`Account`	String (50)
`CostCenter`	String (50)
`GeneralLedger`	String (50)
`WBSElement`	String (50)
`Asset`	String (50)
`InternalOrder`	String (50)
`SAPSerialNumber`	String (100)
`SAPLimitSplitValue`	String (25)

FinancialExtractExport.csv: Accounting Columns for PeopleSoft ERP

A table lists the accounting columns in `FinancialExtractExport.csv` for PeopleSoft ERP.

Column	Format
`Account`	String (50)
`BusinessUnit`	String (50)
`GLBusinessUnit`	String (50)
`Location`	String (50)
`Department`	String (50)
`Product`	String (50)
`Project`	String (50)
`StatisticsCode`	String (50)
`PSProfileId`	String (10)
`BusinessUnitAM`	String (5)

FinancialExtractExport.csv: Accounting Columns for Generic ERP

A table lists the accounting columns in `FinancialExtractExport.csv` for Generic ERP.

Column	Format
`Account`	String (50)
`Company`	String (50)
`BusinessUnit`	String (50)
`CostCenter`	String (50)
`SubAccount`	String (50)
`Product`	String (50)
`Project`	String (50)
`Region`	String (50)
`AccountingCombination`	String (50)

Configuration of Messaging

The messaging feature allows users to initiate conversations from orders and invoices to interact with suppliers on SAP Business Network and other users in your organization to get information they require to process approvable documents.

To configure this feature, you must complete the following tasks:

1. Run the data import tasks (**Import Messaging Topics** and **Import Conversation Context Rules for Messaging**) to import conversation context rules and messaging topics.
2. Configure the messaging setup to map approvable documents with the appropriate conversation topics, users, user groups, and suppliers.

Related Information

Configuring the Messaging Setup [page 189]
Import Conversation Context Rules for Messaging Task [page 182]
Import Messaging Topics Task [page 181]

Import Messaging Topics Task

The **Import Messaging Topics** task imports predefined messaging topics used for conversation by the messaging service.

Users while initiating a conversation can choose one of the messaging topics to specify the subject of the conversation.

The **Import Messaging Topics** task uses the `MessagingTopicExport.csv` file. The following table describes the fields in the `MessagingTopicExport.csv` file:

Field name	Required	Type and description	Sample value
`UniqueName`	Yes	String. The unique internal identifier for a messaging topic.	QuantityIssue
`DisplayName`	Yes	Boolean. The user-visible name for the messaging topic.	Incorrect quantity

Field name	Required	Type and description	Sample value
`DefaultMessage`	No	String. The user-visible messaging topic that is displayed by default.	Incorrect quantity

Following is a sample of the `MessagingTopicExport.csv` file:

```
"UTF-8"
UniqueName,DisplayName,DefaultMessage
RequestForPartialCreditMemo,"Request for partial credit memo",
RequestForTotalCreditMemo,"Request for total credit memo",
PaymentIssue,"Payment issue",
PricingIssue,"Pricing issue",
TaxIssue,"Tax issue",
QuantityMismatch,"Quantity mismatch",
QualityIssue,"Quality issue",
MissingInvoiceData,"Missing invoice data",
ProductServiceQuestion,"Product/Service question",
RequestForSupportingDocuments,"Request for supporting documents",
Wrongquantity,"Wrong quantity",
Wrongdeliveryaddress,"Wrong delivery address",
Confirmation,"Confirmation",
PartialConfirmation,"Partial Confirmation",
Refuseddelivery,"Refused delivery",
```

Import Conversation Context Rules for Messaging Task

The **Import Conversation Context Rules for Messaging** task defines the approvable document types (for example, orders and invoices) for which messaging must to be enabled. In addition, this task maps each approvable document type with predefined topics, suppliers, user groups, and specific users who can participate in conversations related to such documents.

You can also customize the format of the subject line for invoices and orders. For example, you can set options that automatically append the order ID to a topic in the subject line for conversations related to orders.

The **Import Conversation Context Rules for Messaging** task uses the `CollaborationConfigurationDataPull.csv` file. The following table describes the fields in the `CollaborationConfigurationDataPull.csv` file:

Field name	Required	Type and description	Sample value
`UniqueName`	Yes	String. The unique internal identifier for the messaging context.	Invoice
`Name`	Yes	Boolean. The user-visible name for the messaging context.	Invoice

Field name	Required	Type and description	Sample value
`ConfigData`	Yes	String. The configuration data specific to the messaging context. The value of this field contains five parts: 1. **\"id\"**: The unique internal identifier for the messaging context. Set it to the same value as `UniqueName`. 2. **\"contextName\"**: The user-visible name for the messaging context. Set it to the same value as `Name`. 3. **\"contacts\"**: The users or user groups who can participate in conversations related to the defined messaging contexts and topics. 4. **\"topics\"**: The topics to be mapped with the messaging context. You can map multiple topics to one messaging context. Each topic is included in a pair of braces and contains an ID, name, and description. The topics added here must be defined in the `MessagingTopics.csv` file using the **Import Messaging Topics** task. 5. **\"defaultMessage\"**: The default types of information included in a messaging topic. For example, if you add `\"defaultMessage\":\"PO#: {0},Supplier: {1}, Amount:`	`"{ \"id\": \"INVOICE\", \"contextName\": \"Invoice\", \"contacts\": [{ \"id\": \"Invoice Manager\", \"name\": \"Invoice Manager\" }], \"topics\": [{ \"id\": \"RequestForTotalCreditMemo\", \"name\": \"Request for total credit memo\", \"desc\": \"Request for total credit memo\" }, { \"id\": \"MissingInvoiceData\", \"name\": \"Missing invoice data\", \"desc\": \"Missing invoice data\" }], \"defaultMessage\":\"Invoice# : {0},Supplier: {1}, Total Amount: {2}\" }"`

Field name	Required	Type and description	Sample value
		{2}\" for the messaging context Purchase Order in the CollaborationConfigurationDataPull.csv file, the messaging topics mapped to this messaging context will include three types of information: PO number, Supplier, and PO amount.	

Following is a sample of the CollaborationConfigurationDataPull.csv file:

```
"UTF-8"
UniqueName,Name,ConfigData
"INVOICE","Invoice","{
    \"id\": \"INVOICE\",
    \"contextName\": \"Invoice\",
    \"contacts\": [{
        \"id\": \"Invoice Manager\",
        \"name\": \"Invoice Manager\"
    },
    {
        \"id\": \"Purchasing Agent\",
        \"name\": \"Purchasing Agent\"
    }, {
        \"id\": \"Purchasing Manager\",
        \"name\": \"Purchasing Manager\"
    },
     {
        \"id\": \"Supplier\",
        \"name\": \"Supplier\"
    },{
        \"id\": \"Requester\",
        \"name\": \"Requester\"
    }],
    \"topics\": [{
        \"id\": \"RequestForTotalCreditMemo\",
        \"name\": \"Request for total credit memo\",
        \"desc\": \"Request for total credit memo\"
    },{
        \"id\": \"RequestForPartialCreditMemo\",
        \"name\": \"Request for partial credit memo\",
        \"desc\": \"Request for partial credit memo\"
    },{
        \"id\": \"Delay\",
        \"name\": \"Delay\",
        \"desc\": \"Delay\"
    },{
        \"id\": \"PaymentIssue\",
        \"name\": \"Payment Issue\",
        \"desc\": \"Payment issue\"
    },{
        \"id\": \"PricingIssue\",
        \"name\": \"Pricing issue\",
        \"desc\": \"Pricing issue\"
    }, {
        \"id\": \"QuantityMismatch\",
```

```
        \"name\": \"Quantity Mismatch\",
        \"desc\": \"Quantity Mismatch\"
    },  {
        \"id\": \"TaxIssue\",
        \"name\": \"Tax Issue\",
        \"desc\": \"Tax Issue\"
    }, {
        \"id\": \"QualityIssue\",
        \"name\": \"Quality issue\",
        \"desc\": \"Quality issue\"
    }, {
        \"id\": \"MissingInvoiceData\",
        \"name\": \"Missing invoice data\",
        \"desc\": \"Missing invoice data\"
    }, {
        \"id\": \"ProductServiceQuestion\",
        \"name\": \"Product/Service question\",
        \"desc\": \"Product/Service question\"
    },{
        \"id\": \"RequestForSupportingDocuments\",
        \"name\": \"Request for supporting documents\",
        \"desc\": \"Request for supporting documents\"
    }],
        \"defaultMessage\":\"Invoice# :{0},Supplier: {1}, Total Amount: {2}\"
}"
"INVOICE_LINE_ITEM","Invoice Line Item","{
    \"id\": \"INVOICE_LINE_ITEM\",
    \"contextName\": \"Invoice Line Item\",
    \"contacts\": [{
        \"id\": \"Purchasing Agent\",
        \"name\": \"Purchasing Agent\"
    }, {
        \"id\": \"Requester\",
        \"name\": \"Requester\"
    },{
        \"id\": \"Supplier\",
        \"name\": \"Supplier\"
    }],
    \"topics\": [{
        \"id\": \"PricingIssue\",
        \"name\": \"Pricing issue\",
        \"desc\": \"Pricing issue\"
    }, {
        \"id\": \"QuantityMismatch\",
        \"name\": \"Quantity Mismatch\",
        \"desc\": \"Quantity Mismatch\"
    },  {
        \"id\": \"TaxIssue\",
        \"name\": \"Tax Issue\",
        \"desc\": \"Tax Issue\"
    }, {
        \"id\": \"QualityIssue\",
        \"name\": \"Quality issue\",
        \"desc\": \"Quality issue\"
    }, {
        \"id\": \"MissingInvoiceData\",
        \"name\": \"Missing invoice data\",
        \"desc\": \"Missing invoice data\"
    }, {
        \"id\": \"ProductServiceQuestion\",
        \"name\": \"Product/Service question\",
        \"desc\": \"Product/Service question\"
    },{
        \"id\": \"RequestForSupportingDocuments\",
        \"name\": \"Request for supporting documents\",
        \"desc\": \"Request for supporting documents\"
    }],
```

```
     \"defaultMessage\":\"Invoice#:{0}, Supplier:{1}, Line Item: {2}, Qty: {3},
Price: {4}, Amount:{5}\"
}"
"INVOICE_LINE_ITEM_EXCEPTION","INVOICE Line Item Exception","{
    \"id\": \"INVOICE_LINE_ITEM_EXCEPTION\",
    \"contextName\": \"INVOICE Line Item Exception\",
    \"contacts\": [{
        \"id\": \"InvoiceExceptionGroup\",
        \"name\": \"Invoice Exception Group\"
    }, {
        \"id\": \"Invoice Manager\",
        \"name\": \"Invoice Manager\"
    }, {
        \"id\": \"Receiving Manager\",
        \"name\": \"Receiving Manager\"
    },
    {
        \"id\": \"Requester\",
        \"name\": \"Requester\"
    },{
        \"id\": \"Supplier\",
        \"name\": \"Supplier\"
    }],
    \"topics\": [{
        \"id\": \"PricingIssue\",
        \"name\": \"Pricing issue\",
        \"desc\": \"Pricing issue\"
    }, {
        \"id\": \"QuantityMismatch\",
        \"name\": \"Quantity Mismatch\",
        \"desc\": \"Quantity Mismatch\"
    },  {
        \"id\": \"TaxIssue\",
        \"name\": \"Tax Issue\",
        \"desc\": \"Tax Issue\"
    }, {
        \"id\": \"QualityIssue\",
        \"name\": \"Quality issue\",
        \"desc\": \"Quality issue\"
    }, {
        \"id\": \"MissingInvoiceData\",
        \"name\": \"Missing invoice data\",
        \"desc\": \"Missing invoice data\"
    }, {
        \"id\": \"ProductServiceQuestion\",
        \"name\": \"Product/Service question\",
        \"desc\": \"Product/Service question\"
    },{
        \"id\": \"RequestForSupportingDocuments\",
        \"name\": \"Request for supporting documents\",
        \"desc\": \"Request for supporting documents\"
    }],
    \"defaultMessage\":\"Invoice#: {0}, Supplier: {1}, Exception Type: {2}\"
}"
"IR","Invoice Reconciliation","{
    \"id\": \"IR\",
    \"contextName\": \"Invoice Reconciliation\",
    \"contacts\": [{
        \"id\": \"Invoice Manager\",
        \"name\": \"Invoice Manager\"
    },{
        \"id\": \"Purchasing Agent\",
        \"name\": \"Purchasing Agent\"
    }, {
        \"id\": \"Payment Manager\",
        \"name\": \"Payment Manager\"
    }, {
        \"id\": \"Requester\",
```

```
        \"name\": \"Requester\"
    }],
    \"topics\": [{
        \"id\": \"PricingIssue\",
        \"name\": \"Pricing issue\",
        \"desc\": \"Pricing issue\"
    }, {
        \"id\": \"QuantityMismatch\",
        \"name\": \"Quantity Mismatch\",
        \"desc\": \"Quantity Mismatch\"
    },  {
        \"id\": \"TaxIssue\",
        \"name\": \"Tax Issue\",
        \"desc\": \"Tax Issue\"
    }, {
        \"id\": \"QualityIssue\",
        \"name\": \"Quality issue\",
        \"desc\": \"Quality issue\"
    }, {
        \"id\": \"MissingInvoiceData\",
        \"name\": \"Missing invoice data\",
        \"desc\": \"Missing invoice data\"
    }, {
        \"id\": \"ProductServiceQuestion\",
        \"name\": \"Product/Service question\",
        \"desc\": \"Product/Service question\"
    },{
        \"id\": \"RequestForSupportingDocuments\",
        \"name\": \"Request for supporting documents\",
        \"desc\": \"Request for supporting documents\"
    }],
    \"desc\":\"Messaging configuration for Invoice Reconciliation\"
}"
"IR_LINE_ITEM","IR Line Item","{
    \"id\": \"IR_LINE_ITEM\",
    \"contextName\": \"IR Line Item\",
    \"contacts\": [{
        \"id\": \"Invoice Agent\",
        \"name\": \"Invoice Agent\"
    }, {
        \"id\": \"Purchasing Agent\",
        \"name\": \"Purchasing Agent\"
    }, {
        \"id\": \"Requester\",
        \"name\": \"Requester\"
    }],
    \"topics\": [{
        \"id\": \"PricingIssue\",
        \"name\": \"Pricing issue\",
        \"desc\": \"Pricing issue\"
    }, {
        \"id\": \"QuantityMismatch\",
        \"name\": \"Quantity Mismatch\",
        \"desc\": \"Quantity Mismatch\"
    },  {
        \"id\": \"TaxIssue\",
        \"name\": \"Tax Issue\",
        \"desc\": \"Tax Issue\"
    }, {
        \"id\": \"QualityIssue\",
        \"name\": \"Quality issue\",
        \"desc\": \"Quality issue\"
    }, {
        \"id\": \"MissingInvoiceData\",
        \"name\": \"Missing invoice data\",
        \"desc\": \"Missing invoice data\"
    }, {
        \"id\": \"ProductServiceQuestion\",
```

```
        \"name\": \"Product/Service question\",
        \"desc\": \"Product/Service question\"
    },{
        \"id\": \"RequestForSupportingDocuments\",
        \"name\": \"Request for supporting documents\",
        \"desc\": \"Request for supporting documents\"
    }],
    \"desc\":\"Messaging configuration for Invoice Reconciliation Line Item\"
}"
"PO","PO","{
    \"id\": \"PO\",
    \"contextName\": \"Purchase Order\",
    \"contacts\": [{
        \"id\": \"Purchasing  Manager\",
        \"name\": \"Purchasing Manager\"
    }, {
        \"id\": \"Supplier\",
        \"name\": \"Supplier\"
    }, {
        \"id\": \"Purchasing Agent\",
        \"name\": \"Purchasing Agent\"
    }, {
        \"id\": \"Requester\",
        \"name\": \"Requester\"
    }],
    \"topics\": [{
        \"id\": \"Confirmation\",
        \"name\": \"Confirmation\",
        \"desc\": \"Confirmation\"
    }, {
        \"id\": \"PartialConfirmation\",
        \"name\": \"Partial Confirmation\",
        \"desc\": \"Partial Confirmation\"
    }, {
        \"id\": \"RequestForSupportingDocuments\",
        \"name\": \"Request for supporting documents\",
        \"desc\": \"Request for supporting documents\"
    },{
        \"id\": \"Wrongquantity\",
        \"name\": \"Wrong quantity\",
        \"desc\": \"Wrong quantity\"
    }, {
        \"id\": \"Wrongdeliveryaddress\",
        \"name\": \"Wrong delivery address\",
        \"desc\": \"Wrong delivery address\"
    }, {
        \"id\": \"QualityIssue\",
        \"name\": \"Quality issue\",
        \"desc\": \"Quality issue\"
    }, {
        \"id\": \"Refuseddelivery\",
        \"name\": \"Refused delivery\",
        \"desc\": \"Refused delivery\"
    }],
    \"defaultMessage\":\"PO#:{0},Supplier: {1}, Amount: {2}\"
    }"
"PO_LINE_ITEM","PO Line Item","{
    \"id\": \"PO_LINE_ITEM\",
    \"contextName\": \"PO Line Item\",
    \"contacts\": [{
        \"id\": \"Purchasing  Manager\",
        \"name\": \"Purchasing Manager\"
    }, {
        \"id\": \"Supplier\",
        \"name\": \"Supplier\"
    }, {
        \"id\": \"Purchasing Agent\",
        \"name\": \"Purchasing Agent\"
```

```
    }, {
        \"id\": \"Receiving Manager\",
        \"name\": \"Receiving Manager\"
    },{
        \"id\": \"Requester\",
        \"name\": \"Requester\"
    }],
    \"topics\": [{
        \"id\": \"Confirmation\",
        \"name\": \"Confirmation\",
        \"desc\": \"Confirmation\"
    }, {
        \"id\": \"PartialConfirmation\",
        \"name\": \"Partial Confirmation\",
        \"desc\": \"Partial Confirmation\"
    }, {
        \"id\": \"RequestForSupportingDocuments\",
        \"name\": \"Request for supporting documents\",
        \"desc\": \"Request for supporting documents\"
    },{
        \"id\": \"Wrongquantity\",
        \"name\": \"Wrong quantity\",
        \"desc\": \"Wrong quantity\"
    }, {
        \"id\": \"Wrongdeliveryaddress\",
        \"name\": \"Wrong delivery address\",
        \"desc\": \"Wrong delivery address\"
    }, {
        \"id\": \"QualityIssue\",
        \"name\": \"Quality issue\",
        \"desc\": \"Quality issue\"
    }, {
        \"id\": \"Refuseddelivery\",
        \"name\": \"Refused delivery\",
        \"desc\": \"Refused delivery\"
    }],
    \"defaultMessage\": \"PO#:{0}, Supplier:{1}, Description: {2}, Qty: {3}, Amount:
{4}\"
}"
```

Configuring the Messaging Setup

You can configure the messaging setup in Ariba Administrator to enable users to participate in messaging conversations with suppliers and other users in the organization.

Procedure

1. On the dashboard, click **Manage** > **Core Administration**.
2. Click **Procure-to-Pay Manager** > **Messaging Setup**.

   The **Messaging Setup** page appears.
3. To configure users and user groups that can participate in messaging conversations for a document type (order or invoice), perform the following actions:

a. Click the edit icon for the document type under the **Contacts** column.
b. In the **Identify Contacts** window for the document type, choose the appropriate users and user groups and add them to the **Selected Contacts** list.
c. Click **Save**.
d. If required, repeat the steps to configure the contacts for the next document type.

4. To configure users and user groups that can participate in messaging conversations for a document type (order or invoice), perform the following actions:
   a. To configure topics that are applicable for a document type, click the edit icon for the document type under the **Topics** column.
   b. In the **Topics** window for the document type, choose the appropriate topics and add them to the **Selected Topics** list.
   c. Click **Save**.
   d. If required, repeat the steps on the **Messaging Setup** page to configure the topics for the next document type.

Configuration and Management of PCards

A PCard is a corporate credit card. SAP Ariba Buying and Invoicing and SAP Ariba Buying support PCard data import tasks.

The following PCard types are supported in the buying solutions from SAP Ariba:

Card Type	Description
Procurement	A corporate credit card used to purchase goods on behalf of your company. A procurement card can either be assigned to an individual, or can act as a ghosted card.
Combination	A corporate credit card that can be used either as an expense card or a procurement card.
Expense	A corporate credit card used to pay for business and travel expenses incurred on behalf of your company.
Accounting	This type of PCard is defined as an accounting-specific PCard associated with an entire department, purchasing unit (if being used), cost center, division, or for any accounting combination within an organization, where all users associated with that department, cost center, or division are able to use the same PCard for their purchases. **Note** You can use individual (Procurement) or combination cards to define accounting PCards. Expense PCards must not be used.
Unknown	This card type is intended for cards that are used outside of the buying solution. If you import charges associated with such a card, users cannot see or manipulate those charges in the user interface.

Members of the **PCard Manager** and **Procurement Manager** groups can manage PCards by using the PCard Manager workspace in Ariba Administrator.

About Ghosted PCards [page 192]

Creating a Ghosted PCard [page 192]

Individual PCards [page 193]

Accounting PCards [page 193]

PCard Charges [page 193]

PCards Import [page 194]

Accounting PCards [page 195]

Accounting Associations and Limits Import [page 199]

Accounting Associations and Limits Export [page 202]

Charge Tax Code Mappings Import [page 203]

Level 2 Charge Import [page 204]

Level 3 Charges [page 205]

Working with PCards [page 216]

Enabling Additional Users to Reconcile PCard Charge Exceptions [page 221]

PCard Adjustment Types [page 222]

Enhanced Charge Reconciliation for PCard Orders [page 223]

About Ghosted PCards

Ghosted PCcards are cards that are assigned to suppliers. There is no physical card, only a card number. Any orders sent to the supplier are billed to the ghosted card. A ghosted card can never be used for personal charges, even if you set the cardholder to a valid user name in the file `PCard.csv`.

When an order uses a PCard, the buying solution includes the card number on the order, indicating that the order is to be charged to that card number. The supplier charges the order to the PCard, and the issuing bank later sends a statement to the buying solution, listing all the transactions made with that PCard.

> **Note**
>
> When a manual order uses a PCard, a representative of your company must communicate PCard information to the supplier.

You associate a ghosted PCard with a supplier location by specifying the card number in the `GhostPCardNumber` column in the file `SupplierLocationSupplement.csv`. You can also associate a ghosted PCard with a remittance location by specifying the card number in the `RemittancePCardNumber` column in the file `RemittanceLocationDetails.csv`.

Alternatively, you can create a ghosted PCard from the user interface by going to the PCard Manager workspace.

Creating a Ghosted PCard

You can create a ghosted PCard from the PCard Manager workspace.

Prerequisites

Members of the **PCard Manager** and **Procurement Manager** groups can manage PCards. by using the PCard Manager workspace in Ariba Administrator.

Procedure

1. On the dashboard, click **Manage** > **Core Administration** .

2. Click **PCard Manager**.
3. Select **PCards** and click **Create New**. Alternatively, search for and select an existing PCard you want to convert to a ghosted PCard.
4. Click the corresponding **Actions** button and then click **Edit**.
5. On the page that displays, go to the **Supplier Locations Accepting Current Card** tab and click **Add/Remove**.
6. On the **Assign PCard to Supplier Locations** page, select the required supplier locations and click **Done**.

Individual PCards

Individual PCards are physical cards that are assigned to users. Only the cardholder can make purchases with the card, and that individual is responsible for resolving any issues that arise from the card's use.

You assign individual PCards to users in the file `UserPCardMap.csv`.

Related Information

Common Data Import and Administration for SAP Ariba Procurement Solutions

Accounting PCards

You use an import task to create accounting-specific PCards by associating purchasing units (if applicable) and accounting combinations, including cost center, department, account, company, or other custom accounting fields, to PCards.

The import task makes use of two CSV files. You must also configure PCard order limits based on commodity codes and purchasing units. The PCard limits are enforced against the item unit price and/or total line item amount.

When there are multiple PCards that can be used to process a purchase order, you can use ranking functionality to associate ranks with PCards, ensuring that the highest ranked card is used when the PCard order is generated.

PCard Charges

When your site uses PCards, your bank sends regular statements, showing the charges made on those cards.

Banks can send Level 3 (line-item level) charge information directly to suppliers in common data format (CDF). Suppliers can convert the CDF charge information to cXML format, and then send this information to your site for charge reconciliation through SAP Business Network.

Banks provide Level 2 (header level) charge in comma-separated value (CSV) format. You and your bank must agree on the set of columns for each statement file. You import Level 2 charge information into your site by running a data import task.

> **Note**
>
> If your site shares an SAP Business Network account with another site (it uses the Multi-Site Supplier Connectivity feature), Level 3 charge information is not supported.

PCards Import

There are four data import tasks related to PCards: `Importing Charge Tax Code Mappings`, `Importing Purchase Cards`, `Import Users`, and `Importing Level2Charge`.

The following table lists the data import tasks and CSV files that help you define PCard information and specify whether the data is required. Tasks appear in the recommended data import order.

Data Import Task	CSV Files Imported	Required
`Import Charge Tax Code Mapping`	`ChargeTaxCodes.csv`	Required if your site receives charges from SAP Business Network.
`Import Purchasing Cards`	`PCard.csv`	Required if users use any type of PCard.
`Import Users`	`User.csv` `UserPCardMap.csv`	Yes
`Import L2 Charge`	`Charge.csv`	Required if you import Level 2 charges.

The `Import PCards` data import task allows you to define your company's list of valid PCcards. It reads data from the CSV file named `PCard.csv`.

You specify the card number, expiration date, and cardholder name for each PCcard in `PCard.csv`. For ghosted PCcards, leave the cardholder blank, use a generic value such as "Corporate Card," or specify the name of an administrator. Following is an example of `PCard.csv`:

```
UTF8


cardHolder,cardNumber,expirationDate,personalLiability,cardType,GhostedOnly,BankId

Gene Halas,1111111111000001,12/31/2020,true,1,false,BuyerBankLocation-1
Gene Halas,2222222222000001,12/31/2020,true,2,false,BuyerBankLocation-1
Chad Noll,1111111111000002,12/31/2020,true,1,false,BuyerBankLocation-1
Chad Noll,2222222222000002,12/31/2020,true,2,false,BuyerBankLocation-1
```

The value in the `BankId` column links purchasing cards to buyer payment bank locations.

Related Information

Importing Charge Tax Code Mappings [page 203]

About Importing Purchasing Cards [page 194]
About Importing Level2 Charge [page 204]
Common Data Import and Administration for SAP Ariba Procurement Solutions
Invoicing Data Import and Administration Guide

Accounting PCards

Accounting PCards are not available in the default configuration. Configuring them is a two-step process involving actions by SAP Ariba Customer Support and by you.

To enable accounting PCards:

- Configure the new order method
  Have your Designated Support Contact log a service request so an SAP Ariba Customer Support representative follows up to complete the request.
  Ensure that this order method is created and ranked appropriately on your site.
- Import all required PCard data and configure amount limits using the following CSV files:
  - `PCardAccountingAssociation.csv`
  - `PCardAmountLimit.csv`

Related Information

About the Order Methods Java Class [page 144]
About Importing Accounting Associations and Limits [page 199]

Accounting PCards with Existing Pcard Order Methods

You can retain existing individual user PCard and supplier-specific PCard order methods when configuring the accounting PCard order method, but you must take extra care to preserve existing data.

Ensure that:

- From the existing set of PCards loaded into the buying solution, identify the cards (PCard IDs) that you want to use as accounting PCards and import only this PCard data using the `PCardAccountingAssociation.csv` file.
- Configure the `PCardAmountLimit.csv` file data considering only the accounting PCards. Amount limits that you configure using the `PCardAmountLimit.csv file` are enforced only on accounting PCards.
- The order method sequencing must be correctly configured.
  For example, if the preference is for the buying solution to process purchase orders using the accounting PCards, followed by individual PCards, and then the supplier-specific PCards, you must ensure that the corresponding order methods are sequenced in that order. Your Designated Support Contact will help you with this request.

Accounting PCard Ranking

If your site is configured to use accounting PCards, SAP Ariba now provides the ability to rank PCards.You can do this by specifying a rank value to every PCard entry in the `PCardAccountingAssociation.csv` file.

Ranking is a way of defining the order of precedence when there are multiple PCard matches, of which, one PCard must be selected to be associated with the PCard order. The highest rank is determined by the lowest integer (for example, in the sample extract below, the row ranked 1 is the highest and rank 4 is the lowest).

> **Note**
>
> When specifying PCard rank, ensure that the value is greater than zero and is always a positive integer. Negative values or alphabets are not supported. Also remember that an entry in this CSV file that does not have `Rank` value defined will cause the import task to `fail` .

PCard Ranking Scenarios

PCard ranking scenarios determine how PCards are associated with requisitions.

This example provides four PCard ranking scenarios that make use of the same accounting data to illustrate how ranking affects user-submitted requisitions. Each scenario is based on importing PCard accounting data with the following values and rank:

```
     ProcurementUnit,CostCenter,Requester,SupplierLocation,Contact,PCard,PCardRank


     Key,Key,Key,Key,Key,Value,Rank
US002,4100,*,*,*,1111111111000001,1
US002,4200,*,*,*,1111111111000003,3
US003,*,*,*,*,5555555555000001,2
US002,*,*,14,63,2222222222000001,3
US002,4100,cnoll,14,63,2222222222000001,2
*,*,*,*,*,5555555555000004,4
US004,*,arooney,*,*,1111111111000002,2
```

Case 1:

A user (for example, `cnoll`) associated to purchasing unit US002 creates and submits a requisition, with line item cost center `4100` , supplier location as `14` , and contact ID `63` .

The buying solution attempts to determine the best possible PCard match by looking up the `PCardAccountingAssociation.csv` file. In this example, it finds the following multiple entries that are matching the requirement:

`US002,4100,*,*,*,1111111111000001` , `1`

```
US002,*,*,14,63,2222222222000001,3
US002,4100,cnoll,14,63,2222222222000001,2
*,*,*,*,*,5555555555000004,4
```

These entries are matched for the following reasons:

- The first entry matches the purchasing unit and cost center values in the corresponding requisition.
- The second entry matches the purchasing unit `US002` , along with the supplier location and contact ID fields.
- The third entry matches all values in the corresponding requisition.
- The last entry contains wildcard characters for all `Key` fields, making it eligible.

In this scenario, the buying solution associates the PCard `1111111111000001` with the requisition and generates the PCard order because the highlighted entry has the highest rank value (lowest integer).

Case 2:

A user (for example, `ghalas`) associated to purchasing unit US003 creates and submits a requisition.

The buying solution attempts to determine the best possible PCard match by looking up the `PCardAccountingAssociation.csv` file. In this example, it finds the following multiple entries that are matching the requirement:

```
US003,*,*,*,*,5555555555000001,2

*,*,*,*,*,
5555555555000004,4
```

These entries are matched for the following reasons:

- The first PCard entry is associated with purchasing unit US003 (same as the requester)
- The second entry contains wildcard characters for all `Key` fields, making it eligible.

In this scenario, the buying solution associates the PCard 5555555555000001 with the requisition and generates the PCard order because the highlighted entry is ranked higher (2) between the two matching entries.

Case 3:

A user (for example, `arooney`) associated to purchasing unit US004 creates and submits a requisition.

The buying solution attempts to determine the best possible PCard match by looking up the `PCardAccountingAssociation.csv` file. In this example, it finds the following multiple entries that are matching the requirement:

```
US004,*,arooney,*,*,1111111111000002,2
*,*,*,*,*,5555555555000004,4
```

These entries are matched for the following reasons:

- The first PCard entry is a user-specific PCard associated to `arooney` , also belonging to the purchasing unit US004
- The second entry contains wildcard characters for all `Key` fields, making it eligible.

In this scenario, the buying solution associates the PCard `1111111111000002` with the requisition and generates the PCard order because the highlighted entry is associated to the requester (`arooney`) and is also ranked higher.

Case 4:

A user (for example, `adavis`) associated to purchasing unit US004 creates and submits a requisition.

The buying solution attempts to determine the best possible PCard match by looking up the `PCardAccountingAssociation.csv` file. In this example, it finds only one entry that matches the requirement:

```
*,*,*,*,*,5555555555000004,4
```

This entry is matched because this is the only entry that is applicable to the purchasing unit to which the user is associated, because it contains wildcard characters for all Key values.

Accounting PCard Selection for Split Lines

If a requisition line has multiple accounting split lines, the buying solution determines the appropriate PCard that must be associated with these splits by validating only one of the split lines during processing.

The matching PCard entry for that particular split line will then be applied to all other split lines, irrespective of the different accounting combinations that may be specified on the other split lines.

For example, if you create a requisition PR543, with line item L1, and choose to split it into three accounting lines, the buying solution randomly picks one of these three splits to determine an appropriate PCard to be associated, and applies the same PCard to the remaining splits to generate a single PCard order, irrespective of the accounting combinations that are specified on these split lines.

PCard Considerations

Be aware that there are a number of important considerations to keep in mind when working with PCards.

- Expense PCards must not be included in the `PCardAccountingAssociation.csv` file.
- If your buying solution finds two matching PCard entries in the `PCardAccountingAssociation.csv` file having the same rank, then the first PCard entry is selected.
- When importing the accounting association data, if there are instances where one user (requester) is associated with different PCards (but with the same accounting combinations and the same rank), then your buying solution will import only one record and discard other similar entries as duplicates.

  > **Note**
  >
  > You cannot associate more than one PCard with a single user with the same accounting combinations and rank.

- The requisition line item accounting combination must match at least one of the PCard accounting associations that you have loaded into your buying solution to process the resulting order as a PCard order.

  > **Note**
  >
  > If none of the accounting combinations match, then the requisition is processed according to the order method that has been configured on your site.

- The PCards that get matched must be valid cards.

- Configuring PCard amount limits is mandatory.
- PCard amount limits are associated with commodity codes and purchasing units (if used). Commodity Code is a mandatory field for the PCard-to-item match.

> **Note**
>
> When you configure PCard limits and associate them with a commodity code and purchasing unit (PU), these limits are automatically applicable to that commodity code/procurement unit and all of its children in the hierarchy.
>
> For example, if the UNSPSC code (commodity code) for wooden pencils is 44121706, it can be broken down into the following categories:
>
> 44 Office equipment, accessories, and supplies
> 12 Office supplies
> 17 Writing instruments
> 06 Wooden pencils
>
> Now, if you specify the commodity code value as 44 in your `PCardAmountLimit.csv` file, then the limits you configure for this commodity code will be applicable to all the child commodity codes (categories) in this hierarchy (categories as shown in the illustration above).
>
> Similarly, if your site is configured to use purchasing units, the limits that you specify for a particular PU will be applicable to that PU and all of its child PUs in the hierarchy.

- PCard limits can be loaded by specifying either the `PriceLimit` or `AmountLimit` value. When only one of these field values is blank, your buying solution does not validate the requisition line against that field.
- PCard limits can be loaded without specifying values for either `PriceLimit` or `AmountLimit` value. When both these values are blank, your buying solution replaces the blanks with 0 (zero). In such instances, your buying solution processes the resulting order based on the ordering method that has been configured on your site (whether as a PCard order or any other).
- During a change order that results in the creation os a V2 (version 2), if the originally used PCard has been rendered invalid, your buying solution does not attempt to match a different PCard to process this partial order. Such orders will remain in Ordering.

Accounting Associations and Limits Import

As a member of the **Customer Administrator** or **PCard Manager** group, you can configure accounting PCards by creating associations between PCards and the required accounting combinations.

Related Information

Configuring PCard Accounting Associations [page 200]
Configuring PCard Limits [page 201]

Configuring PCard Accounting Associations

If you are a member of the **Customer Administrator** or **PCard Manager** group, you can configure PCard accounting associations.

Prerequisites

The `PCardAccountingAssociation.csv` file must already exist.

- All fields in this CSV file are mandatory and cannot be blank or null values
- All " `Key` " fields must have valid values or replaced with the wildcard character "*"
- The " `Value` " field (`PCard`) value must be the `PCard ID`
- The `Rank` value must be a positive integer that is greater than zero

Procedure

1. From the dashboard, click **Manage** > **Core Administration**.
2. Click **PCard Manager**.
3. Click **PCard Association**.

   Alternatively, you can access the **PCard Association** import task from **Manage** > **Core Administration**, click **Site Manager**, then click **Data Import/Export Tasks** and search for **PCard Associations**.
4. On the **PCard Association** page, click **Browse** in the **Upload PCard Accounting Associations** section, and navigate to select the `PCardAccountingAssociation.csv` file.
5. Click **Upload**.

Sample PCardAccountingAssociation.csv File

```
ProcurementUnit,CostCenter,Requester,SupplierLocation,Contact,PCard,PCardRank
Key,Key,Key,Key,Key,Value,Rank
US002,4100,*,*,*,1111111111000001,1
US002,4200,*,*,*,1111111111000003,3
US002,*,ghalas,*,*,1111111111000002,2
US002,*,cnoll,*,*,5555555555000000,2
US002,*,*,14,63,2222222222000001,1
US002,*,*,13,13,4444444444000001,3
US002,*,*,*,*,5555555555000001,4
```

Configuring PCard Limits

If you are a member of the **Customer Administrator** or **PCard Manager** group, you can configure PCard limits.

Prerequisites

The `PCardAmountLimit.csv` file must already exist.

Procedure

1. From the dashboard, click **Manage** > **Core Administration**.
2. Click **PCard Manager**.
3. Click **PCard Association**.

   Alternatively, you can access the **PCard Association** import task from **Manage** > **Core Administration**, click **Site Manager**, then click **Data Import/Export Tasks** and search for **PCard Associations**.
4. On the **PCard Association** page, click **Browse** in the **Upload PCard Limits** section, and navigate to select the `PCardAmountLimit.csv` file.
5. Click **Upload**.

Sample PCardAmountLimit.csv File

```
Domain,CommodityCode,ProcurementUnit,PriceLimit,AmountLimit,Currency
Key,Key,Key,Value,Value,Value
unspsc,53,All,0,200,USD
unspsc,531016,US002,50,100,USD
unspsc,4216,US002,200,2000,USD
```

If the `PCardAmountLimit.csv` file contains blank or incorrect values, the buying solution handles them as follows:

- In the `Domain` , `CommodityCode` and `ProcurementUnit` columns, blank and incorrect values are treated as null values.
- In the `PriceLimit` and `AmountLimit` columns, blank and incorrect values are defaulted to $0.
- In the `Currency` column, blank and incorrect values are defaulted to the default currency for the site.

Accounting Associations and Limits Export

As a member of the **Customer Administrator** or the **PCard Manager** group, you can export accounting associations and limits.

Related Information

Exporting PCard Accounting Associations [page 202]
Exporting PCard Limits [page 203]

Exporting PCard Accounting Associations

If you are a member of the **Customer Administrator** or **PCard Manager** groups, you can export accounting associations and limits.

Procedure

1. On the dashboard, click **Manage** > **Core Administration**.
2. Click **PCard Manager**.
3. Click **Data Import/Export Tasks**.

   Alternatively, you can access the export task from **Manage** > **Core Administration**, click **Site Manager**, and then click **Data Import/Export Tasks**.
4. Click the **Export** tab.
5. Search for**Export PCard Accounting Associations** and click**Export**.

   The **Export PCard Accounting Associations** window opens.
6. Click **OK** or you may choose to open or save the `ExportPCardAccountingAssociation.csv` file.

Sample ExportPCardAccountingAssociations.csv File

```
ProcurementUnit,CostCenter,Requester,SupplierLocation,Contact,PCard,PCardRank
Key,Key,Key,Key,Key,Value,Rank
All,*,rpcuser1,*,*,5.25253E+15,1
All,*,rpcuser2,*,*,6.46465E+15,1
```

Exporting PCard Limits

Members of the **Customer Administrator** or **PCard Manager** groups can export accounting associations and limits.

Procedure

1. On the dashboard, click **Manage** > **Core Administration**.
2. Click **PCard Manager**.
3. Click **Data Import/Export Tasks**.

   Alternatively, you can access the export task from **Manage** > **Core Administration**, click **Site Manager**, and then click **Data Import/Export Tasks**.
4. Click the **Export** tab.
5. Search for **Export PCard Limits** and click **Export**.

   The **Export PCard Limits** window opens.
6. Click **OK** and you may choose to open or save the `PCardLimitsExport.csv` file.

Sample PCardLimitsExport.csv File

```
Domain,CommodityCode,ProcurementUnit,PriceLimit,AmountLimit,Currency
Key,Key,Key,Value,Value,Value
unspsc,42,All,10,1000,USD
unspsc,53,All,0,200,USD
unspsc,531016,US002,50,100,USD
```

Charge Tax Code Mappings Import

The **Import Charge Tax Code Mapping** data import task maps the common tax names found on charge statements to procurement line types.

Following is an example of `ChargeTaxCodes.csv`:

```
UTF8

     Domain,TaxCode,ProcureTypeUniqueName

MasterCard,,SalesTax
MasterCard,00,SalesTax
MasterCard,01,SalesTax
MasterCard,02,SalesTax
MasterCard,03,SalesTax
MasterCard,04,SalesTax
```

```
MasterCard,05,SalesTax
```

The following table describes the tax-related procurement line types in the default configuration:

Internal Identifier	Description
SalesTax	(SAP and Generic only) Sales tax. A tax that is imposed on retail sales of tangible personal property and certain services.
UseTax	(SAP and Generic only) Use tax. A tax that is imposed on the use, storage, or consumption of tangible personal property.
SUTTax	(PeopleSoft only) Sales/use tax. Sales tax is a tax that is imposed on retail sales of tangible personal property and certain services. Use tax is a tax that is imposed on the use, storage, or consumption of tangible personal property.
VATTax	Value-added tax (VAT). A general consumption tax that is assessed on the value added to goods and services at each stage in the production and distribution chain.
GSTTax	Goods and services tax (GST). A value-added tax on the sale of goods and services that exists in a number of countries/regions.
PSTTax	Provincial sales tax (PST). A tax imposed on the consumers of goods and services in Canadian provinces.
HSTTax	Harmonized sales tax. A single harmonized value-added tax that replaced the provincial sales taxes and the federal goods and services taxes in Nova Scotia, New Brunswick, Newfoundland, and Labrador.
QSTTax	Quebec provincial sales tax (QST). A sales tax imposed on retail sales in Quebec.

Related Information

Tax and Charge Configuration

Level 2 Charge Import

You import level 2 (header level) charge information by running the `Import L2 Charge` data import task.

The `Import Level2 Charge` data import task imports charge statements in CSV format. It reads data from the CSV file named `Charge.csv`. Following is an example of `Charge.csv`:

```
UTF8
LocalTaxIdentified,City,OrderNumber,Description,LocalTax,PCardNumber,MerchantName,
BankReferenceNumber,ExpenseCode,Amount,TransactionDate,AmountCurrency,MerchantId,Loc
alTaxCurrency
0,,DO10,,2222222222000000,Office Supplies,,1,,150,,USD,1,USD
```

> **Note**
>
> While the `LocalTaxIdentified` field is included as part of the charges data that is imported, this field value is not considered during reconciliation.

> **Note**
>
> The charges that you load, using this CSV file are not matched with the purchase orders in real time. These charges are matched only after the corresponding scheduled task is run (scheduled to run every day at 2:00 AM Pacific Time, in the default configuration).

Automating Level 2 Charge Importing

You can automate Level 2 charge importing by using the SAP Ariba data transfer tool.

Procedure

1. Name the CSV file that contains your Level 2 charge data `Level2Charge.csv` (not `Charge.csv`).
2. Include the `Level2Charge.csv` file in the ZIP file for the Import Batch Data data import task.
3. Run the **Import Batch Data** data import task from the SAP Ariba data transfer tool.

Level 3 Charges

Level 3 charges come into the buying solution in cXML formats from SAP Business Network, and contain information on the purchasing card orders, including line item, shipping, tax, and supplier details.

> **Note**
>
> The reconciliation of all level 3 charges is handled in the buying solution at the order header level and not at the line-item level. Line level exceptions are not validated.

Related Information

CDF to cXML Mapping [page 206]
Examples for Understanding CDF to cXML Mappings [page 207]
Reconciliation of Level 3 Charges [page 214]

CDF to cXML Mapping

A CDF file received from MasterCard is converted to a cXML format in SAP Business Network before it is sent to the buying solution.

The following table lists some of the key tags, subtags, and attributes of a CDF file and their corresponding mappings in the cXML file that is generated.

CDF			cXML		
Parent Tag	**Subtag**	**Attribute**	**Section**	**Subtag**	**Attribute**
TransmissionHeader_1000	TransmissionRecordHeader	`ProcessorNum`	ChargeVersion		`processorNumber`
		`MasterCardFileSequenceNum`			`sequenceNumber`
IssuerEntity	CorporateEntity				
	AccountEntity	`AccountNumber`	AccountActivity		`accountNumber`
AccountEntity	FinancialTransactionEntity				
FinancialTransactionEntity	LineitemDetailEntity	`LineItemNum`	DetailLineItem		`lineItemNumber`
		`DateOfPurchase`	DetailLineItem		`purchaseDate`
		`CustomerCode`	DetailLineItem		`customerCode`
	ProductCode	`ProductCode`	ProductCode		`Domain`
		`ItemDescription`		UnitRate	`Description`
		`ItemQuantity`		UnitRate	`quantity`
		`UnitPrice`		UnitRate	`Money`
		`ItemUnitOfMeasurementType`		UnitRate	`UnitOfMeasure`
		`LineItemTotalAmount`	TaxableAmount		`Money`
		`POLineNum`	DetailLineItem		`poLineNumber`
		`TaxLocation`	TaxDetail	TaxLocation	
	TaxAddendum_5300	`TaxAmount`	TaxDetail	TaxAmount	`Money`
		`TaxRate`	TaxDetail	TaxableAmount	`Money`

CDF			cXML		
Parent Tag	Subtag	Attribute	Section	Subtag	Attribute
	FinancialTransaction_5000	`AmountInOriginalCurrency`		TotalAmountInOriginalCurrency	`Money`
		`AmountInPostedCurrency`		TotalAmountInPostedCurrency	`Money`
		`AmountInBillingCurrency`			`TotalAmountInBillingCurrency`
		`FreightAmount`		ShippingAmount	`Money`
		`TransactionDiscountAmount`		DiscountAmount	`Money`
		`TotalTaxAmount`		TaxAmount	`Money`
		`BanknetReferenceNum`	L3ChargeFileRequest	ChargeSummary	`bankReferenceNum`
		`CardAcceptorVATNum`			`buyerVATID`
		`TransactionDate`	ChargeSummary		`transactionDate`

Examples for Understanding CDF to cXML Mappings

Examples can help you understand how purchase order charge values are specified in the CDF file, and how they are mapped in the cXML file that is generated by SAP Business Network.

Related Information

Example 1: Purchase Order Charge with Header Level Details: Tax, Shipping, and Discount Values [page 208]
Example 2: Purchase Order Charge with Line Level Details: Tax, Shipping, and Discount Values [page 211]

Example 1: Purchase Order Charge with Header Level Details: Tax, Shipping, and Discount Values

The first example of CDF to cXML mappings consists of three pieces of data: a summary of the purchase order and charge details, a CDF file, and a corresponding cXML file.

The purchase order has the following details:

- PO ID: PCO17
- Line Items: 2
  - LI 1: Quantity 20 at Unit Price of $6.67
  - LI 2: Quantity 8 at Unit Price $73.00
- Total Ordered: $717.40

Charge Details:

- Total Tax: $3.00
- Discount: -$9.00
- Shipping: $11.00
- Subtotal: $1,023.00

Total Amount: $1,026.00

CDF Sample

The charge for this order comes into SAP Business Network in the CDF format (from MasterCard). A sample CDF format (header level only) for the above charge looks similar to this:

> **Note**
>
> The below extract is not in its entirety. Some of the tags have been deleted to retain only the tags that provide the information required to substantiate the example.

```
     <TransmissionHeader_1000 >

<TransactionDate >2009-06-08</TransactionDate>
            <ProcessingDate >2009-06-10</ProcessingDate>
            <ApprovalCode >017808</ApprovalCode>
            <BanknetReferenceNum >pady-21oct10-4</BanknetReferenceNum>
            <CardAcceptorTerminalId >76868453</CardAcceptorTerminalId>
            <DebitOrCreditIndicator >D</DebitOrCreditIndicator>
            <
     AmountInOriginalCurrency
     CurrencyCode = "840" CurrencyExponent = "4" CurrencySign = "D"
>0000000010260000</AmountInOriginalCurrency>
            <OriginalCurrencyCode >840</OriginalCurrencyCode>
            <AmountInPostedCurrency CurrencyCode = "840" CurrencyExponent = "4"
CurrencySign = "D" >0000000010260000</AmountInPostedCurrency>
            <PostedCurrencyCode >840</PostedCurrencyCode>
            <PostedCurrencyConversionDate >09161</PostedCurrencyConversionDate>
            <PostedConversionRate >1.000000</PostedConversionRate>
            <AmountInBillingCurrency CurrencyCode = "840" CurrencyExponent = "4"
CurrencySign = "D" >0000000010260000</AmountInBillingCurrency>
            <BillingCurrencyCode >840</BillingCurrencyCode>
            <BillingCurrencyConversionDate >09161</BillingCurrencyConversionDate>
            <BillingConversionRate >1.000000</BillingConversionRate>
            <CustomerCode >PCO17</CustomerCode>
```

```
<
     FreightAmount
     CurrencyCode = "840" CurrencyExponent = "4" CurrencySign = "D"
>0000000000110000</FreightAmount>
            <
     TransactionDiscountAmount
     CurrencyCode = "840" CurrencyExponent = "4" CurrencySign = "D"
>0000000000090000</TransactionDiscountAmount>
            <TransactionDiscountType>VALCUST</TransactionDiscountType>
            <TotalTaxAmount CurrencyCode = "840" CurrencyExponent = "4"
CurrencySign = "D" >0000000000030000</TotalTaxAmount>
</TaxAddendum_5300 >

     <LineItemDetailEntity >

                         <CorporateCardLineItemDetail_5010 >

     <LineItemNum >0001</LineItemNum>

         <ItemDescription >item-1</ItemDescription>
              <ItemQuantity Exponent = "4" >0000000000200000</ItemQuantity>
              <ItemUnitOfMeasurementType >EA</ItemUnitOfMeasurementType>
<UnitPrice CurrencyCode = "840" CurrencyExponent = "4" CurrencySign = "D"
>0000000000066700</UnitPrice>
<LineItemTotalAmount CurrencyCode = "840" CurrencyExponent = "4" CurrencySign = "D"
>0000000001334000</LineItemTotalAmount>
              <POLineNum >0008</POLineNum>
            </CorporateCardLineItemDetail_5010 >

     </LineItemDetailEntity >



    <CorporateCardLineItemDetail_5010 >

     <LineItemNum >0002</LineItemNum>

<ItemDescription >item-2</ItemDescription>
                              <ItemQuantity Exponent = "4" >0000000000080000</
ItemQuantity>
                              <ItemUnitOfMeasurementType >EA</
ItemUnitOfMeasurementType>
                              <UnitPrice CurrencyCode = "840" CurrencyExponent =
"4" CurrencySign = "D" >0000000000730000</UnitPrice>
<LineItemTotalAmount CurrencyCode = "840" CurrencyExponent = "4" CurrencySign = "D"
>0000000005840000</LineItemTotalAmount>
                              <POLineNum >0002</POLineNum>
            </CorporateCardLineItemDetail_5010 >
  </TransmissionTrailer_9999 >
</CDFTransmissionFile>
```

cXML Sample

This CDF file is converted to a cXML format by SAP Business Network and sent to the buying solution. A sample cXML file format corresponding to the above CDF is here:

```
<L3ChargeFileRequest>
<
     ChargeSummary
     transactionDate="2009-06-08" conversionRate="1.000000"
bankReferenceNumber="cnoll-21oct10-3">
<TotalAmountInPostedCurrency>
<Money currency="USD">1026.0000</Money>
</TotalAmountInPostedCurrency>
<Total><Money currency="USD">1026.0000</Money></Total>
<TotalAmountInBillingCurrency>
<Money currency="USD">1026.0000</Money>
```

```
</TotalAmountInBillingCurrency>
<
     TaxDetail
     category="01" percentageRate="0">
<TaxAmount><Money currency="USD">3.0000</Money></TaxAmount>
</TaxDetail>
<
     ShippingAmount
    >
<Money currency="USD">11.0000</Money>
</ShippingAmount>
<
     DiscountAmount
    >
<Money currency="USD">9.0000</Money>
</DiscountAmount>
<OrderIDInfo orderID="PCO17"></OrderIDInfo>
<<AddendumItem>
<DetailAddendum>
<BasicAddendum>
<
     DetailLineItem lineItemNumber="0001"
     subLineItemNumber="0000" purchaseDate="2009-06-08" customerCode="PCO17"
poLineNumber="0001">
<Rate quantity="20.0000">
<Total>
<Money currency="USD">133.4000</Money>
</Total>
<UnitRate>
<Money currency="USD">6.6700</Money>
<UnitOfMeasure>EA</UnitOfMeasure>
</UnitRate>
</DetailLineItem>
<BasicAddendum>
<
     DetailLineItem lineItemNumber="0002
    " subLineItemNumber="0000" purchaseDate="2009-06-08" customerCode="PCO17"
poLineNumber="0002">
<Rate quantity="8.0000">
<Total><Money currency="USD">584.0000</Money></Total>
<UnitRate>
<Money currency="USD">73.0000</Money>
<UnitOfMeasure>EA</UnitOfMeasure>
</UnitRate>
</DetailAddendum>
</AddendumItem>
</ChargeTransaction>
</AccountActivity>
</L3ChargeFileRequest>
</Request>
```

> **Note**
>
> The CDF tags and the corresponding tags in the cXML file (that SAP Business Network generates and sends to the buying solution) have been color-coded.

Example 2: Purchase Order Charge with Line Level Details: Tax, Shipping, and Discount Values

The second example of CDF to cXML mappings consists of three pieces of data: a summary of the purchase order and charge details, a CDF file, and a corresponding cXML file.

The purchase order has the following details:

- PO ID: PCO17
- Line Items: 2
  - LI 1: Quantity 20 at Unit Price of $6.67
  - LI 2: Quantity 8 at Unit Price $73.00
- Total Ordered: $717.40
- Total Charge Amount: $1026.00

Charge Details:

- Shipping: $11.00
- Discount: -$9.00
- For LI 1:
  - Quantity: 20
  - Unit Price: $.6.67
  - Total Line Item Amount: $133.40
  - Goods and Services Tax: $90.00
- For LI 2:
  - Quantity: 8
  - Unit Price: $73.00
  - Goods and Services Tax: $20.00

CDF Sample

A sample CDF format (line level details) for the above charge looks similar to this:

> **Note**
>
> The below extract is not in its entirety. Some of the tags have been deleted to retain only the tags that provide the information required to substantiate the example.

```
<CDFTransmissionFile>
  <TransmissionHeader_1000 >
 </TransmissionHeader_1000 >
<IssuerEntity ICANumber = "1131" IssuerNumber = "00011111131" >
      <CorporateEntity CorporationNumber = "0000000011111111131" >
        <AccountEntity AccountNumber = "2222222222002001" >
<
     FinancialTransaction_5000
     >
     <
     AmountInOriginalCurrency
     CurrencyCode = "840" CurrencyExponent = "4" CurrencySign = "D"
>0000000010260000</AmountInOriginalCurrency>
            <OriginalCurrencyCode >840</OriginalCurrencyCode>
            <AmountInPostedCurrency CurrencyCode = "840" CurrencyExponent = "4"
CurrencySign = "D"         >0000000010260000</AmountInPostedCurrency>
```

```
            <AmountInBillingCurrency CurrencyCode = "840" CurrencyExponent = "4"
CurrencySign = "D"         >0000000010260000</AmountInBillingCurrency>
            <
     CustomerCode
     >PCO17</CustomerCode>
            <
     FreightAmount
     CurrencyCode = "840" CurrencyExponent = "4" CurrencySign = "D"
>0000000000110000</FreightAmount>
            <
     TransactionDiscountAmount
     CurrencyCode = "840" CurrencyExponent = "4" CurrencySign = "D"
>0000000000090000</TransactionDiscountAmount>
</FinancialTransaction_5000 >
<
     LineItemDetailEntity
     >
            <CorporateCardLineItemDetail_5010 >

     LineItemNum >0001</LineItemNum>

              <CustomerCode >PCO17</CustomerCode>
              <ItemDescription >item-1</ItemDescription>
              <ItemQuantity Exponent = "4" >0000000000200000</ItemQuantity>
              <ItemUnitOfMeasurementType >EA</ItemUnitOfMeasurementType>
              <UnitPrice CurrencyCode = "840" CurrencyExponent = "4" CurrencySign =
"D" >0000000000066700</UnitPrice>
              <DiscountIndicator >N</DiscountIndicator>
              <TypeOfSupply >00</TypeOfSupply>
              <LineItemTotalAmount CurrencyCode = "840" CurrencyExponent = "4"
CurrencySign = "D"         >0000000001334000</LineItemTotalAmount>
              <POLineNum >0001</POLineNum>
            </CorporateCardLineItemDetail_5010 >
    <
     TaxAddendum_5300
    >
        <TaxAmount CurrencyCode="840" CurrencyExponent="4"
CurrencySign="D">0000000000900000</TaxAmount>
             <TaxLocation>PO PROVINCIAL</TaxLocation>
            <NetAmount CurrencyCode="840" CurrencyExponent="4"
CurrencySign="D">0000000001334000</NetAmount>
            <TaxableAmount CurrencyCode="840" CurrencyExponent="4"
CurrencySign="D">0000000001334000</TaxableAmount>
    </TaxAddendum_5300>
</LineItemDetailEntity >

     <LineItemDetailEntity >

          <CorporateCardLineItemDetail_5010 >

     <LineItemNum >0002</LineItemNum>

        <CustomerCode >PCO17</CustomerCode>
        <ItemDescription >item-2</ItemDescription>
            <ItemQuantity Exponent = "4" >0000000000080000</ItemQuantity>
            <ItemUnitOfMeasurementType >EA</ItemUnitOfMeasurementType>
            <UnitPrice CurrencyCode = "840" CurrencyExponent = "4" CurrencySign =
"D" >0000000000730000</UnitPrice>
    <LineItemTotalAmount CurrencyCode = "840" CurrencyExponent = "4" CurrencySign =
"D" >0000000005840000</LineItemTotalAmount>
                              <POLineNum >0002</POLineNum>
            </CorporateCardLineItemDetail_5010 >

     <TaxAddendum_5300>

    <TaxAmount CurrencyCode="840" CurrencyExponent="4"
CurrencySign="D">0000000000200000</TaxAmount>
    <TaxLocation>PO PROVINCIAL</TaxLocation>
```

```
    <NetAmount CurrencyCode="840" CurrencyExponent="4"
CurrencySign="D">0000000005840000</NetAmount>
    <TaxableAmount CurrencyCode="840" CurrencyExponent="4"
CurrencySign="D">0000000005840000</TaxableAmount>
    </TaxAddendum_5300>
</LineItemDetailEntity >
  </FinancialTransactionEntity >
 </AccountEntity >
</CorporateEntity >
</IssuerEntity >
</CDFTransmissionFile>
```

cXML Sample

The above CDF file is converted to a cXML format by SAP Business Network and sent to the buying solution. A sample xCML file format corresponding to the above CDF is here:

```
<L3ChargeFileRequest>
<
     ChargeSummary
     transactionDate="2009-06-08" conversionRate="1.000000"
bankReferenceNumber="foo-21oct10-5">
<
     TotalAmountInPostedCurrency
    ><Money currency="USD">1026.0000</Money></TotalAmountInPostedCurrency>
<TotalAmountInBillingCurrency><Money currency="USD">1026.0000</Money></
TotalAmountInBillingCurrency>
<
     ShippingAmount
    ><Money currency="USD">11.0000</Money></ShippingAmount>
<
     DiscountAmount
    ><Money currency="USD">9.0000</Money></DiscountAmount>
<
     OrderIDInfo
     orderID="PCO17"></OrderIDInfo>
</ChargeSummary>
<AddendumItem>
  <DetailAddendum>
    <BasicAddendum>
      <
     DetailLineItem lineItemNumber="0001
    " subLineItemNumber="0000" purchaseDate="2009-06-08"
customerCode="PCO17"poLineNumber="0001">
      <Rate quantity="20.0000">
      <Total><Money currency="USD">133.4000</Money></Total>
      <UnitRate><Money currency="USD">6.6700</Money><UnitOfMeasure>EA</
UnitOfMeasure></UnitRate>
      <TaxableAmount><Money currency="USD">133.4000</Money></TaxableAmount>
      <
     TaxAmount
    ><Money currency="USD">90.0000</Money></TaxAmount>
      <TaxLocation xml:lang="en">PO PROVINCIAL</TaxLocation>
      </TaxDetail>
   </DetailLineItem>
  </BasicAddendum>
</DetailAddendum>
<AddendumItem>
   <DetailAddendum>
     <BasicAddendum>
       <
     DetailLineItem lineItemNumber="0002
    " subLineItemNumber="0000" purchaseDate="2009-06-08"
customerCode="PCO17"          poLineNumber="0002">
        <Rate quantity="8.0000">
        <Total><Money currency="USD">584.0000</Money></Total>
```

```
        <UnitRate><Money currency="USD">73.0000</Money><UnitOfMeasure>EA</
UnitOfMeasure></UnitRate>
        <TaxableAmount><Money currency="USD">584.0000</Money></TaxableAmount>
        <
     TaxAmount
    ><Money currency="USD">20.0000</Money></TaxAmount>
        <TaxLocation xml:lang="en">PO PROVINCIAL</TaxLocation>
        </TaxDetail>
        </DetailLineItem>
     </BasicAddendum>
   </DetailAddendum>
</AddendumItem>
</L3ChargeFileRequest>
```

Reconciliation of Level 3 Charges

Business users can assign and reconcile purchasing card charges that the buying solution could not match to orders during the automatic reconciliation phase. Members of the **PCard Manager** group can view and reconcile all charges that require reconciliation.

The level 3 charge information that comes into the buying solution is retained as is. The buying solution does not perform any calculations that previously affected the charge amounts.

Calculation of Unit Price When Line Item Amount Is Non-Zero

The buying solution calculates the unit price for all PCCR objects based on certain formula when the line item amount in non-zero.

- If Line Item Amount is Non-Zero, Item Quantity is Non-Zero, and the Unit Price is Zero or null then:
  `UnitPrice = Line Item Amount / Item Quantity`
- If Line Item Amount is Non-Zero, Item Quantity is Zero or null, and the Unit Price is Zero or null then:
  `Item Quantity = 1, UnitPrice = Line Item Amount`
- If Line Item Amount is Non-Zero, Item Quantity is Non-Zero, and Unit Price is Non-Zero; AND Item Quantity X UnitPrice != (not equal to) Line Item Amount then:
  `Item Quantity = 1, UnitPrice = Line Item Amount`

The following scenarios illustrates the working of these formula:

Scenario 1:

A purchase order has a charge amount of $100.00, with one line item having:

- Quantity: 10
- Unit Price: $0.00
- Amount: $50.00
- Total Tax for the order: $30.00

In this scenario, the buying solution uses the first formula listed above, calculating the unit price by dividing the line item amount by quantity (that is, 50/10 = 5). So unit price is calculated as $5.00 and the same is populated in the order line.

> **Note**
>
> Since the order line amount and the tax amount do not add up to the order amount ($100.00), an adjustment line item (ALI) will be created for $20.00, and you will be able to set the appropriate type of adjustment.

Scenario 2:

An order has a charge amount of $100.00, with one line item having:

- Quantity: 100
- Unit Price: $0.50
- Amount: $50.00
- Total Tax for the order: $30.00

In this scenario, buying solution re-calculates the line item quantity as 1 (one), the unit price as $50.00, and the amount also as $50.00 (calculated using the third formula that is listed above).

> **Note**
>
> Since the order line amount and the tax amount do not add up to the order amount (of $100.00), an adjustment line item (ALI) will be created for $20.00, and you will be able to set the appropriate type of adjustment.

Calculation of Unit Price When Item Quantity Is Zero

The buying solution calculates the unit price for all PCCR objects based on certain formula when the quantity is zero.

The unit price is calculated based on the following formula:

- If Line Item Amount is Non-Zero, Item Quantity is Zero or null, and the Unit Price is Zero or null then: `Item Quantity = 1, UnitPrice = Line Item Amount`
- If Line Item Amount is Non-Zero, Item Quantity is Zero or null, and the Unit Price is Non-Zero then: `Item Quantity = Line Item Amount / UnitPrice`
- If Line Item Amount is Non-Zero, Item Quantity is Non-Zero, and the Unit Price is Non-Zero; AND, the Item Quantity X UnitPrice != Line Item Amount then: `Item Quantity=1, UnitPrice=Line Item Amount`

The following scenario illustrates the working of these formula:

An order has a charge amount of $100.00, with one line item having:

- Quantity: 0
- Unit Price: $0.50
- Amount: $50.00
- Total Tax for the order: $30.00

In this scenario, the buying solution re-calculates the line item quantity as 100, unit price as $0.50, and the amount as $50.00.

> **Note**
>
> Since the order line amount and the tax amount do not add up to the order amount ($100.00), an adjustment line item (ALI) will be created for $20.00, and you will be able to set the appropriate type of adjustment.

Charges Having Line Amount as Zero

The buying solution allows users to reconcile those charges that have the line amount as 0 (zero).

For example, if there is a PO charge amount of $100.00 with a PO line item having:

- Quantity: 10
- Unit Price: $0.50
- Amount: $.0.00
- Total tax: $20.00.

In this scenario, the buying solution re-calculates all values to be zero (that is, quantity, unit price, and amount all having a value of zero) and allows the charge to be reconciled.

> **Note**
>
> Since the order line amount and the tax amount don't add up to the PO amount (of $100.00), an adjustment line item (ALI) will be created for $80.00, and you will be able to set the appropriate type of adjustment.

Related Information

Purchasing Guide for Procurement Professionals

Working with PCards

The PCard Manager workspace allows you perform many actions related to PCards.

You can view, add, modify, and delete PCards, assign PCards to users, and associate PCards with supplier locations, import and export PCard-related data, view details, errors, and the history of past data imports and exports, work with adjustment types, and edit adjustment types, including name, description, and adjustment category.

Modifying PCard Information

Members of the **PCard Manager** group can modify PCard information, including general information and purchase card numbers.

Viewing Information About PCards

Members of the **PCard Manager** group can view PCard information.

Procedure

1. On the dashboard, click **Manage** > **Core Administration** .
2. Click **PCard Manager**.
3. Click **PCards** and select **No Choice** to display all purchasing cards, or select from the available options (**Unknown, Expense Card, Procurement Card,** or **Combination Card**) and click **Search**.

   The following table describes the fields on the purchasing cards page.

Column	Description
Card Type	The card type.
PCard Holder	The name of the card holder.
Expiration Date	The date the card expires.

4. To view details, such as users assigned the purchasing card, click a purchasing card number in the **Card Number** column.
5. Click **Done** to return to the previous page.

Modification of PCard Numbers

You can modify a PCard number in the PCard Manager workspace or by running the **Import PCards** task.

You can do one of the following to modify a PCard number:

- Use the PCards task in the PCard Manager workspace to delete the PCard, and then add the PCard with the correct card number.
- Run the **Import PCards** data import task to deactivate the PCard (Deactivate operation), and then run the **Import PCards** data import task again to import the PCard with the correct card number (Load operation).

Modifying General PCard Information

You can modify general PCard information, such as the Adapter Source, and Card Type.

Procedure

1. On the dashboard, click **Manage** > **Core Administration**.
2. Click **PCard Manager**.
3. Click **PCards**, select the purchasing card, and click **Actions** > **Edit**.

   The **Edit PCard** page appears.
4. Click the **General** tab. The following table describes the fields on the **General** tab:

Field	Description
Adapter Source	The adapter source, which can be one of the following values: • **AribaManaged**—The object is not maintained in an external system. Objects with this data source are maintained in Ariba Administrator. • **External**—The object and all its data fields are maintained in an external system, such as an ERP system or a set of CSV files. Objects with this adapter source can be overwritten or deleted when data is imported from the external system. • **ManuallyMaintained**—The source of the object is an external system, but the object cannot be overwritten or deleted when data is imported from the external system. You cannot modify this field in edit mode.
Card Number	The card number. The card number is not shown in its entirety for security reasons.
Card Type	The card type.
PCard Holder	The name of the card holder.
PCard Charge Reconciler	The user who can reconcile charge exceptions associated with this card.
Expiration Date	The date the card expires.
PCard Bank	The bank that issued the card.
Liability	Whether the card holder or the company is liable for charges on the card.

5. Click **Save** to save your changes, or **Cancel** to return to the previous page without saving your changes.

Creating New PCards

Members of the **PCard Manager** group can create (add) new PCards.

Procedure

1. On the dashboard, click **Manage** > **Core Administration** .
2. Click **PCard Manager**.
3. Click **PCards**, and click **Create New**.

   The **Create PCard** page appears.
4. On the **Create PCard** page, enter the following information:

Field	Description
Adapter Source	The adapter source, which can be one of the following values: • **AribaManaged**—The object is not maintained in an external system. Objects with this data source are maintained in Ariba Administrator. • **External**—The object and all its data fields are maintained in an external system, such as an ERP system or a set of CSV files. Objects with this adapter source can be overwritten or deleted when data is imported from the external system. • **ManuallyMaintained**—The source of the object is an external system, but the object cannot be overwritten or deleted when data is imported from the external system.
Card Number	The number of the card.
Card Type	The card type.
PCard Holder	The name of the card holder.
PCard Charge Reconciler	The user who can reconcile charge exceptions associated with this card.
Expiration Date	The date the card should expire.
PCard Bank	The bank that issued the card.
Liability	Whether the card holder or the company is liable for charges on the card.

5. Click **Save** to save the new PCcard, or **Cancel** to return to the previous page without saving the new PCard.

Assigning PCards to Users

Members of the **PCard Manager** group can assign an existing PCard to a user.

Procedure

1. On the dashboard, click **Manage** > **Core Administration**.
2. Click **PCard Manager**.
3. Click **PCards**, select the PCard, and click **Edit**.
4. On the **Users With Current PCard** tab, click **Add/Remove** to display the available choices.
5. Choose one or more users and click **OK**.
6. Click **Save** to save your changes, or **Cancel** to return to the previous page without saving your changes.

Associating PCards with Supplier Locations

Members of the **PCard Manager** group can associate PCards with supplier locations.

Procedure

1. On the dashboard, click **Manage** > **Core Administration**.
2. Click **PCard Manager**.
3. Click **PCards**, select the PCard, and click **Edit**.
4. On the **Supplier Locations Accepting Current PCard** tab, click **Add/Remove** to display the available choices.
5. Choose one or more supplier locations and click **OK**.
6. Click **Save** to save your changes, or **Cancel** to return to the previous page without saving your changes.

Deleting PCards

Members of the **PCard Manager** group can deactivate (delete) PCards.

Procedure

1. On the dashboard, click **Manage** > **Core Administration**.

2. Click **PCard Manager**.
3. Click **PCards**, select the PCard you want to delete, and then click **Actions** > **Deactivate**.

   Alternatively, you can select the PCard that you want to deactivate and click **Delete**.

   > **Note**
   >
   > The **Delete** action does not delete the PCard from the system, but only deactivates it. To activate a PCcard, select that card and click **Actions** > **Activate** at any time.

4. Click **OK** to delete the PCcard, or **Cancel** to return to the previous page without deleting the PCcard.

Enabling Additional Users to Reconcile PCard Charge Exceptions

You can permit additional users to reconcile PCard charge exceptions even if they are not members of the **Procurement Manager** or **PCard Manager** groups.

You can assign a user to an existing PCard as a PCard Charge Reconciler (PCCR) and specify the charge exceptions that the user can reconcile in the `ChargeExceptionType.csv` file. The buying solution using the standard approval flow mechanism adds the user as an approver to all applicable PCCR documents. The user can then access these PCCR documents from the **To Do** action tile, review the exceptions, suggest corrections, and then submit the adjusted PCCR for further approval.

> **Note**
>
> The user must be a member of the **Charge Users** group to reconcile charge exceptions, and must be a member of the **Charge Rejection Specialist** group to reject charges.

Enabling users who are not members of the **Procurement Manager** or **PCard Manager** groups to reconcile PCard charge exceptions involves the following tasks:

- Specifying a user as a PCard charge reconciler
- Specifying the charge exception types that the user can reconcile

Specifying a User as a PCard Charge Reconciler

As a member of the **PCard Manager** group, you can specify a user who can reconcile charge exceptions associated with a purchasing card.

Procedure

1. In Ariba Administrator, navigate to **PCard Manager** > **PCards**.

2. Select a purchasing card and then click **Actions** > **Edit**.
3. On the **General** tab, select a user from the **PCard Charge Reconciler** pull-down menu.
4. Click **Save** to save your changes.

Specifying the Exception Types That Users Can Reconcile

You use the `Import Charge Exception Types` data import task with the `ChargeExceptionType.csv` file to define the exception types. The `PCardReconcilerAllowed` column in the CSV file specifies which charge exception types a PCard charge reconciler can reconcile.

Procedure

1. On the dashboard, click **Manage** > **Core Administration**.
2. Click **Site Manager** > **Data Import/Export Tasks**.
3. Search for the **Import Charge Exception Types** data import task and click **Import**.
4. Click **Browse**, navigate to the location of the `ChargeExceptionType.csv` file and click **Run** to import it.

PCard Adjustment Types

Use the `Adjustment Types` task in the **PCard Manager** workspace in Ariba Administrator to view, add, modify, and delete adjustment types.

In the default configuration, the buying solution provides the following adjustment types:

- Sales Tax
- Shipping

You can create new adjustment types as required.

Adding a PCard Adjustment Type

Members of the **PCard Manager** group can add a PCard adjustment type.

Procedure

1. On the dashboard, click **Manage** > **Core Administration**.

2. Click **PCard Manager**.
3. Click **Adjustment Types** and click **Create New** to add an adjustment type.
4. Enter or modify the following information:

Field	Description
Adapter Source	The adapter source. Choose from • **AribaManaged**: Indicates that the adjustment types are managed and maintained within the Ariba Administrator • **External**: Indicates that the adjustment type data can be overwritten or deleted when data is imported from the external system. • **ManuallyMaintained**: Indicates an adjustment type that is set to be manually maintained cannot be updated through the data import task. You must update this adjustment type manually through the application user interface.
Unique Name	The unique name of the adjustment type. **Note** The Sales Tax adjustment type that is available in the default configuration has the unique name as **Default**.
Name	The user-visible name of the adjustment type. The default configuration includes two adjustment types — **Sales Tax** and **Shipping/Handling**.
Description	The description of the adjustment type.
Category	A grouping for adjustment categories. This field is visible in the user interface, available for your use in approval rules or initialization, but is not otherwise used in the default configuration.
Account	The unique name of an account. When users adjust prices, adjustments must also be handled on the accounting side. you specify an account in this field, all adjustments are made against that account by default, so users do not need to supply account names or numbers when making adjustments.

5. Click **Save**.

Enhanced Charge Reconciliation for PCard Orders

As a member of the **Customer Administrator** group, you can enable an enhanced charge reconciliation process for PCard orders that allows reconciliation when only the PCard number is available in the imported charge. The buying solution will find the most recent PCard order corresponding to the PCard number.

Related Information

Enabling Enhanced Charge Reconciliation for Purchasing Card Orders [page 224]
Purchasing Guide for Procurement Professionals

Enabling Enhanced Charge Reconciliation for Purchasing Card Orders

Prerequisites

You must be a member of the **Customer Administrator** group to modify configuration parameters.

Context

This feature is controlled by a site configuration parameter, which you can set in Ariba Administrator under **Customization Manager** > **Parameters**.

- `Application.Purchasing.FetchPCardOrderByPCardNumber` (enabled for self-service)

Determines whether the buying solution from SAP Ariba finds the most recent purchasing card order corresponding to the purchasing card number if the `Charge.csv` file does not have the order number.

The default value of this parameter is **No**.

Procedure

1. Click **Manage** > **Core Administration**
2. Click **Customization Manager** > **Parameters**
3. Click **List All** to get a listing of all parameters available for self-service or enter any part of the parameter name to search. For example, enter **`PCard`** and click **Search** to see all parameters related to purchasing cards that have the string `PCard` anywhere in the name.
4. Review the parameter settings and modify it as required. In this case, set it to to enable the enhanced charge reconciliation.

Configuration and Management of Virtual Cards

Buyers can pay suppliers using single-use virtual cards for the catalog and non-catalog item they buy from suppliers. The card provider connnected to the buyer's bank generates a virtual card per purchase order and sends the card details to the supplier. The supplier uses the virtual card to receive payment from the buyer's bank and the buyer's account is charged. After reconciling the amount against the purchase order in SAP Ariba buying solution, the buyer updates the details to the buyer's ERP system.

Prerequisites

- Enable the **Export Virtual Card Charge Payment Requests** parameter. For more information, refer to Use single-use virtual cards to pay suppliers.
- Create approval flows for both virtual card charge and virtual card charge reconciliation approvable types to ensure that the charges are processed correctly and to prevent charge document failures. For more information, see Creating a New Approval Process.

High-Level Steps to Configure and Use Virtual Cards

To understand, configure, and use this feature, we suggest you learn the following topics:

This feature enables you to use single-use virtual cards to pay suppliers for the materials that you have bought from them.	For more information, refer to the following topic: • Virtual Cards [page 227]

Set Up Virtual Card Accounts for Buyers

To understand the process flow for setting up virtual card accounts for buyers, refer to the Buyer Account for the Virtual Card Account – Process Flow [page 228] section.

1.	To use single-use virtual cards to pay suppliers, you must first set up a virtual card account.	For more information, refer to the following topics: • SAP Buyer Account for Virtual Cards [page 228] • Setting Up a Buyer Account for a Virtual Card [page 229] • Viewing Details of a Virtual Card Account for Buyers [page 233] • Modifying the Details of a Buyer Account for the Virtual Card [page 233]

2.	Then, you can send the following master data that needs to be sent from the SAP Ariba buying solution to SAP: • Buying organization • Suppliers	For more information, refer to the following topic: • Master Data to SAP [page 234]
3.	After the successful transfer of master data, you can enable the support of virtual cards to suppliers.	For more information, refer to the following topics: • Enabling Virtual Card Support for New Suppliers [page 235] • Enabling Virtual Card Support for Existing Suppliers [page 236]

Procure Catalog or Non-Catalog Items Using a Virtual Card Order

To understand the process flow for procuring items using a virtual card order (VCO), refer to the Procurement Using a Virtual Card Order – Process Flow [page 238] section.

1.	To buy catalog or non-catalog items and pay for them using virtual cards, you can create a purchase requisition for the items you want to buy. In the requisition, you can specify a supplier enabled with the support of the virtual card. Then you can create a purchase order using the purchase requisition. This purchase order is known as a virtual card order (VCO).	For more information, refer to the following topics: • Virtual Card Order [page 239] • SAP Ariba Virtual Card Order Method [page 239]
2.	After the supplier receives VCO in the supplier's account in SAP Business Network, the buyer sends the request for the virtual card to SAP. Then, SAP contacts the card provider, and the card provider contacts the buyer's bank. Then, the card provider shares the virtual card number for the virtual card order with the supplier.	For more information, refer to the following topic: • Virtual Card Details for Suppliers [page 240]

Charge a Virtual Card and Perform Charge Reconciliation Against Virtual Card Orders

To understand the process flow for charge reconciliation and payment, refer to the Charge Reconciliation and Payment – Process Flow [page 240] section.

After the successful delivery of goods to buyers, the supplier charges the virtual card of the buyer. Then, you can import the charge that the bank paid to the supplier using a .csv file, create a reconciliation document, perform reconciliation against the VCO, and export the virtual card charge payment request to your ERP system.	For more information, refer to the following topics: • Charge a Virtual Card and Perform Charge Reconciliation Against the Virtual Card Order [page 241] • Importing Charges [page 242] • Viewing the Details of Virtual Card Charges [page 243] • Creating the Virtual Card Charge Reconciliation Document [page 244] • Reconciling Charges [page 244] • Exporting Virtual Card Charge Payment Requests to the Buyer's ERP system [page 245]

Scheduled Tasks Used for Virtual Card Support

The following are the details of the scheduled tasks used for supporting virtual cards:

Name	Purpose	Typical Scheduling Duration
Retrieve the Status of the Virtual Card Setup	To know whether SAP has authorized the status of the virtual card setup created by the buyer.	Twice every day
Qualify Buying Organization's Information for Using Virtual Card	To send the buyer's business unit master data information to SAP. This scheduled task runs after SAP approves the virtual card setup created by the buyer.	Once every day
Qualify Supplier's Information for Using Virtual Card	To send the information of suppliers with the virtual card support enabled to SAP. This scheduled task runs after the buyer's business unit master data information is sent to SAP.	Once every day
Retrieve the Status of Qualification Documents for Virtual Cards in Batch	This scheduled task is run separately after sending the business unit master data and the supplier information to SAP. This scheduled task is run to change the status of the business unit and the supplier status from **New** to **Submitted**.	Four times every day
Retrieve masked Virtual Card Number	This scheduled task is run to associate the virtual card number with the virtual card order.	Every 20 minutes
Process Imported Virtual Card Charges	This scheduled task is run to create the charge reconciliation document for the buyer.	Once every day

Virtual Cards

A virtual card is a financial tool that helps buyers make digital payments to their suppliers.

Virtual cards are embedded within SAP Ariba Buying and SAP Ariba Buying and Invoicing. A virtual card is issued against the buyer's purchasing card account with the bank. Virtual cards function like traditional credit cards. However, the following features distinguish virtual cards from credit cards:

- A virtual card is used to pay a specific supplier for a specific transaction.

- Each virtual card is assigned to a specific purchase order, referred to as a virtual card order. This mapping with the virtual card order enables the reconciliation of charges and speeds up the generation of the payment object.
- Buyers cannot use a virtual card to pay for items not included in the virtual card order.
- Virtual cards contain software-generated card numbers. This enhances security and confidentiality.

Buyer Account for the Virtual Card Account – Process Flow

The process flow for setting up a buyer account for the virtual card is as follows:

> ⓘ Note
>
> You can set up a buyer account for virtual cards only in a child site and not in a parent site.

1. In SAP Ariba Buying and SAP Ariba Buying and Invoicing, the buyer administrator navigates to the **Virtual Card Setup** page.
2. To set up a new buyer account for the virtual card, the buyer administrator submits the required account and bank information to SAP. The request displays the **Composing** status.
3. After submitting to SAP, the request's status changes from **Composing** to **Submitted**.
4. Based on the request submitted by the buyer administrator, SAP sets up the buyer account for the virtual card. This manual activity is completed within 48 hours after SAP receives the buyer administrator's request.
   The manual activity also includes SAP contacting the virtual card provider and the card provider contacting the bank associated with the buyer where the buyer has an active purchase card account. Based on the buyer's purchase card account, the bank approves the buyer using virtual cards.
   Then, the following scheduled tasks are run in the SAP Ariba buying solution:
   1. The **Retrieve the Status of the Virtual Card Setup** scheduled task to know whether SAP has authorized the status of the virtual card setup.
   2. After SAP authorizes the virtual card setup, the status of the request changes to **Authorized**.
   3. The **Qualify Buying Organization's Information for Using Virtual Card** scheduled task to send the business unit information to SAP. The business unit elements display the status, **New**.
   4. The **Retrieve the Status of Qualification Documents for Virtual Cards in Batch** scheduled task to change the status of the business unit elements changes from **New** to **Submitted**.
   5. The **Qualify Supplier's Information for Using Virtual Card** scheduled task to send the information of suppliers with virtual card support enabled to SAP. The suppliers' status displays as **New**.
   6. The **Retrieve the Status of Qualification Documents for Virtual Cards in Batch** scheduled task to change the suppliers' status changes from **New** to **Submitted**.
5. The status of the virtual card setup changes from **Authorized** to **Active**.

SAP Buyer Account for Virtual Cards

An SAP buyer account for the virtual card must be set up to enable buyers to use the virtual card for supplier payments.

As the buyer administrator, you must provide the account and the bank information to create the SAP buyer account for the virtual card. While creating it, the account setup request is in **Composing**. After you submit the request to SAP, the status changes to **Submitted**.

At SAP, the request for setting up the buyer account for the virtual card is processed manually. Ideally, this process will be completed, and the virtual card account for the buyer will be set up within 48 hours.

In SAP Ariba Buying and SAP Ariba Buying and Invoicing, you can schedule the **Retrieve the Status of the Virtual Card Setup** task to know the following status of the buyer account created for the virtual card:

- When autorization of the buyer account for the virtual card is in progress, the status is displayed as **In Authorization**.
- After SAP authorizes the buyer account for the virtual card, the status is displayed as **Authorized**.

Related Information

Setting Up a Buyer Account for a Virtual Card [page 229]
Viewing Details of a Virtual Card Account for Buyers [page 233]
Modifying the Details of a Buyer Account for the Virtual Card [page 233]

Setting Up a Buyer Account for a Virtual Card

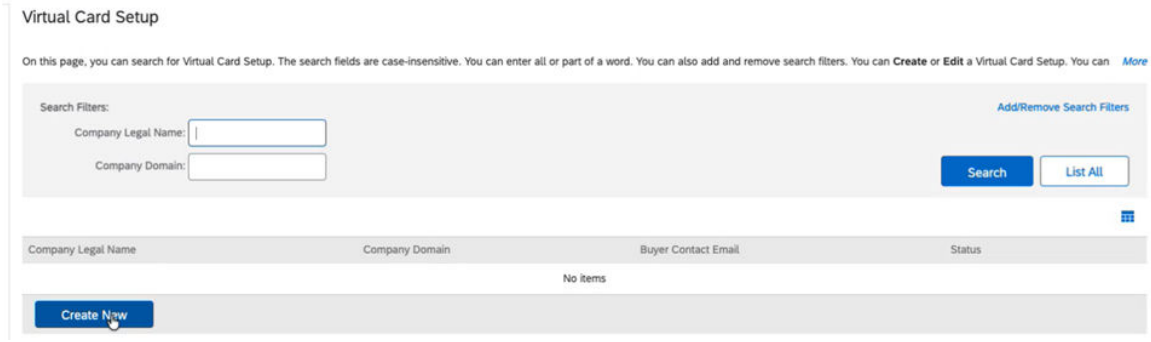
This topic describes how to set up a buyer account for a virtual card account in SAP Ariba buying solutions.

Context

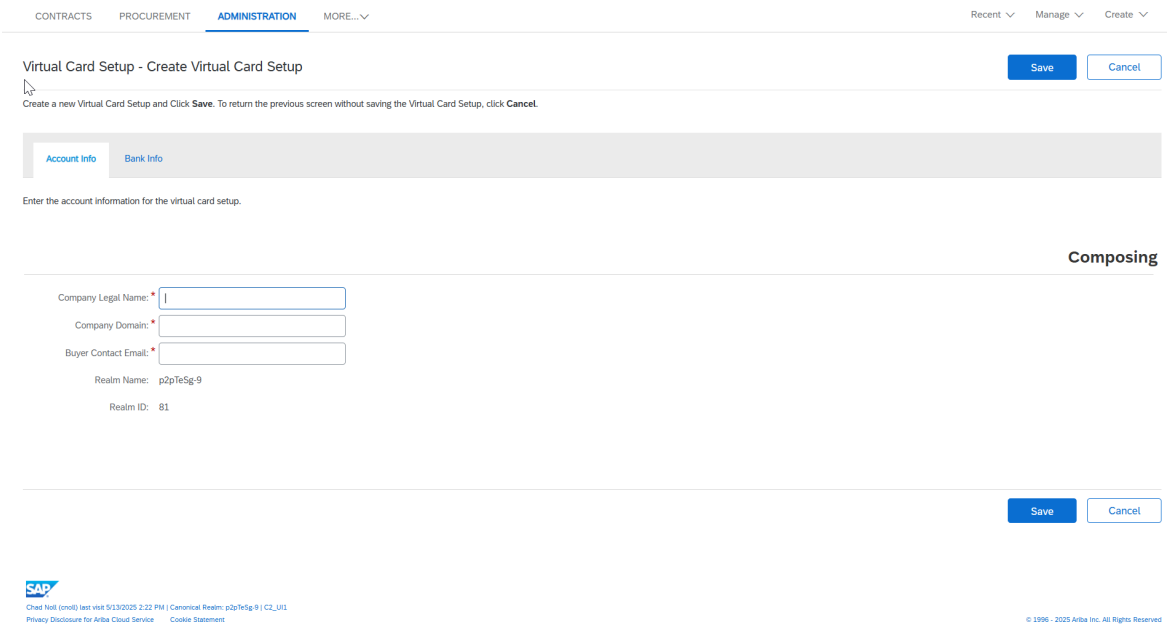
To enable a buyer to use virtual cards for supplier payments, the buyer administrator must first set up the buyer account for the virtual card. The buyer administrator must be part of the **VCard Manager** group to get the required permissions to perform this task.

Procedure

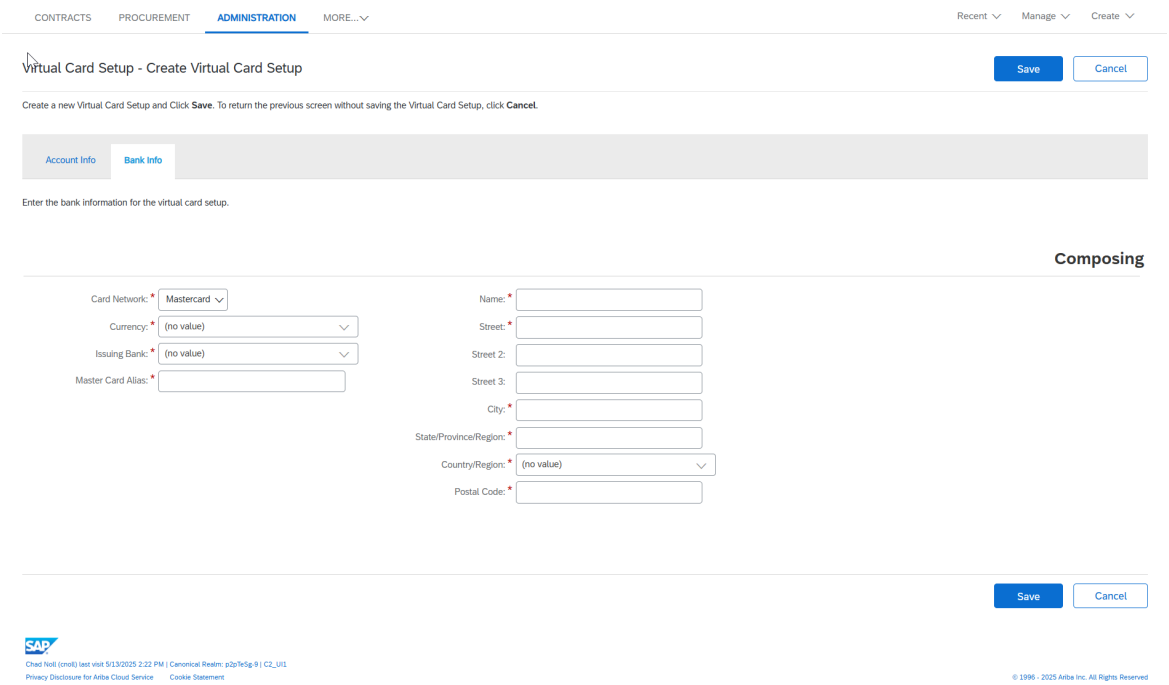
1. On the dashboard, click **Manage** > **Core Administration**.
2. Click **Virtual Card Manager** > **Virtual Card Setup**.
3. On the right side, on the **Virtual Card Setup** page, do the following:
   a. Click **Create New**.



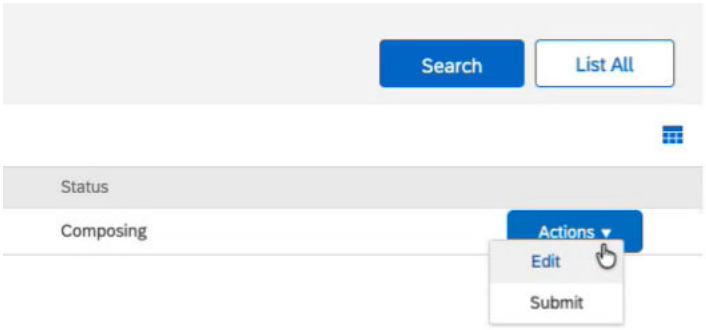
a. In the **Account Info** tab, enter the details. For more information, refer to Account Info [page 231].



b. In the **Bank Info** tab, enter the details. For more information, refer to Bank Info [page 232].



4. Click **Save**. The virtual card setup request is saved in **Composing** status.
5. On the **Virtual Card Setup** page, use **Search Filters** or **List All** to locate the saved virtual card setup request.
6. Select **Actions** > **Submit**.



The virtual card setup request moves to **In Authorization** status.

Account Info

This topic helps you understand the fields in the **Account Info** tab.

Field	Description
Company Legal Name	The legal name of the buyer's organization.
Company Domain	The domain of the buyer's organization. For example, Pharmaceutical, Automobile, and so on.

Field	Description
Buyer Contact Email	The email address on which the buyer can be contacted.
Realm Name	The name of the buyer's realm for which the virtual card payment needs to be enabled. This field appears populated.
Realm ID	The ID of the buyer's realm for which the virtual card payment needs to be enabled.

Setting Up a Buyer Account for a Virtual Card [page 229]

Bank Info

This topic helps you understand the fields in the **Bank Info** tab.

Field	Description
Card Network	The name of the organization that provides the virtual card to the buyer. For example, Mastercard, Visa.
Currency	The currency that the virtual card uses for paying suppliers.
Issuing Bank	The issuing bank options are drawn from the payment bank locations that are configured in the buyer's realm. For more information, refer to: • Adding and Modifying Payment Bank Locations in the Administration User Interface • Import Buyer Bank Payment Locations
Master Card Alias	Mastercard alias information.
Billing address details: • **Name** • **Street** • **Street 2** • **Street 3** • **City** • **State/Province/Region** • **Country/Region** • **Postal Code**	The billing address of the virtual card. The bank uses this address to send the card statement to the buyer.

Setting Up a Buyer Account for a Virtual Card [page 229]

Viewing Details of a Virtual Card Account for Buyers

This topic describes how to view the details of a buyer account for a virtual card.

Context

The buyer administrator can view the details of the existing buyer account for the virtual card in the SAP Ariba buying solution.

Procedure

1. On the dashboard, click **Manage** > **Core Administration**.
2. Click **Virtual Card Manager** > **Virtual Card Setup**.
3. On the right side, on the **Virtual Card Setup** page, in the **Search Filters** section, do one of the following:
   - Enter the company's legal name, company's domain name, or both, and click **Search** to list the buyer accounts for virtual cards created for your organization.
   - Click **List All** without entering the company's legal name or domain name to list all buyer accounts for virtual cards created for all domains in your organization.
4. In the table that lists the accounts, identify the account that you want to view in detail, and click the company's legal name.
5. After viewing the details, click **Done** to return to the page that lists the account details.

Modifying the Details of a Buyer Account for the Virtual Card

This topic describes how to modify the details of a buyer account for the virtual card.

Context

As a buyer administrator, you can modify only the details of the buyer account of the virtual card in the **Composing** state. The **Edit** operation is not supported for the accounts in the **Authorized** state.

Procedure

1. On the dashboard, click **Manage** > **Core Administration**.
2. Click **Virtual Card Manager** > **Virtual Card Setup**.
3. On the right side, on the **Virtual Card Setup** page, search for the virtual card accounts that you want to edit.
4. In the table that lists the virtual card accounts, identify the account in the **Composing** state that you want to edit and click **Actions** > **Edit**.
5. On the **Virtual Card Setup - Edit Virtual Card Setup** page, navigate to the **Account Info** tab and the **Bank Info** tab as required and modify the required details.
6. Click **Save**.

Master Data to SAP

After the buyer account for the virtual card is set up, information related to the buying organization and suppliers is sent from the SAP Ariba buying solution to SAP Taulia, where the account request is processed. This transfer is handled through the integration between SAP Ariba and SAP Taulia and does not involve an ERP system.

Information related to the buying organization such as business unit, purchasing organization, and procurement unit is sent to SAP. The **Qualify Buying Organization's Information for Using Virtual Card** scheduled task transfers this information to SAP Taulia.

After transferring information related to the buying organization, the information about the suppliers enabled with virtual card support is to be transferred to SAP Taulia. We suggest you to perform this transfer after setting up the buyer account for the virtual card. The supplier information will be transferred automatically to SAP Taulia when the **Qualify Supplier's Information for Using Virtual Card** scheduled task runs.

These scheduled tasks run automatically. You do not have access to run them manually. For more information, see Scheduled Tasks Used for Virtual Card Support [page 227].

> **Note**
>
> If you have already transferred the information of suppliers enabled with virtual card support to SAP Taulia before setting up the buyer account for the virtual card, the **Qualify Supplier's Information for Using Virtual Card** scheduled task runs after the **Qualify Buying Organization's Information for Using Virtual Card** scheduled task run to transfer the buying organization information.

Virtual Card Support for Suppliers

The following topics explains how to enable the virtual card support for new and existing suppliers:

- Enabling Virtual Card Support for New Suppliers [page 235]

- Enabling Virtual Card Support for Existing Suppliers [page 236]

Enabling Virtual Card Support for New Suppliers

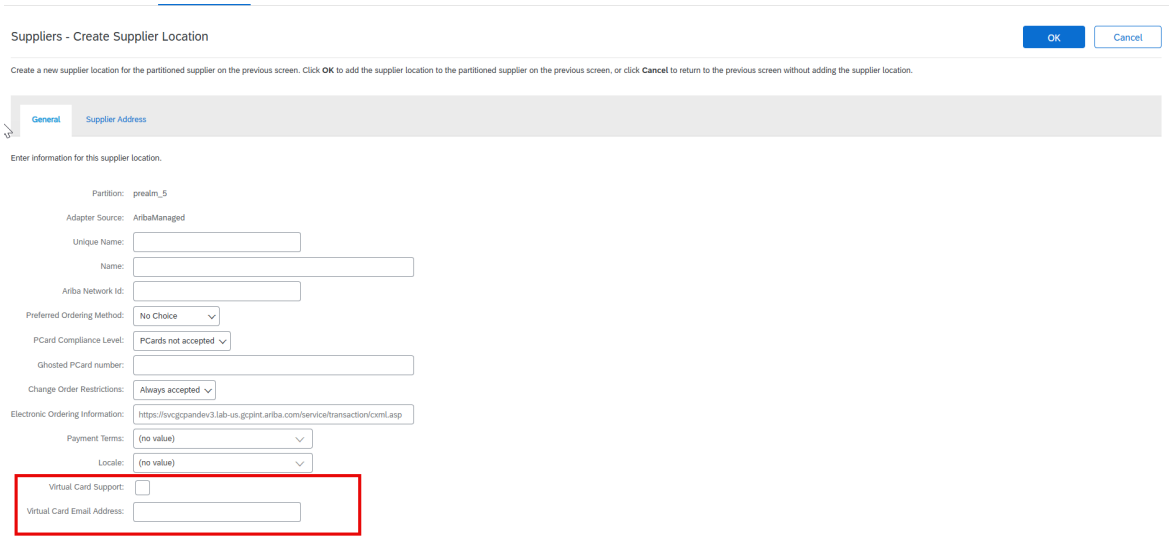
Context

You can enable virtual card support for a new supplier. Later, buyers can use this supplier while creating the purchase requests and creating a virtual card order.

You can enable virtual card support for the supplier available in the **Partitioned Supplier** tab. In the **Partitioned Supplier** tab, you can either create a new supplier or relink a common supplier.

Procedure

1. On the dashboard, click **Manage** > **Core Administration**.
2. Click **Supplier Manager** > **Suppliers**.
3. In the **Suppliers** page on the right side, click **Create New**.
4. In the **Partitioned Suppliers** tab, do the following:
   a. Click **Create New**.
   b. On the **Suppliers - Create Partitioned Supplier** page, navigate to the **Supplier Locations** tab.
   c. Click **Create New**.
   d. In the **General** tab, select the **Virtual Card Support** check box.
   e. In the **General** tab, in the **Virtual Card Email Address** field, enter the email address on which the supplier can receive the virtual card details from the bank.



5. Click **OK** > **OK** > **Save** to save your changes.

Enabling Virtual Card Support for Existing Suppliers

Context

This topic describes how to enable virtual card support for an existing supplier.

Procedure

1. On the dashboard, click **Manage** > **Core Administration**.
2. Click **Supplier Manager** > **Suppliers**.
3. On the right side, on the **Supplier** page, select the supplier for which you want to enable the virtual card support.
4. Click **Actions** > **Edit**.
5. Navigate to the **Partitioned Suppliers** tab.
6. In the **Partitioned Suppliers** tab, select the supplier to which you want to enable the virtual card support and click **Edit**.
7. Navigate to the **Supplier Locations** tab.
8. In the **Supplier Locations** tab, select the supplier location where you want to enable the virtual card support and click **Edit**.
9. In the **General** tab, do the following:
    a. Select the **Virtual Card Support** check box.
    b. In the **Virtual Card Email Address** field, enter the email address on which the supplier can receive the virtual card details from the bank.
10. Click **OK** > **OK** > **Save** to save your changes.

Results

After enabling virtual card support, all orders are processed as virtual card orders. If a change order is made to an existing purchase order, the SAP Ariba solution cancels the original purchase order and generates a new virtual card order.

Import Virtual Card Configuration Details to the Vendor Location

This topic explains how to enable virtual card support for supplier locations using a data import task. The import task allows administrators to configure virtual card support for multiple supplier locations for both new and existing suppliers, without editing each location individually.

Prerequisites

- The supplier location already exists in the system (not ad hoc vendors).
- You have permission to run data import and export tasks in **Site Manager** > **Data Import/Export**. This permission is available to users assigned to the Customer Administrator group. For more information, see SAP Ariba Procurement Solutions Group Descriptions Table.
- Prepare a UTF-8 encoded `SupplierLocationVCardConfig.csv` file with the virtual card configuration details. The file should contain the following columns:

Column Name	Description
UniqueName	Unique identifier for the supplier location, typically the supplier location ID.
ContactID	Contact ID for the supplier location.
supportsvCard	**True** or **False**. Setting it to true enables the support.
vCardEmailAddress	Supplier's email address for receiving virtual card details from the bank

Example:

```
UTF8
UniqueName,ContactID,supportsvCard,vCardEmailAddress
Supplier123,Contact456,true,vcards@acme.com
67890,654,true,jane@xyz.com
```

Procedure

1. On the dashboard, click **Manage** > **Core Administration**.
2. Navigate to **Site Manager** > **Data Import/Export**.
3. Locate **Import Virtual Card Configuration Details to the Vendor Location** and click **Import**.
4. Select the operation type: **Load** or **Update Only**.
5. Navigate to the folder containing the prepared `SupplierLocationVCardConfig.csv` file and select it.
6. Click **Run**. The import task begins.

Results

After the import task completes successfully, the selected supplier locations are enabled for virtual card support, and the specified email addresses are configured to receive virtual card details.

Procurement Using a Virtual Card Order – Process Flow

The process flow for procuring catalog and non-catalog items using a virtual card order is as follows:

1. In the SAP Ariba buying solution, create a purchase requisition for catalog or non-catalog items. In the requisition, specify a supplier enabled with the virtual card support. Submit the requisition for approval.
2. The `SAP Ariba Virtual Card` (`ariba.vcard.ordering.AribaSSPVCard`) order method is used to process and send virtual card orders to suppliers.
3. After the approvers verify and approve the requisition, a virtual card order (VCO) is created.
4. The virtual card order is sent to the supplier's account in SAP Business Network. The supplier receives the virtual card order.
5. The status of the virtual card order changes to **Ordered**.
6. The buyer's request for providing virtual card against a virtual card order is sent to SAP. Then, SAP contacts the card provider, and the card provider contacts the buyer's bank. Then, the card provider shares the virtual card number for the virtual card order with the supplier.
   In the **History** tab of the virtual card order in the SAP Ariba buying solution, the **Virtual Card Requested** action is recorded.
7. The **Retrieve masked Virtual Card Number** scheduled task is run in the SAP Ariba buying solution to associate the virtual card number with the virtual card order.
8. The card provider sends an email that contains the last four digits of the card number and the expiration date. The expiration date indicates the validity period of the virtual card and is defined by the card provider.
   An example of the details sent is as follows:
   - Card Account: XXXX-XXXX-XXXX-4724
   - Expiration Date: 05/2026

   The email also indicates that the authorization for the card account is for the gross amount and that the authorization is valid for a limited period.

   > **Note**
   >
   > Authorization refers to the reservation of funds for a transaction. Funds are reserved when the transaction is approved, either at the point of sale or during an online purchase. Although the card is authorized at the time of the transaction, the clearing and settlement is done later. The authorization period may vary depending on the card provider. While settlement typically occurs within 3 days, some providers may extend the authorization period up to 7 days or longer.

   In the **History** tab of the virtual card order, the **Virtual Card Received** action is recorded. You can also view the masked virtual card number for the requester's reference.
9. Once the supplier receives the receipt from the buyer, they charge the virtual card based on the agreement between both parties.

10. Based on the buyer's agreement with the bank, the supplier receives payment from the bank.

> **Note**
>
> If the buyer and the bank agree to the buyer's validation and authentication before paying the supplier, the supplier will receive the payment after the completion of this validation and authentication.

Also, refer to Charge Reconciliation and Payment – Process Flow [page 240].

SAP Ariba Virtual Card Order Method

To process and send virtual card orders to suppliers, the `SAP Ariba Virtual Card` (`ariba.vcard.ordering.AribaSSPVCard`) order method is used in the SAP Ariba buying solution. In SAP Ariba Buying and SAP Ariba Buying and Invoicing, the line items in the virtual card order are first tested against the `SAP Ariba Virtual Card` order method. If you have other order methods configured in your environment, you can reconfigure them to provide priority to the `SAP Ariba Virtual Card` order method.

> **Note**
>
> If you want to set priority to the `SAP Ariba Virtual Card` order method over other order methods configured in your environment, you can create a ticket with SAP Ariba Customer Support. They will help you set the priority.

The `SAP Ariba Virtual Card` order method accepts a line item if the following conditions are met:

- Line item is targeted at a supplier location that accepts virtual card orders.
- Buyer has an active virtual card account with their bank.
- Buyer has sent the business unit master data to SAP and qualified.
- Buyer has sent the supplier information master data to SAP and qualified.

The `SAP Ariba Virtual Card` order method aggregates line items with same line-item split accounting, virtual card number, transaction number, and transaction date.

Virtual Card Order

In SAP Ariba Buying and SAP Ariba Buying and Invoicing, buyers can create purchase requisitions for catalog or non-catalog items. In the requisition, the buyer can specify a supplier enabled with the virtual card support. After the approval of the requisition, a purchase order is created. Because the supplier added to the purchase order uses the virtual card as the payment method, the purchase order is known as a virtual card order (VCO).

A virtual card order is an approvable document. For more information, refer to Virtual Card Order (VCO)

A field called **Payment Method** is displayed on the purchase order details page for virtual card orders. When you click the **Virtual Card** hyperlink, the virtual card details such as the card number, card status, and card holder name are displayed. The card holder is the buyer whose bank account is configured while setting up the virtual card. For more information, see Setting Up a Buyer Account for a Virtual Card [page 229].

The **Bill To** field of the virtual card order in the supplier's SAP Business Network account carries the following note to inform the suppliers that they can use a virtual card to get the payment for the items they sell to buyers:

You will receive the virtual card number in the email address <supplier's email address> from the network provider directly.

Virtual Card Details for Suppliers

Typically, within 10 minutes after requesting the virtual card details, the supplier receives an email with the details. The buyer's bank and the card provider share the following details of the virtual card with the supplier:

- Virtual card number
- Date of expiration of the virtual card
- Security code of the virtual card
- The amount that the suppliers can charge using the virtual card
- Account number associated with the virtual card
- Whether the supplier can use the virtual card for multiple use
- Date from which the virtual card is valid
- Date until which the virtual card remains valid
- Billing name and address
- Custom data fields
- Description

> ⓘ Note
>
> If the supplier does not receive the virtual card details from the card provider within 12 hours, it is considered a failed virtual card request. Then, the buyers can cancel the virtual card order. In this case, the **Virtual Card Request Failed** action is recorded in the **History** tab of the virtual card order.

Charge Reconciliation and Payment – Process Flow

The following process flow explains the charge reconciliation and subsequent payment to the bank by the buyer:

Using a regular card statement to the buyer, the bank communicates the amount that the supplier charges for the catalog and non-catalog items that they have sold to the buyer.

The buyer creates a .csv file containing the charge details. For more information, refer to .

The buyer runs the **Import Level 2 Charge for Virtual Card Orders** task to import the .csv file to the SAP Ariba buying solution.

The **Process Imported Virtual Card Charges** scheduled task is run to create the charge reconciliation document for the buyer.

The buyer completes the charge reconciliation.

The buyer runs the **Export Virtual Card Charge Payment Requests** export task to export the payment details to the buyer's ERP system.

Charge a Virtual Card and Perform Charge Reconciliation Against the Virtual Card Order

You can import the charge that the bank paid to the supplier using a .csv file. This .csv file must contain the following mandatory information:

- Masked virtual card number against which the bank paid the supplier.
- Number of the virtual card order with which the masked virtual card number is associated.
- The amount that the bank paid to the supplier.

You can use the **Import Level 2 Charge for Virtual Card Orders** task to import the .csv file that contains the charge details to the SAP Ariba buying solution.

A virtual card charge is an approvable document. For more information, refer to Virtual Card Charge (VCC).

After you verify the charge details that you imported to the SAP Ariba buying solution, the **Process Imported Virtual Card Charges** scheduled task is run to create charge reconciliation. You can view the following types of exceptions reported in the charge reconciliation object:

- Exceptions that you have permission to reconcile
- Order mismatch

You can proceed with reconciling these exceptions available in the reconciliation document against the virtual card order. After the successful reconciliation of exceptions, the buyer runs the **Export Virtual Card Charge Payment Requests** export task to export the payment details to the buyer's ERP system. The virtual card charge payment requests containing the following four .csv files are exported to the buyer's ERP system:

File	Description
`CRPaymentHeaderExport.csv`	The charge reconciliation details at the header file.
`CRPaymentDetailExport.csv`	The charge reconciliation details at the payment line items file.
`CRPaymentSplitAccountingExport.csv`	The split accounting information in the line item file.
`CRPaymentBankExport.csv`	The merchant bank information in the payment bank details file.

Alternatively, you can use the **Export Virtual Card Charge Payment Requests in 3 file Format** export task to export the payment details to the buyer's ERP system. Using this task, you can export the following three .csv files:

File	Description
`VCRPaymentHeaderExport.csv`	The charge reconciliation details at the header file.
`CRPaymentDetailWithSplitAccountingExport.csv`	The charge reconciliation details at the payment line items along with split accounting information in the line item details file.

File	Description
`CRPaymentBankExport.csv`	The merchant bank information in the payment bank details file.

For more information, refer to Exporting Virtual Card Charge Payment Requests to the Buyer's ERP system [page 245].

For more information about each virtual card-related action that a user can perform in the SAP Ariba buying solution and their action IDs, refer to Virtual Card Actions.

Importing Charges

This topic describes how to import charge details from the bank statement to the SAP Ariba buying solutions.

Context

A regular card statement that the buyer receives from the bank contains the amount the supplier charges for the items they have sold to the buyer. One of the following occurs:

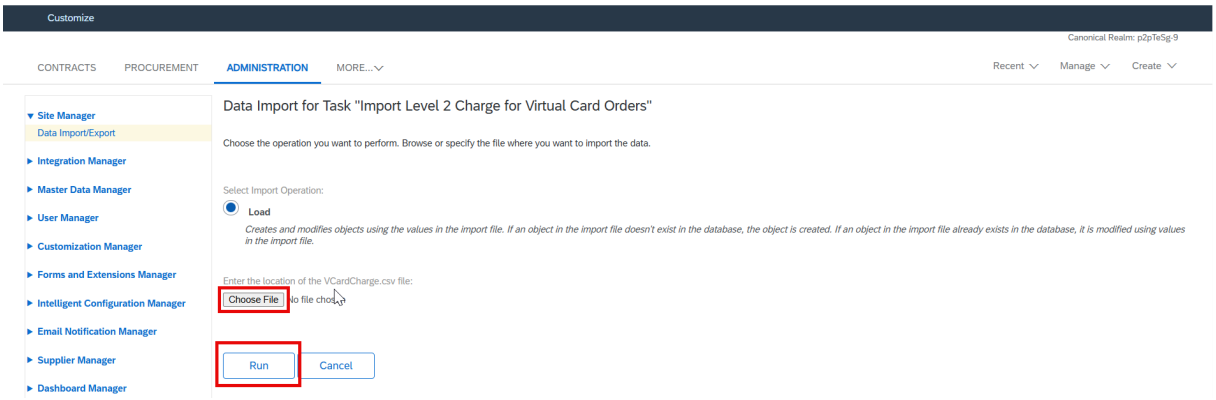
Condition	Bank's Action	Result
The buyer and the bank agree to the buyer's validation and authentication before paying the supplier	The bank first share the statement with the buyer	The reconciliation occurs before the supplier receives the payment from the bank.
The buyer and the bank agree to the buyer's validation and authentication with the bank after paying the supplier	The bank first make payment to the supplier.	The bank shares the statement with the buyer after paying the supplier. The reconciliation occurs after paying the supplier.

After receiving the card statement from the bank, the buyer can import charges to the SAP Ariba buying solution by using a .csv file.

Procedure

1. Create a .csv file that contains the following mandatory information based on the credit card statement from the bank:
   - Masked virtual card number against which the bank paid the supplier.
   - Number of the virtual card order with which the masked virtual card number is associated.
   - The amount that the bank paid to the supplier.
2. On the dashboard, click **Manage** > **Core Administration** .

3. Click **Site Manager** > **Data Import/Export**.
4. On the **Data Import/Export** page, do one of the following:
   - In the **Task Name** field in the **Search Filters** area, provide the name of the task (**Import Level 2 Charge for Virtual Card Orders**) and click **Search**.
   - Under **Import**, in the list of import tasks, locate the **Import Level 2 Charge for Virtual Card Orders** task.
5. On the right side of the task, click **Import**.
6. In the **Data Import for Task "Import Level 2 Charge for Virtual Card Orders"** page, browse and select the .csv file for importing the data based on the card statement from the bank.



7. Click **Run**.
8. In the **Confirm Data Import** dialog box to confirm the running of the import task, click **OK**.
   On completion of the import task, the status of the task changes to **Completed**.

Viewing the Details of Virtual Card Charges

Context

You can view the details of the virtual card charges that you have imported to the SAP Ariba buying solution.

Procedure

1. On the **Home** page of the SAP Ariba buying solution, in the search drop-down list, choose **Virtual Card Charge** and click the search icon.
2. On the **Search** page, enter the required filter values to search for the virtual card charge and click **Search**.
3. In the search results, click the ID of the virtual card charge record.
4. In the details page of the virtual card charge record, view the details such as header charges, line item charges, comments, and attachments.
5. To view the details of the charge line item, click **Details**.

Creating the Virtual Card Charge Reconciliation Document

Context

To create the virtual card reconciliation document, the **Process Imported Virtual Card Charges** scheduled task is run after importing the virtual card charges.

Procedure

1. After the **Process Imported Virtual Card Charges** scheduled task is run, navigate to the **Reconciliation** tab of the imported charge.
   You can view the charge reconciliation record.
2. Click the ID of the charge reconciliation object to view its details.
3. In the **Exceptions** tab, view the exceptions available with the charge reconciliation document.

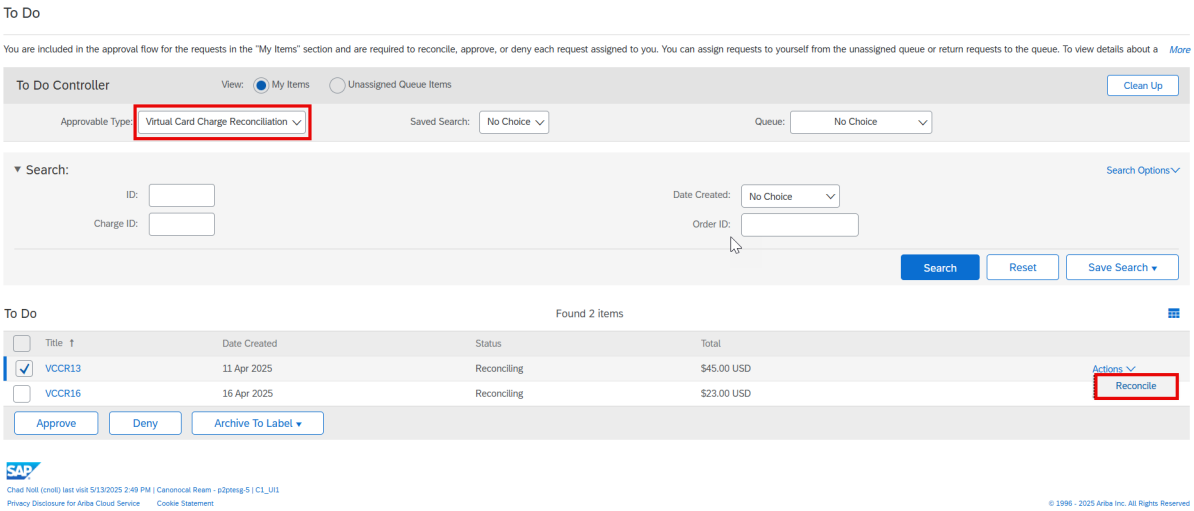
Reconciling Charges

Context

This topic describes how to reconcile virtual card charges in the SAP Ariba buying solution.

Procedure

1. On the left side of the **Home** page of the SAP Ariba buying solution, in the **Common Actions** area, click **Reconcile Charges**.
2. In the **To Do Controller** section of the **To Do** page, in the **Approvable Type** drop-down list, choose **Virtual Card Charge Reconciliation**.
3. In the **To Do** section, on the right side of the charge reconciliation record, click **Actions** > **Reconcile**.



4. On the page that displays the details of the charge reconciliation record, click **Update Exceptions**.
   The exceptions available on the charge reconciliation documents are cleared.

Exporting Virtual Card Charge Payment Requests to the Buyer's ERP system

Context

After reconciling virtual card charges, the charge payment request is created. Then, the charge payment request is exported to the buyer's ERP system.

Procedure

1. On the dashboard, click **Manage** > **Core Administration**.
2. Click **Site Manager** > **Data Import/Export**.
3. On the **Data Import/Export** page, do one of the following:
   - In the **Task Name** field in the **Search Filters** area, provide the name of the task (**Export Virtual Card Charge Payment Requests**) and click **Search**.
   - Under **Export**, in the list of export tasks, locate the **Export Virtual Card Charge Payment Requests** task.
4. On the right side of the task, click **Export**.
5. In the **Export Virtual Card Charge Payment Requests** dialog box, select the required dates and click **OK**.

Ad Hoc Vendors for Non-Catalog Items in Virtual Card Orders

Buyers can create virtual card orders to procure non-catalog items from ad hoc vendors. Ad hoc vendors are suppliers added to SAP Ariba Buying and SAP Ariba Buying and Invoicing on an as-needed basis without needing to onboard them. Ad hoc vendors receive virtual card details on the email address of their vendor location. They can charge the virtual card to accept payment for the non-catalog items they sell to buyers.

To add ad hoc vendors to requisitions for non-catalog items, identify the vendors existing in the SAP Ariba buying solution that you want to designate as ad hoc vendors and run the **Import Ad Hoc Vendor** task.

For more information, refer to Importing Ad Hoc Vendors to SAP Ariba Buying Solutions [page 247].

You can select ad hoc vendors and add and/or select their vendor locations on purchase requisitions for non-catalog items. Then, approvers verify and approve these requisitions to create virtual card orders (VCO).

> **Note**
>
> For an ad hoc vendor, you must select Email as the preferred ordering method. No other preferred ordering methods are supported for ad hoc vendors.

Before enabling virtual card support for ad hoc vendors, you must ensure that the ad hoc vendors under the partition supplier adhere to the following conditions:

- Ad hoc vendors are not registered with SAP Business Network and do not have an Ariba Network ID (ANID).
- Ad hoc vendors are not associated with any catalog item.
- Ad hoc vendors are not associated with any existing order. You can associate ad hoc vendors with virtual card orders only.
- Vendor locations of ad hoc vendors must be configured with email as the preferred ordering method.

Prerequisites

- Enable the **Use single-use virtual cards to pay suppliers** (`Application.Common.SupportsvCardBasedOrder`) parameter in your environment.
- Enable the **Use ad hoc vendors for non-catalog items** (`Application.Common.UseAdHocVendorsForNonCatalogItem`) parameter in your environment.
- Ensure that the ad hoc vendors are enabled with virtual card support.

Related Information

Deactivating Supplier Locations Associated with Ad hoc Vendors [page 250]

Process for Supporting Ad Hoc Vendors for Non-Catalog Items in Virtual Card Orders

This topic provides the process for supporting ad hoc vendors for non-catalog items in virtual card orders.

1. Perform all the prerequisite tasks listed in the **Prerequisites** section on the Ad Hoc Vendors for Non-Catalog Items in Virtual Card Orders [page 246] topic.
2. Import ad hoc vendors to the SAP Ariba buying solution. For more information, refer to Importing Ad Hoc Vendors to SAP Ariba Buying Solutions [page 247].
3. Start a purchase requisition for a non-catalog item.
4. Add an ad hoc vendor enabled with virtual card support to the requisition and specify a vendor location.
5. Approve the requisition to create a virtual card order based on it.
6. The virtual card order is sent to SAP Business Network by the buyer. Then, the ad hoc vendor is informed about the virtual card order through an email.
7. The ad hoc vendor receives the virtual card order.
8. The buyer sends a request for the virtual card to SAP. Then, SAP contacts the card provider, and the card provider contacts the buyer's bank. Then, the card provider shares the virtual card number for the virtual card order with the supplier.
   In the **History** tab of the virtual card order, the **Virtual Card Requested** action is recorded.
9. The **Retrieve masked Virtual Card Number** scheduled task is run in the SAP Ariba buying solution to associate the virtual card number with the virtual card order.
10. The card provider sends an email that contains the complete card number, expiry date, and security code to the supplier.
    In the **History** tab of the virtual card order, the **Virtual Card Received** action is recorded.

Importing Ad Hoc Vendors to SAP Ariba Buying Solutions

Context

You can run the **Import Ad Hoc Vendor** task to import suppliers existing in the SAP Ariba buying solution as ad hoc vendors.

A .csv file is used to import the system ID of the designated ad hoc vendor. The following is an example of this .csv file:

Sample Code

```
SystemID
"464684740"
```

Procedure

1. Create a .csv file that contains the system ID of the suppliers to be designated as ad hoc vendors enabled with virtual card support.
2. In the SAP Ariba buying solution, navigate to **Manage** > **Core Administration**.
   The page with the core administration options is displayed.
3. On the left, under **Site Manager**, click **Data Import/Export**.
4. On the **Data Import/Export** page, locate the **Import Ad Hoc Vendor** task. On the right side of the task, click **Import**.
5. On the **Data Import for Task "Import Ad Hoc Vendor"** page, click **Choose File**.
6. Navigate to the .csv file containing the system ID of the designated ad hoc vendors and click **Open**.
7. Click **Run**.
8. On the message box to confirm the running of the **Import Ad Hoc Vendor** task, Click **OK**.

   > **Note**
   >
   > The imported ad hoc vendor details, including their supplier locations, cannot be modified.

9. On the **Data Import/Export** page, ensure that the status of the **Import Ad Hoc Vendor** task is displayed as **Completed**.

Adding an Ad Hoc Vendor and Vendor Location in a Purchase Requisition

Context

You can add an ad hoc vendor and the vendor location while creating a purchase requisition for a non-catalog item.

Procedure

1. In SAP Ariba Buying and Invoicing, navigate to **Create** > **Create Requisition**.
2. On the **Create Requisition** page, under the question **What would you like to create?**, click **Requisition**.
3. On the right side of the search field, click the **Add a non-catalog item** icon.
4. On the **Create Non-Catalog Item** page, in the **Supplier Information** section, do the following:

Field	Action
Use Ad Hoc Vendor	Select the check box. This option is visible only if you belong to the **Ad Hoc Vendor Agent** group. **Note** In an environment where the `Application.Approvable.OrganizationalVisibilityPartitioning` parameter is enabled, users belonging to the **Ad Hoc Vendor Agent** group can use an ad hoc vendor when adding line items to a requisition even though they don't have the specific purchasing unit assigned to the **Ad Hoc Vendor Agent** group in their profile.
Vendor	• In the drop-down list, click **Search more**. • In the **Choose Value for Vendor** dialog box, identify the ad hoc vendor and click **Select**. • Click **Done**.
Contact	• Click **Add Contact**. • On the **Add Ad Hoc Vendor Contact** dialog box, enter the email address of the vendor location and click **Search**. • If already configured, the details of the vendor location is displayed. Otherwise, you can enter the required details to configure the vendor location. • Click **Add**. **Note** Email ID can't contain the plus symbol (+).
Payment Terms	• Click **Select**. • On the **Choose Value for Payment Terms** dialog box, identify the payment term and click **Select**. • Click **Done**.
Supplier Part Number	Enter the supplier part number of the ad hoc vendor you have selected.
Supplier Part Auxiliary ID	Enter the supplier part auxiliary ID of the ad hoc vendor you have selected.

5. Click **Add to Cart** to proceed with the creation of the virtual card order.

Note

A requisition must contain only ad hoc items or only catalog items.

Deactivating Supplier Locations Associated with Ad hoc Vendors

You can run the **Deactivate supplier locations associated with ad hoc vendors** import task to deactivate supplier locations associated with ad hoc vendors to ensure that the email IDs of deactivated supplier locations can't be reused. When users try to use the email ID of the deactivated supplier location while creating a contact for an ad hoc vendor, the SAP Ariba solution displays an error message to indicate that the supplier is not an ad hoc vendor.

Update the `AdhocVendorSupplierLocationAnonymizePull.csv` file with information about the suppliers you want to deactivate. In the file, the `SystemID` represents the supplier ID of the ad hoc vendor and `vCardEmailAddress` represents an email address that was previously created as an ad hoc vendor contact for a virtual card order.

Following is an example of the `AdhocVendorSupplierLocationAnonymizePull.csv` file.

```
"UTF-8"
SystemID, vCardEmailAddress
"12345", "Vendor1@Organization1.com"
"34567", "Vendor2@Organization2.com"
"45678", "Vendor2@Organization3.com"
```

Charge Reconciliation

Charge reconciliation is the process of verifying a bank's charge data against the numbers for those charges in SAP Ariba Buying solutions.

Charge reconciliation tasks are available in SAP Ariba Buying and Invoicing. They are also available in SAP Ariba Buying when Receiving is enabled. The ability to reconcile charges is enabled by default through the `Application.Charge.Enabled` parameter.

The purpose of the reconciliation process is to review the bank's listed charges and make sure they are correct.

The buying solution handles charge reconciliation with a combination of automatic and manual review of charge statements as follows:

- During automatic reconciliation, the buying solution attempts to match the charge to a purchase order and reconcile the charge.
- If the buying solution cannot match a charge to a purchase order, it presents the charge in the user interface for manual assignment.
  - Charges associated with procurement cards must be matched to a purchase order or marked as non-purchase order charges.
  - Charges associated with combination cards can be treated as purchases or expenses.
  - Charges associated with expense cards can only be assigned to expense reports.
- If the automatic reconciliation process detects any discrepancies, the buying solution initiates the manual reconciliation phase and records those discrepancies as charge exceptions on a purchasing card charge reconciliation (PCCR) document.
- The buying solution routes PCCR documents to approvers using the standard approval flow mechanism. The approvers review the exceptions, suggest corrections, and then submit the adjusted PCCR for further approval.

Skip Validation Feature for Charge Reconciliation [page 251]

Auto-Accept Feature for Charge Reconciliation [page 252]

Charge Exception Types [page 252]

Data Import Tasks for Charge Exception Types [page 253]

Import Charge Exception Types Task [page 253]

Import Charge Exception Type Validation Data Task [page 254]

Adjustment Types [page 255]

Skip Validation Feature for Charge Reconciliation

The automatic validation process checks to see if the validation of a PCCR document can be skipped. The buying solution skips validation if the total amount of the PCCR document is below a certain amount.

In the default configuration, the skip amount is set to 5, which represents an amount in the default currency for your site.

If you want to change the default skip amount, contact SAP Ariba Customer Support.

Auto-Accept Feature for Charge Reconciliation

The buying solution accepts a charge automatically if the header amount is less than a specified amount or less than a specified percentage of the original order.

In the default configuration, the auto-accept amount is set to 10, which represents an amount in the default currency for your site, and the auto-accept percentage is set to 15 percent.

If you want to change the default auto-accept amount or percentage, contact SAP Ariba Customer Support.

Charge Exception Types

A charge exception represents a single discrepancy between a bank's charge data and the data in the buying solution.

PCCR documents that require reconciliation or approval appear in a business user's **To Do** list. Business users assign and reconcile charges that the buying solution could not match to orders during the automatic reconciliation phase.

In the default configuration, each charge exception type is associated with a group. The buying solution uses that group to determine the list of users on the approval flow for each PCCR document. For example, in the default configuration, the "Unmatched PO" exception is associated with the **PCard Manager** group.

The default approval rules use the groups associated with exception types to determine the set of approvers. The approval rules look over the PCCR document, find all exceptions on that document, and define as approvers those users who have permission to reconcile the specified exceptions. These approvers are typically the first approvers on the flow, and are the users who are expected to reconcile any listed exceptions.

Depending on your configuration, if the user specified as the PCard charge reconciler is allowed to reconcile a particular charge exception type, then that user is added as an approver instead of the group for that particular exception. For the exception types that the user is not authorized to reconcile, the group is added to the approval flow.

The buying solution includes a standard set of default charge exception types.

Related Information

Approval Process Management Guide
Reference Data Workbooks
Purchasing Guide for Procurement Professionals

Data Import Tasks for Charge Exception Types

There are two data import tasks for charge exception types: `Importing Charge Exception Types` and `Importing Charge Exception Type Validation Data`.

The following table lists the data import tasks and CSV files that define charge exceptions and specifies whether the data is required. Tasks appear in the recommended data import order.

Data Import Task	CSV Files Imported	Required
`Import Charge Exception Types`	`ChargeExceptionType.csv`	No
Import Charge Exception Type Validation Data	`ChargeExceptionTypeValidation.csv`	No

You manually enter data in the CSV files.

Related Information

About Importing Charge Exception Types [page 253]
About Importing Charge Exception Type Validation Data [page 254]

Import Charge Exception Types Task

You use the `Import Charge Exception Types` to define your own set of charge exception types. This task reads data from the CSV file named `ChargeExceptionType.csv`.

The following example of `ChargeExceptionType.csv` shows the default charge exception types:

```
UTF8


UniqueName,Name,Description,Permission,Group,RequesterAllowed,PCardReconcilerAllowed

UnmatchedPO,PO Unmatched,Ariba Charge is unable to match to the associated pcard
order, {0}.,PCard Manager,No,Yes
CancelledPOException,Cancelled PO Exception,Ariba charge is matched to the pcard
order, {0}, which is cancelled,PCard Manager,No,No
POAmountVariance,PO Amount Variance,The total amount of the charge, {1}, is greater
than the total amount of the pcard order, {0}.,PCard Manager,No,Yes
```

Translations for Charge Exception Types

You can use the `Import Charge Exception Type Translations` data import task to import translations for charge exception type names and descriptions.

For example:

```
UTF8

     UniqueName,Name,Description,Language

UnmatchedPO,Unmatched PO,Order {0} not found.,English
```

Import Charge Exception Type Validation Data Task

You use the `Import Charge Exception Type Validation Data` data import task to define your own charge exception type validation data. Charge exception type validation data defines exactly which fields to compare and what tolerances to allow for each type of charge exception.

The `Import Charge Exception Type Validation Data` data import task reads data from the CSV file named `ChargeExceptionTypeValidation.csv`.

The following example of `ChargeExceptionTypeValidation.csv` shows the default charge exception validation data:

```
UTF8


UniqueName,HeaderOnly,InvoiceCategory,LineCategory,LineType,FieldPathToValidate,Fiel
dPathTarget,
FieldPathToValidateAgainst,FieldPathSourceOnDispute,NullAllowed,ToleranceOperation,
AbsoluteTolerance,PercentageTolerance,AbsoluteRank,RelativeRank,AutoReject,Quantifia
bleOnly,
SpendCategoryDriver,SpendCategory,IgnoreAutoAccept

UnmatchedPO,TRUE,0,0,,,,IsMatched,Charge.SupplierOrderInfo.OrderNumber,FALSE,boolean
,0,0,1,1,
FALSE,FALSE,,,FALSE
CancelledPOException,TRUE,0,0,,,,IsActivePO,Order.OrderID,FALSE,boolean,0,0,1,2,FALS
E,FALSE,,,
FALSE
POAmountVariance,TRUE,3,0,,Order.AmountInvoiced,TotalCost,Order.AmountOrderedInvoice
able,
POAmountOrderedLessPreviouslyInvoiced,FALSE,or,10,1.2,2,1,FALSE,FALSE,,,FALSE
```

Charge Exception Type Validation Data to Avoid PO Variance Exceptions

In the default configuration, the charge exception type validation data that you import allows for all the out-of-the-box exception types to be displayed on the user interface.

If you do not want the PO Amount Variance exception to occur and be displayed, you must import the `ChargeExceptionTypeValidation.csv` file with the following data set:

```
     Cp1252


UniqueName,HeaderOnly,InvoiceCategory,LineCategory,LineType,FieldPathToValidate,Fiel
dPathTarget,FieldPathToValidateAgainst,FieldPathSourceOnDispute,NullAllowed,Toleranc
eOperation,AbsoluteTolerance,PercentageTolerance,AbsoluteRank,RelativeRank,AutoRejec
t,QuantifiableOnly,SpendCategoryDriver,SpendCategory,IgnoreAutoAccept

UnmatchedPO,TRUE,0,0,,,,IsMatched,Charge.SupplierOrderInfo.OrderNumber,FALSE,boolean
,0,0,1,1,FALSE,FALSE,,,FALSE
CancelledPOException,TRUE,0,0,,,,IsActivePO,Order.OrderID,FALSE,boolean,0,0,1,2,FALS
E,FALSE,,,FALSE
POAmountVariance,TRUE,3,0,,TotalLIAmountLessTaxAndAdjustment,TotalLIAmountLessTaxAnd
Adjustment,POAmountOrderedLessPreviouslyInvoiced,POAmountOrderedLessPreviouslyInvoic
ed,FALSE,or,10,1.2,2,1,FALSE,FALSE,,,FALSE
InvalidAccountingInformation,TRUE,0,0,,,,HasAccountingDefaulted,,FALSE,boolean,0,0,2
,1,FALSE,FALSE,,,FALSE
```

This configuration ensures the PO Amount Variance exception does not occur.

> **Note**
>
> We recommend that you make a copy of the out of the box `ChargeExceptionTypeValidation.csv` file and import this, if you want to revert to the default behavior at any time in the future.

Adjustment Types

When there is a discrepancy in purchasing card records, the buying solution offers the user creating the adjustments (either the cardholder or another administrator, depending on your approval rules) a menu of choices to explain the discrepancy.

Each configuration defines a set of adjustment types that populate this menu. For example, you might set up “taxes” as an adjustment category, to indicate that a discrepancy was the result of a tax computation.

For descriptions of the CSV files read by the integration events, see the Data Dictionary.

Data Import Tasks for Adjustment Types

There are two data import tasks for adjustment types: `Import Adjustment Types` and `Import Adjustment Type Translations`, both of which import `AdjustmentType.csv`.

The following table lists the data import tasks and CSV files that are imported for the adjustment types:

Data Import Task	CSV Files Imported
`Import Adjustment Types`	`AdjustmentType.csv`
`Import Adjustment Type Translations`	`AdjustmentType.csv`

Import Adjustment Type Task

The `AdjustmentTypePull` integration event (display name "Import Adjustment Type") defines the available adjustment types. The buying solution provides this integration event, which reads data from a CSV file. This integration event loads ERP-specific data. In the default configuration, it reads data from the `AdjustmentType.csv` CSV file.

For example, the sample `AdjustmentType.csv` file in the default configuration includes the following lines:

```
"UTF-8"

     UniqueName,Name,Description,Category,Account

"Default","Sales Tax","Adjustment: Sales Tax","standard","7710"
"Shipping","Shipping/Handling","Adjustment: Shipping & Handling","standard","7752"
```

> **Note**
>
> - If you have a multilingual configuration, you can use the `Import Adjustment Type Translations` to define the translated versions of adjustment types.
> - You can use the `Export Adjustment Type` export task to export adjustment type data to a CSV file.

Adjustment Types in Ariba Administrator

You can use the Adjustment Types task in the PCard Manager workspace in Ariba Administrator to view, add, modify, and delete adjustment types.

In the default configuration, the buying solution provides the following adjustment types:

- Sales Tax
- Shipping/Handling

As a user associated with the **PCard Manager** group, you can choose to edit the existing adjustment types or create new adjustment types, as required.

Charge Payment Requests in SAP Ariba Buying and Invoicing

After a PCard charge has been approved, the final step is to send a payment to the card vendor. When the vendor creates a PCard charge reconciliation (PCCR) document, SAP Ariba Buying and Invoicing also creates a new approvable document, called a charge payment request, to track the payment process.

> **Note**
>
> Standard export data are enhanced periodically to include more fields. When added, new fields are always inserted at the end of the record in all exported CSV files. Customers who implement further integrations with exported CSV files must ensure that their integrations are robust enough to ignore these newly added fields if they are not required.

About Payment Models and Remittance Import

SAP Ariba Buying and Invoicing exports charge payment requests that use the External System payment model only. The External System payment model specifies that payments are handled in an external system, such as an ERP system.

SAP Ariba Buying and Invoicing does not export charge payment requests that use the SAP Business Network payment model. If the payment model is set to SAP Business Network, the final state for charge payment requests is Scheduling.

> **Note**
>
> You cannot import remittance information about charge payments made through an external system into SAP Ariba Buying and Invoicing.

Charge Payment Request Data Export Tasks [page 257]

Charge Payment Request Data for PeopleSoft ERP [page 259]

Charge Payment Request Data for SAP ERP [page 263]

Charge Payment Request Data for Generic ERP [page 266]

Charge Payment Request Data Export Tasks

You can export payment requests (also referred to as OK-to-Pay invoice data) by running data export tasks in Ariba Administrator.

The `Export Charge Payment Requests` data export task exports data to four CSV files.

(Generic only) The `Export Charge Payment Requests in 3 File Format` data export task, which exports data to three CSV files.

SAP Ariba provides the `Export Charge Payment Requests in 3 File Format` data export task for ERP systems, such as Oracle, that cannot handle line-item details and split-accounting information in separate files.

> **Note**
>
> You cannot use the SAP Ariba data transfer tool to export charge payment requests.

Adding Custom Accounting Fields to Charge Payment Export

You can add custom accounting fields to be included in the charge payment export data, similar to the custom fields that you can add to Payment Export.

Procedure

1. On the dashboard, click **Manage** > **Core Administration**.
2. Click **Customization Manager** > **Field Configuration**.
3. In the **Browse Classes** field, search for Split Accounting.
4. Select the field to be added and click **Edit**.
5. Click the **Advanced** tab.
6. Select the **Include in Charge Payment Exports** and click **OK**.

Results

When you run the export charge payment data export task from Ariba Administrator, the custom fields you included are listed in the following files:

- `CRPaymentSplitAccountingExport.csv` file used by the `Export Charge Payment Requests` task for PeopleSoft ERP
- `CRPaymentDetailWithSplitAccountingExport.csv` file used by the `Export Charge Payment Requests in 3 File Format` task for Generic ERP
- `CRPaymentDetailExport.csv` file used by the `Export Charge Payment Requests` task for SAP ERP

Exporting Charge Payment Requests

You can export charge payment requests from Ariba Administrator.

Procedure

1. On the dashboard, click **Manage** > **Core Administration** .
2. Click **Site Manager**.
3. Click **Data Import/Export**, and click the **Export** tab.
4. Search for the `Export Charge Payment Requests` data export task and click **Export**.
5. Select a date range for the export by choosing dates in the **From** and **To** fields. Ariba Administrator exports all charge payment requests created during this time period.

   The first time you export the data, the **From** text box is empty.
6. Click **Run** to export the data.

   Only charge payment requests that are in the "Scheduled" state are exported.
7. After the data has finished downloading, click **Open** to view the contents of the `CRPaymentExport.zip` file or click **Save** to save the file to your local hard drive.

Charge Payment Request Data for PeopleSoft ERP

When you run `Export Charge Payment Requests` data export task from Ariba Administrator, the data is stored in a ZIP file containing four CSV files.

The CSV files included in the ZIP file are:

- `CRPaymentHeaderExport.csv`
- `CRPaymentDetailExport.csv`
- `CRPaymentBankExport.csv`
- `CRPaymentSplitAccountingExport.csv`

Related Information

Charge Payment Header Data for PeopleSoft ERP [page 260]
Charge Payment Detail Data for PeopleSoft ERP [page 261]
Charge Payment Bank Data for PeopleSoft ERP [page 262]
Charge Payment Split Accounting Data for PeopleSoft ERP [page 263]

Charge Payment Header Data for PeopleSoft ERP

The `CRPaymentHeaderExport.csv` file provides header data for charge payment.

The following table describes the fields in the `CRPaymentHeaderExport.csv` file:

Field Name	Description	Format
PAYMENTID	The unique reference ID of the payment.	String (50)
REF_DOC_NO	The unique name of the charge.	String (50)
TRANSACTION_DT	The charge transaction date.	Date
PAYEEBANK	The unique internal identifier for the buyer bank payment location.	String (50)
BANK_ACCT_KEY	The bank account type for the buyer bank payment location, for example, `chec` (checking).	String (50)
BANK_CD	The unique code number of the bank.	String (50)
GROSS_AMOUNT	The gross amount of the payment. Negative (credit) payments include a minus sign (-) before the amount.	Numeric
CURRENCY	The currency for the total cost specified in the PCCR.	String (3)
ITEM_TEXT	The unique name of the PCCR, for example, "PCCR10."	String (50)
BUSINESS_UNIT	The unique internal identifier of the PeopleSoft business unit.	String (5)
DUE_DATE	The payment net due date.	Date
SUPPLIER_ID	The supplier ID loaded in the charge.	String (50)
SUPPLIER_NAME	The supplier name that corresponds to the supplier loaded in the charge, if a matching supplier is found. Otherwise, it is the supplier name loaded in the charge.	String (50)
PCARD_NUMBER	The masked purchasing card number specified in the PCCR.	Encrypted String
PCARD_ORDER	The unique name of the order specified in the PCCR.	String (50)
PCARD_HOLDER	The name of the purchasing card holder.	String (50)
ADAPTER_SOURCE	The adapter source for the charge.	String (50)

Following is an example of `CRPaymentHeaderExport.csv`:

```
PAYMENTID,REF_DOC_NO,TRANSACTION_DT,PAYEEBANK,BANK_ACCT_KEY,BANK_CD,GROSS_AMOUNT,CUR
RENCY,
ITEM_TEXT,BUSINESS_UNIT,DUE_DATE,SUPPLIER_ID,SUPPLIER_NAME,PCARD_NUMBER,PCARD_ORDER,
PCARD_HOLDER,ADAPTER_SOURCE

"PAY77","PCC77","02/15/2008 00:00:00 -0800","USA BANK","chec","12104","155.5000",
"USD","PCCR77","US003",,"sid496","JCN Technologies","XXXX-XXXX-XXXX-3601",,
"AAAeAEyEhFtya","External"
"PAY78","PCC78","02/15/2008 00:00:00 -0800","USA BANK","chec","12104","234.5000",
"USD","PCCR78","US003",,"sid496","JCN Technologies","XXXX-XXXX-XXXX-3601",,
"AAAeAEyEhFtya","External"
"PAY79","PCC79","02/15/2008 00:00:00 -0800","USA BANK","chec","12104","170.0000",
"USD","PCCR79","US003",,"sid496","JCN Technologies","XXXX-XXXX-XXXX-3601",,
"AAAeAEyEhFtya","External"
"PAY73","PCC73","02/15/2008 00:00:00 -0800","USA BANK","chec","12104","34.0000",
"USD","PCCR73","US001",,"sid496","sid496","XXXX-XXXX-
```

```
XXXX-3601","PCO11","AAAeAEyEhFtya",
"External"
```

Charge Payment Detail Data for PeopleSoft ERP

The `CRPaymentDetailExport.csv` file provides line-level details for charge payment.

The following table describes the columns in the `CRPaymentDetailExport.csv` file:

Column	Description	Format
`PMNT_ID`	The unique reference ID of the payment. This value matches a payment defined in `CRPaymentHeaderExport.csv`.	String (50)
`ECQUEUEINSTANCE`	The unique name of the PCCR, for example "PCCR77."	String (50)
`BUSINESS_UNIT`	The unique internal identifier of the PeopleSoft business unit specified in the PCCR.	String (50)
`VOUCHER_LINE_NUM`	The line number of the charge on the voucher.	Numeric
`PARENT_LINE_NUM`	The voucher line number of the parent line for the charge line item.	Numeric
`LINE_TYPE_LOOKUP_CODE`	The line category. Line categories separate procurement line items from the various special handling categories, such as tax and freight. The line categories are as follows: • 2 - Tax Charge (for all tax-related charges) • 4 - Freight Charge (for all freight-related charges) • 8 - Handling Charge (for all handling-related charges) • 32 - Discount (for discounts related to invoicing)	Numeric
`DESCR`	The description of the charge line item.	String (2000)
`AMOUNT`	The currency amount of charge line item.	Numeric
`UNIT_PRICE`	The unit price of the charge line item.	Numeric
`CURRENCY_CD`	The currency code for the charge line item amount.	String (50)

Following is an example of `CRPaymentDetailExport.csv`:

```
PMNT_ID,ECQUEUEINSTANCE,BUSINESS_UNIT,VOUCHER_LINE_NUM,PARENT_LINE_NUM,LINE_TYPE_LOO
KUP_CODE,
DESCR,AMOUNT,UNIT_PRICE,CURRENCY_CD

"PAY77","PCCR77","US003","1",,,"Charge line","155.5000","155.5000","USD"
"PAY78","PCCR78","US003","1",,,"Charge line","234.5000","234.5000","USD"
"PAY79","PCCR79","US003","1",,,"Charge line","170.0000","170.0000","USD"
"PAY73","PCCR73","US001","1",,,"Charge line","34.0000","34.0000","USD"
```

Charge Payment Bank Data for PeopleSoft ERP

The `CRPaymentBankExport.csv` file includes bank data for charge payment.

The following table describes the columns in the `CRPaymentBankExport.csv` file:

Column	Description	Format
`PMNT_ID`	The unique reference ID of the payment. This value matches a payment defined in `CRPaymentHeaderExport.csv`.	String (50)
`ECQUEUEINSTANCE`	The unique name of the PCCR, for example "PCCR77."	String (50)
`BUSINESS_UNIT`	The unique internal identifier of the PeopleSoft business unit specified in the PCCR.	String (50)
`PYMNT_METHOD`	The unique internal identifier of the payment method. A payment method defines how a buying organization makes a payment to a supplier.	String (50)
`BANK_ACCT_KEY`	The bank account key.	String (50)
`COMPOSITE_ACCT_KEY`	The unique internal identifier of the bank payment location configured in the buying solution.	String (50)
`SCHEDULED_PAY_DT`	The date of the scheduled payment.	Date
`BANK_CD`	The unique code number of the bank.	String (25)
`PAYMENT_AMT`	The payment amount.	Numeric
`PAYMENT_GROSS_AMT`	The payment gross amount.	Numeric
`PAYMENT_DISC_AMT`	The payment discount amount.	Numeric
`PAYMENT_CURRENCY_CD`	The transaction currency code for the payment.	String (50)

Following is an example of `CRPaymentBankExport.csv`:

```
PMNT_ID,ECQUEUEINSTANCE,BUSINESS_UNIT,PYMNT_METHOD,BANK_ACCT_KEY,COMPOSITE_ACCT_KEY,
SCHEDULED_PAY_DT,BANK_CD,PAYMENT_AMT,PAYMENT_GROSS_AMT,PAYMENT_DISC_AMT,PAYMENT_CURR
ENCY_CD

"PAY77","PCCR77","US003",,"chec","BuyerBankLocation-1","02/15/2008 08:52:47
-0800","TRBNK",
"155.5000","155.5000","0.0000","USD"
"PAY78","PCCR78","US003",,"chec","BuyerBankLocation-1","02/15/2008 08:52:47
-0800","TRBNK",
"234.5000","234.5000","0.0000","USD"
"PAY79","PCCR79","US003",,"chec","BuyerBankLocation-1","02/15/2008 08:52:47
-0800","TRBNK",
"170.0000","170.0000","0.0000","USD"
"PAY73","PCCR73","US001",,"chec","BuyerBankLocation-1","02/15/2008 08:52:46
-0800","TRBNK",
"34.0000","34.0000","0.0000","USD"
```

Charge Payment Split Accounting Data for PeopleSoft ERP

The `CRPaymentSplitAccountingExport.csv` file includes split accounting details for charge payment.

The following table describes the columns in the `CRPaymentSplitAccountingExport.csv` file:

Column	Description	Format
`PMNT_ID`	The unique reference ID of the payment. This value matches a payment defined in `CRPaymentHeaderExport.csv`.	String (50)
`ECQUEUEINSTANCE`	The unique name of the PCCR, for example "PCCR77."	String (50)
`BUSINESS_UNIT`	The unique internal identifier of the PeopleSoft business unit specified in the PCCR.	String (50)
`VOUCHER_LINE_NUM`	The charge line number on the voucher.	Numeric
`DISTRIB_LINE_NUM`	The distribution line number.	Numeric
`BUSINESS_UNIT_GL`	The unique internal identifier of the PeopleSoft general ledger business unit specified in the PCCR.	String (50)
`ACCOUNT`	The account ID.	String (20)
`DEPTID`	The PeopleSoft department ID.	String (20)
`QTY_VCHR`	The quantity of the charge line item.	Numeric
`AMOUNT`	The currency amount of the charge line item.	Numeric

Following is an example of `CRPaymentSplitAccountingExport.csv`:

```
PMNT_ID,ECQUEUEINSTANCE,BUSINESS_UNIT,VOUCHER_LINE_NUM,DISTRIB_LINE_NUM,BUSINESS_UNI
T_GL,ACCOUNT,DEPTID,QTY_VCHR,AMOUNT

"PAY77","PCCR77","US003","1","1","US003",,"14000","1.0000","155.5000"
"PAY78","PCCR78","US003","1","1","US003",,"14000","1.0000","234.5000"
"PAY79","PCCR79","US003","1","1","US003",,"14000","1.0000","170.0000"
"PAY73","PCCR73","US001","1","1","US001","131000","14000","1.0000","34.0000"
```

Charge Payment Request Data for SAP ERP

When you run the `Export Charge Payment Requests` data export task in Ariba Administrator it exports data in a ZIP file containing two CSV files: `CRPaymentHeaderExport.csv`, and `CRPaymentDetailExport.csv`.

Related Information

Charge Payment Header Data for SAP ERP [page 264]
Charge Payment Detail Data for SAP ERP [page 265]

Charge Payment Header Data for SAP ERP

The `CRPaymentHeaderExport.csv` file includes data for charge payment header data.

The following table describes the fields in the `CRPaymentHeaderExport.csv` file:

Field Name	Description	Format
PAYMENTID	The unique reference ID of the payment.	String (50)
REF_DOC_NO	The unique name of the charge.	String (50)
BANK_ACCT_KEY	The bank account type for the buyer bank payment location, for example, `chec` (checking).	String (4)
PAYEEBANK	The unique internal identifier of the bank payment location in your buying solution.	String (50)
BANK_CD	The unique code number of the bank.	String (50)
GROSS_AMOUNT	The gross amount of the payment.	Numeric
CURRENCY	The currency for the total cost specified in the PCCR.	String (3)
ITEM_TEXT	The unique name of the PCCR, for example, "PCCR10."	String (50)
COMP_CODE	The unique name of the company code specified in the PCCR.	String (5)
DOC_DATE	The charge transaction date.	Date
DUE_DATE	The payment net due date.	Date
SUPPLIER_ID	The supplier ID loaded in the charge.	String (50)
SUPPLIER_NAME	The supplier name that corresponds to the supplier loaded in the charge, if a matching supplier is found. Otherwise, it is the supplier name loaded in the charge.	String (50)
PCARD_NUMBER	The masked purchasing card number specified in the PCCR.	Encrypted String
PCARD_HOLDER	The name of the purchasing card holder.	String (50)
PCARD_ORDER	The unique name of the order specified in the PCCR, for example, "PCO18."	String (50)
ADAPTER_SOURCE	The adapter source for the charge.	String (50)

Following is an example of `CRPaymentHeaderExport.csv`:

```
PAYMENTID,REF_DOC_NO,BANK_ACCT_KEY,PAYEEBANK,BANK_CD,GROSS_AMOUNT,CURRENCY,ITEM_TEXT
,COMP_CODE,
DOC_DATE,DUE_DATE,SUPPLIER_ID,SUPPLIER_NAME,PCARD_NUMBER,PCARD_HOLDER,PCARD_ORDER,AD
APTER_SOURCE

"PAY128","PCC128","01","BuyerBankLocation-1","13432","200.0000","USD","PCCR128","300
0",
"20080205",,"2","ManPower","XXXX-XXXX-XXXX-7969","skrishna",,"External"
"PAY129","PCC129","01","BuyerBankLocation-1","13432","450.0000","USD","PCCR129","300
0",
"20080205",,"2","0000005300","XXXX-XXXX-XXXX-7969","skrishna","PCO14","External"
"PAY131","PCC131","01","BuyerBankLocation-1","13432","350.0000","USD","PCCR131","300
0",
"20080210",,"2","0000005300","XXXX-XXXX-XXXX-7969","skrishna","PCO18","External"
```

```
"PAY130","PCC130","01","BuyerBankLocation-1","13432","200.0000","USD","PCCR130","300
0",
"20080210",,"2","ManPower","XXXX-XXXX-XXXX-7969","skrishna",,"External"
```

Charge Payment Detail Data for SAP ERP

The `CRPaymentDetailExport.csv` file provides the line-level details for charge payment.

The following table describes the columns in the `CRPaymentDetailExport.csv` file:

Column	Description	Format
`PAYMENTID`	The unique reference ID of the payment. This value matches a payment defined in `CRPaymentHeaderExport.csv`.	String (50)
`ITEM_AMOUNT`	The charge item amount.	Numeric
`WBS_ELEM`	The unique internal identifier of the SAP work breakdown structure (WBS) element.	String (50)
`ASSET_NO`	The unique internal identifier of the SAP asset number.	String (50)
`COSTCENTER`	The unique internal identifier of the SAP cost center.	String (50)
`INVOICE_DOC_ITEM`	The charge line item number.	Numeric
`ORDERID`	The internal order ID.	String (50)
`GL_ACCOUNT`	The unique internal identifier of the SAP general ledger account.	String (50)
`COMP_CODE`	The unique internal identifier of the SAP company code.	String (50)
`PLANT`	The unique internal identifier of the SAP plant, which is the ship-to address.	String (50)
`ITEM_TEXT`	The charge line item description.	String (2000)
`LINE_TYPE_LOOKUP_CODE`	The line category. Line categories separate procurement line items from the various special handling categories, such as tax and freight. The line categories are as follows: • `1` - Line Item (for all line items, including non-catalog, customer catalog and PunchOut) • `2` - Tax Charge (for all tax-related charges) • `4` - Freight Charge (for all freight-related charges) • `8` - Handling Charge (for all handling-related charges) • `16` - Special Charge (for other special charges, not related to tax, freight, or handling) • `32` - Discount (for discounts related to invoicing)	Numeric
`HDR_LEVEL`	A value that specifies whether the item is a header-level or line-level item as follows: • `Yes` - header level • `No` - line level	Boolean

Following is an example of `CRPaymentDetailExport.csv`:

```
PAYMENTID,ITEM_AMOUNT,WBS_ELEM,ASSET_NO,COSTCENTER,INVOICE_DOC_ITEM,ORDERID,GL_ACCOU
NT,COMP_CODE,PLANT,ITEM_TEXT,LINE_TYPE_LOOKUP_CODE,HDR_LEVEL

"PAY128","200.0000",,,"0000004400","1",,"0000445000","3000","3000","ManPower",,"No"
"PAY129","450.0000",,,"0000004400","1",,"0000400200","3000","3000","ManPower",,"No"
"PAY131","350.0000",,,"0000004400","1",,"0000400200","3000","3000","ManPower",,"No"
"PAY130","200.0000",,,"0000004400","1",,"0000445000","3000","3000","ManPower",,"No"
```

Charge Payment Request Data for Generic ERP

When you run the `Export Charge Payment Requests` and `Export Charge Payment Requests in 3 File Format` data export tasks from Ariba Administrator, it exports data to a ZIP file.

The `Export Charge Payment Requests` data export task exports a ZIP file containing the following CSV files:

- `CRPaymentHeaderExport.csv`
- `CRPaymentDetailExport.csv`
- `CRPaymentBankExport.csv`
- `CRPaymentSplitAccountingExport.csv`

The `Export Payment Requests in 3 File Format` data export task exports a ZIP file containing the following CSV files:

- `CRPaymentHeaderExport.csv`
- `CRPaymentBankExport.csv`
- `CRPaymentDetailWithSplitAccountingExport.csv`

Related Information

Charge Payment Header Data for Generic ERP [page 267]
Charge Payment Detail Data for Generic ERP [page 268]
Charge Payment Bank Export Data for Generic ERP [page 269]
Charge Payment Split Accounting Data for Generic ERP [page 270]
Charge Payment Detail with Split Accounting Data for Generic ERP [page 271]

Charge Payment Header Data for Generic ERP

The `CRPaymentHeaderExport.csv` file is exported by both `Export Charge Payment Requests` and `Export Charge Payment Requests in 3 File Format` tasks.

The following table describes the fields in the `CRPaymentHeaderExport.csv` file:

Field Name	Description	Format
`PurchasingUnit`	The unique internal identifier of the purchasing unit specified in the PCCR.	String (5)
ChargeNumber	The unique name of the charge.	String (50)
`PCardNumber`	The masked purchasing card number specified in the charge reconciliation.	Encrypted String
`PCardOrder`	The unique name of the order specified in the PCCR.	String (50)
`PaymentId`	The unique reference ID of the payment.	String (50)
`ChargeReconciliationId`	The unique name of the PCCR, for example, "PCCR63."	String (50)
`TransactionDate`	The charge transaction date.	Date
`SupplierId`	The supplier ID loaded in the charge	String (50)
`Supplier`	The supplier name that corresponds to the supplier loaded in the charge, if a matching supplier is found. Otherwise, it is the supplier name loaded in the charge.	String (50)
`NumLines`	The number of lines on the PCCR.	Numeric
`GrossAmount`	The gross amount of the payment.	Numeric
`GrossAmountCurrency`	The currency for the total cost specified in the PCCR.	String (3)
`CardHolder`	The name of the purchasing card holder.	String (50)
`AdapterSource`	The adapter source for the charge.	String (50)

Following is an example of `CRPaymentHeaderExport.csv`:

```
PurchasingUnit,ChargeNumber,PCardNumber,PCardOrder,PaymentId,ChargeReconciliationId,
TransactionDate,SupplierId,Supplier,NumLines,GrossAmount,GrossAmountCurrency,CardHol
der,
AdapterSource

"US002",,"XXXX-XXXX-XXXX-7969",,"PAY63","PCCR63","02/05/2008 00:00:00 -0800","2",
ManPower,,"200.0000","USD","skrishna","External"
"US002",,"XXXX-XXXX-XXXX-7969","PCO29","PAY62","PCCR62","02/05/2008 00:00:00
-0800","2",
ManPower,,"450.0000","USD","skrishna","External"
"US002",,"XXXX-XXXX-XXXX-7969","PCO32","PAY76","PCCR76","02/10/2008 00:00:00
-0800","2",
ManPower,,"350.0000","USD","skrishna","External"
"US002",,"XXXX-XXXX-XXXX-7969",,"PAY75","PCCR75","02/10/2008 00:00:00 -0800","2",
ManPower,,"200.0000","USD","skrishna","External"
"US002",,"XXXX-XXXX-XXXX-7969","PCO33","PAY77","PCCR77","02/10/2008 00:00:00
-0800","2",
ManPower,,"400.0000","USD","skrishna","External"
```

Charge Payment Detail Data for Generic ERP

The `CRPaymentDetailExport.csv` file is exported by the `Export Charge Payment Requests` task.

The following table describes the columns in the `CRPaymentDetailExport.csv:` file:

Column	Description	Format
`PaymentId`	The payment's unique reference ID. This value matches a payment defined in `CRPaymentHeaderExport.csv`.	String (50)
`ChargeReconciliationId`	The PCCR document ID.	String (50)
`PurchasingUnit`	The unique internal identifier of the purchasing unit specified in the PCCR.	String (50)
`LineNumber`	The charge line number in the voucher.	Numeric
`ParentLineNumber`	The voucher line number of the parent line for the charge line item.	Numeric
`LineTypeCategory`	The line category. Line categories separate procurement line items from the various special handling categories, such as tax and freight. The line categories are as follows: • `1` - Line Item (for all line items, including non-catalog, catalog and PunchOut) • `2` - Tax Charge (for all tax-related charges) • `4` - Freight Charge (for all freight-related charges) • `8` - Handling Charge (for all handling-related charges) • `16` - Special Charge (for other special charges, not related to tax, freight, or handling) • `32` - Discount (for discounts related to invoicing)	String
`Description`	The charge line item description.	String (2000)
`Amount`	The charge line item amount.	Numeric
`UnitPrice`	The charge line item unit price.	Numeric
`AmountCurrency`	The currency code for the charge line item amount.	String (50)
AccrualTaxAmount	The tax amount that has been accrued.	Numeric

Following is an example of `CRPaymentDetailExport.csv`:

```
PaymentId,ChargeReconciliationId,PurchasingUnit,LineNumber,ParentLineNumber,LineType
Category,
Description,UnitPrice,Amount,AmountCurrency,AccrualTaxAmount

"PAY63","PCCR63",,"1",,,"ManPower","200.0000","200.0000","USD",,
"PAY62","PCCR62",,"1",,,"ManPower","450.0000","450.0000","USD",,
"PAY76","PCCR76",,"1",,,"ManPower","350.0000","350.0000","USD",,
"PAY75","PCCR75",,"1",,,"ManPower","200.0000","200.0000","USD",,
"PAY77","PCCR77",,"1",,,"ManPower","400.0000","400.0000","USD",,
```

Charge Payment Bank Export Data for Generic ERP

The `CRPaymentBankExport.csv` file is exported by both `Export Charge Payment Requests` and `Export Charge Payment Requests in 3 File Format` tasks.

The following table describes the columns in the `CRPaymentBankExport.csv` file:

Column	Description	Format
`PaymentId`	The payment's unique reference ID. This value matches a payment defined in `CRPaymentHeaderExport.csv`.	String (50)
`ChargeReconciliationId`	The PCCR document ID.	String (50)
`PurchasingUnit`	The unique internal identifier of the purchasing unit.	String (50)
`PaymentMethod`	The unique internal identifier of the payment method. A payment method defines how a buying organization makes a payment to a supplier.	String (50)
`BuyerBankLocation`	The unique internal identifier of the bank payment location configured in the buying solution.	String (50)
`BankAccountType`	The unique internal identifier of the bank account type, such as `checking` or `savings`.	String (50)
`ScheduledPaymentDate`	The date of the scheduled payment.	Date
`BankId`	The bank identifier.	String (25)
`Amount`	The payment amount.	Numeric
`GrossAmount`	The payment gross amount.	Numeric
`DiscountAmount`	The discount amount.	Numeric
`AmountCurrency`	The transaction currency code for the payment.	String (50)

Following is an example of `CRPaymentBankExport.csv`:

```
PaymentId,ChargeReconciliationId,PurchasingUnit,PaymentMethod,BuyerBankLocation,Bank
AccountType,
ScheduledPaymentDate,BankId,Amount,GrossAmount,DiscountAmount,AmountCurrency

"PAY63","PCCR63","US002",,"BuyerBankLocation-1","checking","02/05/2008 21:33:20
-0800",
"TRBNK","200.0000","200.0000","0.0000","USD"
"PAY62","PCCR62","US002",,"BuyerBankLocation-1","checking","02/05/2008 21:50:48
-0800",
"TRBNK","450.0000","450.0000","0.0000","USD"
"PAY76","PCCR76","US002",,"BuyerBankLocation-1","checking","02/09/2008 19:50:39
-0800",
"TRBNK","350.0000","350.0000","0.0000","USD"
"PAY75","PCCR75","US002",,"BuyerBankLocation-1","checking","02/09/2008 19:50:39
-0800",
"TRBNK","200.0000","200.0000","0.0000","USD"
"PAY77","PCCR77","US002",,"BuyerBankLocation-1","checking","02/09/2008 20:01:08
-0800",
"TRBNK","400.0000","400.0000","0.0000","USD"
```

Charge Payment Split Accounting Data for Generic ERP

The `CRPaymentSplitAccountingExport.csv` file is exported by the `Export Charge Payment Requests` task.

The following table describes the columns in the `CRPaymentSplitAccountingExport.csv` file:

Column	Description	Format
`PaymentId`	The payment's unique reference ID. This value matches a payment defined in `CRPaymentHeaderExport.csv`.	String (50)
`ChargeReconciliationId`	The PCCR document ID.	String (50)
`PurchasingUnit`	The unique internal identifier of the purchasing unit specified in the PCCR.	String (50)
`LineNumber`	The charge line number of the voucher.	Numeric
`SplitLineNumber`	The split accounting charge line number.	Numeric
`Company`	The unique internal identifier of the company.	String (50)
`BusinessUnit`	The unique internal identifier of the business unit.	String (50)
`CostCenter`	The unique internal identifier of the cost center.	String (50)
`Account`	The unique internal identifier of an account, for example, `Expense`,	String (50)
`SubAccount`	The unique internal identifier of the subaccount.	String (50)
`Product`	The unique internal identifier of the product.	String (50)
`Region`	The unique internal identifier of the accounting region.	String (50)
`Quantity`	The charge line item's quantity.	Numeric
`Amount`	The charge line item's amount.	Numeric

Following is an example of `CRPaymentSplitAccountingExport.csv`:

```
PaymentId,ChargeReconciliationId,PurchasingUnit,LineNumber,SplitLineNumber,Company,B
usinessUnit,
CostCenter,Account,SubAccount,Product,Region,Quantity,Amount

"PAY63","PCCR63",,"1","1","02","US002","4000",,"5009",,,"1.0000","200.0000"
"PAY62","PCCR62",,"1","1","02","US002","4000","7752","5009",,,"1.0000","450.0000"
"PAY76","PCCR76",,"1","1","02","US002","4000","7752","5009",,,"1.0000","350.0000"
"PAY75","PCCR75",,"1","1","02","US002","4000",,"5009",,,"1.0000","200.0000"
"PAY77","PCCR77",,"1","1","02","US002","4000","7752","5009",,,"1.0000","400.0000"
```

Charge Payment Detail with Split Accounting Data for Generic ERP

`CRPaymentDetailWithSplitAccountingExport.csv` is exported by `Export Charge Payment Requests in 3 File Format` only.

The following table describes the columns in `CRPaymentDetailWithSplitAccountingExport.csv`:

Column	Description	Format
`PaymentId`	The payment's unique reference ID. This value matches a payment defined in `CRPaymentHeaderExport.csv`.	String (50)
`ChargeReconciliationId`	The PCCR document ID.	String (50)
`PurchasingUnit`	The unique internal identifier of the purchasing unit.	String (50)
`LineNumber`	The charge line number of the voucher.	Numeric
`ParentLineNumber`	The voucher line number of the parent line of the charge line item.	Numeric
`LineTypeCategory`	The line category. Line categories separate procurement line items from the various special handling categories, such as tax and freight. The line categories are as follows: • `1` - Line Item (for all line items, including non-catalog, customer catalog and PunchOut) • `2` - Tax Charge (for all tax-related charges) • `4` - Freight Charge (for all freight-related charges) • `8` - Handling Charge (for all handling-related charges) • `16` - Special Charge (for other special charges, not related to tax, freight, or handling) • `32` - Discount (for discounts related to invoicing)	String
`Description`	The charge line item description.	String (2000)
`UnitPrice`	The charge line item unit price.	Numeric
`AmountCurrency`	The currency code for the charge line item amount.	String (50)
`SplitLineNumber`	The split accounting charge line number.	Numeric
`Company`	The unique internal identifier of the company.	String (50)
`BusinessUnit`	The unique internal identifier of the business unit.	String (50)
`CostCenter`	The unique internal identifier for the cost center.	String (50)
`Account`	The unique internal identifier of the account, for example, `Expense`,	String (50)
`SubAccount`	The unique internal identifier of the subaccount.	String (50)
`Product`	The unique internal identifier of the product.	String (50)
`Region`	The unique internal identifier of the accounting region.	String (50)
`Quantity`	The charge line item quantity.	Numeric
`Amount`	The charge line item amount.	Numeric

Following is an example of `RPaymentDetailWithSplitAccountingExport.csv`:

```
PaymentId,ChargeReconciliationId,PurchasingUnit,LineNumber,ParentLineNumber,LineType
Category,
Description,UnitPrice,AmountCurrency,SplitLineNumber,Company,BusinessUnit,CostCenter
,Account,
SubAccount,Product,Region,Quantity,Amount

"PAY102","PCCR102",,"1",,,"SSP Test Supplier account","0.0000","USD","1","GD",
"US005","3300","7520","1210",,,"1.0000","0.0000"
```

Configuration of Receiving

The receiving process begins when items from a supplier arrive at their shipping destination. By default, the person responsible for the receiving of ordered items creates a manual receipt of the items by quantity and submits it for approval.

SAP Ariba Buying and Invoicing and SAP Ariba Buying (with receiving enabled) use a standard set of receiving types to determine whether a receipt is required for an item on a purchase order or contract. You can configure receiving type rules that match receiving types to commodity codes or supplier part numbers. For example, you can also configure the buying solution to create receipts automatically (called auto-receiving) based on amount.

You can use receiving type rules to configure:

- Auto-receiving based on supplier part number or commodity code
- Amount thresholds for auto-receiving, so that goods are automatically received if the is less than or equal to the configured threshold amount
- Custom conditions that determine how an item is received based on fields in the requisition
- Tolerance settings to control over- and under-receiving orders, at both the customer site in general, as well as at specific purchasing unit levels

There are also self-service configuration parameters for controlling various aspects of receiving.

> **Note**
>
> Contact SAP Ariba Support to enable receiving on SAP Ariba Buying or to disable it on SAP Ariba Buying and Invoicing.

Receiving Types [page 274]

Amount Thresholds for Auto-Receiving [page 276]

Asset Data Information on Receipts [page 279]

Receiving Tolerances [page 280]

Data Import Tasks for Receiving Types [page 283]

Receiving Type Values in CSV Files [page 283]

Import Receiving Types by Commodity Code [page 284]

Importing Receiving Types by Part Number [page 289]

Managing Receiving Types by Commodity Code [page 291]

Configuring Auto-Receiving Lead Times for Non-Catalog Items [page 297]

Managing Asset Receiving [page 297]

Related Information

Receiving Parameters

Receiving Types

There are 14 receiving types.

The following table describes the standard receiving types.

Receiving Type	Description
Auto-receipt on charge reconciliation - reconcile by amount **Auto-receipt on charge reconciliation - reconcile by quantity**	Automatically receives the order if originating from a PCard and reconciles by amount or quantity.
A**uto-receipt on due date - receive by amount** **Auto-receipt on due date - receive by quantity**	Automatically generate a receipt on the due date. The due date for a customer catalog item is calculated as the date on which the purchase order is created plus the standard lead time for the item specified in the customer catalog. The lead time of the non-catalog items is controlled by a parameter that has a default lead time of 0 (zero days). Because the receipts are created automatically, the received quantity or amount is the same as the purchase order quantity or amount and you cannot specify any asset information of the received goods.
Auto-receipt on invoice - reconcile by amount **Auto-receipt on invoice - reconcile by quantity**	Automatically generate a receipt when the invoice is generated. When an invoice is received in the system, an invoice reconciliation (IR) is created after the invoice is matched to a PO. You cannot specify any asset information of the received goods in this receiving type. **Note** If you manually reject the invoice or match the invoice to a different purchase order, the received quantities of the original PO are not removed automatically. If the PO line quantity is specified, the receipt is created for that quantity, and the received amount is calculated by multiplying the received quantity by the PO line unit price.
Auto-receipt on order date - receive by amount **Auto-receipt on order date - receive by quantity**	Automatically generate a receipt when the item is ordered (goes to the **Ordered** state).
Auto-receipt on invoice or charge reconciliation - reconcile by amount **Auto-receipt on invoice or charge reconciliation - reconcile by amount**	This receiving type lets customers configure a commodity code or supplier part number to be auto-received regardless of whether the order is invoiced or charged through a purchasing card. • For orders originating from a PCard, automatically receives the order and reconciles by amount or quantity. • For orders for which an invoice is generated, automatically generates a receipt when the invoice is created.

Receiving Type	Description
No receipt required - reconcile by amount	No receipt is generated, but the item is automatically considered received, though the order is in the **Ordered** state.
No receipt required - reconcile by quantity	Note Receiving exceptions are not generated for this receiving type.
Manual receipt - receive by amount	Receiving is done manually by amount or quantity.
Manual receipt - receive by quantity	

Considerations for receiving types include:

- Members of the **Receiving Administrator**, **Purchasing Agent**, and **Customer Administrator** groups can manage receiving types by using the Receiving Types By Commodity Code and Receiving Types by Part Number tasks in Ariba Administrator.
- The default receiving type is manual receiving by quantity.
- For manual receiving, add approvers to the approval flow for manually created receipts. If approvers are not added to the flow, the receipt is displayed as **Receipt withheld** and it will not be sent to SAP Business Network.
- Line items that are set for auto-receive can be manually received if:
  - all the lines are configured for auto-receiving
  - the order is partially received

  > Note
  >
  > When you manually receive a line item that was originally configured for auto-receiving, the buying solution converts the receiving type for that line to a manual receipt type - by amount or quantity (depending on the original receiving type).
- Auto-received receipts do not need to be approved by receipt approvers.
- Negative receiving can be performed on auto-received line items.
- Product returns on auto-received line items is not supported.
- Organizations generally use quantity-based receiving for goods items, and amount-based receiving for services items. When you use amount-based receiving types, all invoice accumulators and invoice exception calculations are based on the **Amount** field only.

  > Note
  >
  > The amount-based commodities or service line items do not need advanced pricing details, such as quantity, unit of measure, unit conversion, and pricing description.

Amount Thresholds for Auto-Receiving

Amount thresholds determine whether lines or orders are auto-received or manually received based on the amount of the line item or order. The buying solution automatically receives lines or orders that have an amount less than or equal to the threshold amount.

There are two ways to configure amount thresholds for auto-receiving:

- You can configure auto-receiving amount thresholds for order lines based on commodity code or supplier part number. This way, the amount threshold for one commodity code or part number can be different from the threshold for another commodity code or part number.
- SAP Ariba Customer Support can configure site-wide amount thresholds at the order level and the line level. These site-wide thresholds apply to all lines and orders, regardless of commodity code or part number.

Related Information

Amount Thresholds Based on Part Number or Commodity Code [page 276]
Amount Thresholds for All Orders and Line Items [page 277]
Workflow for Amount Thresholds [page 277]
Example:Auto-Receiving by Part Number [page 278]
Example: Receiving a Purchase Order Based on Commodity Code [page 278]
Example: Auto-Receiving All Non-Catalog Items [page 279]

Amount Thresholds Based on Part Number or Commodity Code

If the receiving type rules for part numbers or commodity codes include amount thresholds, when a line item matches a receiving type rule (including matching all conditions), the buying solution checks the threshold and assigns the appropriate receiving type.

If the line item amount qualifies for auto-receiving based on the threshold, the line item is assigned the auto-receiving type specified.

If the line item amount is equal to or greater than the threshold amount, the line item is assigned a manual receiving type. The mode of receiving (that is, by quantity or by amount) is the same as the mode of the specified auto-receiving type.

You can configure amount thresholds in the following ways:

- Import the amount thresholds using the data import tasks `Import Receiving Types By Commodity Code` and `Import Receiving Types By Part Number`.
- Use the **Procure-to-Pay Manager** workspace in Ariba Administrator.

Related Information

About Data Import Tasks for Receiving Types [page 283]
About Managing Receiving Types by Commodity Code [page 291]

Amount Thresholds for All Orders and Line Items

You can have SAP Ariba Customer Support configure amount thresholds that apply to all orders and line items, regardless of part number or commodity code.

When these thresholds are configured, the buying solution automatically receives by due date any order or line item that is less than or equal to the specified threshold amount.

The buying solution evaluates the total order amount first. If the entire order amount is over the order-level threshold, the line-level threshold is considered.

> **Note**
>
> If your site includes thresholds set for specific supplier part numbers or commodity codes, the buying solution checks those thresholds first. Then, for the line items that did not qualify for auto-receiving based on part number or commodity code threshold, the buying solution checks the amount thresholds that apply to all orders and line items.

SAP Ariba Customer Support uses the following site configuration options (parameters) for setting amount thresholds that apply to all orders and line items:

- `Application.Procurement.AutoReceiveOrderAmount` (for orders)
- `Application.Procurement.AutoReceiveLineItemAmount` (for line items)

Workflow for Amount Thresholds

The workflow for amount thresholds is a three step process.

The following process describes the workflow for determining receiving types using amount thresholds:

1. The receiving administrator configures receiving type rules that set line level thresholds for part numbers and commodity codes.
2. A user creates a requisition, the requisition is approved, and the order is created and sent to the supplier.
3. The buying solution goes through the following process to assign a final receiving type:
    1. The buying solution checks the line items in the order against the configured receiving type rules. When a line item matches a rule (including matching all conditions in the rule), the buying solution checks the line level amount threshold in the rule, if set. If the line item qualifies for auto-receiving, it is assigned the specified auto-receiving type. Otherwise, it is assigned a manual receiving type. The mode used (by quantity or by amount) is the mode specified in the receiving type.
    2. For any line items that are assigned a manual receiving type in the previous step, the buying solution checks the site-wide order level amount threshold, if set. If the order total is less than or equal to the

threshold amount, all the line items that were assigned a manual receiving type are assigned the receiving type **Auto-receive on order date**.

3. For any line items that are still assigned a manual receiving type after the order level threshold is checked, the buying solution checks the site-wide line level amount threshold. If a line item amount is less than or equal to the threshold amount, it is assigned the receiving type **Auto-receive on order date**.

Example:Auto-Receiving by Part Number

This example illustrates the flow for determining receiving types based on supplier part numbers and line item amounts.

1. The receiving administrator configures the following receiving type rules:
   - receiving type rule 1:
     Supplier = JCN, Part Number = *
     Receiving Type = SystemReceived (Auto-receipt on order date – receive by quantity)
     LineLevelThreshold = 20, Currency = USD
   - receiving type rule 2:
     Supplier = JCN, Part Number = AB*
     Receiving Type = SystemReceived (Auto-receipt on order date – receive by quantity)
     LineLevelThreshold = (blank), Currency = (blank)
2. An order containing the following line items is sent to a supplier:
   - Line item 1: Supplier = JCN, Part Number = AB1987, Amount = $100 USD
   - Line item 2: Supplier = JCN, Part Number = CD8979, Amount = $25 USD
3. The buying solution checks the receiving type rules to find the rule that matches the supplier part number of the line item.
   - Line item 1 matches the part number value in receiving type rule 1 (*, for all part numbers). However, the line item amount is $100, which is greater than the threshold amount of $20.
     Line item 1 matches the part number value in receiving type rule 2 (AB*). That rule does not have line-level threshold.
     Therefore, line item 1 is assigned the receiving type **Auto-receive on order date – receive by quantity**, based on rule 2.
   - Line item 2 matches the part number value in receiving type rule 1 (*, for all part numbers). However, the line item amount is $25, which is greater than the threshold amount of $20.
     Line item 2 does not match receiving type rule 2, because the part number CD8979 does not match the part number criteria of AB*.
     Therefore, line item 2 is assigned the receiving type **Manual receipt – receive by quantity** based on rule 1.

Example: Receiving a Purchase Order Based on Commodity Code

This example illustrates the flow for determining receiving types based on commodity code, line item amount, and custom conditions.

1. The receiving administrator has configured auto-receiving for line items with commodity code 8795 with an auto-receive threshold of $99.

2. In addition, the administrator has configured six custom conditions for the commodity code. All six conditions must be met for the line item to be auto-received.
3. A purchase order is sent to the supplier. The purchase order includes a line item the specifies commodity code 8795 and has an amount of $85.
4. The buying solution does the following:
    1. Checks the site-wide order level threshold ($75). In this case, the order level threshold is exceeded.
    2. Checks the line level threshold set in the CSV file. The line item amount ($85) is less than the threshold ($99).
    3. Checks all six custom conditions.
       If the line item meets all six custom conditions, it is marked for auto-receipt.
       If a line item meets more than one condition, the receiving type rule with the lowest ranking number is used. If no ranking is specified, the selection is random.
       If the line item does not meet all six custom conditions, the buying solution checks for a site-wide line level threshold. In this example, no site-wide line level threshold is set, so the line item is assigned a manual receiving type.

Example: Auto-Receiving All Non-Catalog Items

This example uses a commodity code receiving type rule to enable the buying solution auto-receive all line items that are non-catalog items.

Commodity Code = ALL Requisition.LineItem.IsAdHoc = Yes (this is a custom condition) Receiving Type = SystemReceived (Auto-receipt on order date – receive by quantity) No amount threshold specified

In this case, when the order is sent, the buying solution checks the receiving type rules and assigns all non-catalog line items the auto-receiving type, regardless of commodity code or amount.

Asset Data Information on Receipts

If your company tracks assets, you can configure the buying solution to include the asset data on receipts. If you choose to include asset data on receipts, the user supplies the asset data while filling out the receipt. The asset data comprises Serial Number, Tag Number, and Location.

To require asset data for receipts:

- Ask your SAP Ariba Customer Support representative to enable the asset data feature for your site. The asset data feature is disabled in the default configuration.
- Specify which receipts require asset data by setting:
    - The `IsAssetReq` column in the `ReceiptByCommodityCode.csv` or `ReceiptByPartNumber.csv` files, or
    - The **Asset Data Required** field on the Receiving Types By Commodity Code or Receiving Types by Part Number pages in Ariba Administrator.

Asset data is applicable to the manual receiving by quantity" (`ByCount`) receiving type only.

For a particular line item, the asset data setting associated with the part number takes precedence over the asset data setting associated with the commodity code.

Related Information

About Import Receiving Types By Commodity Code [page 284]
About Import Receiving Types By Part Number [page 289]

Receiving Tolerances

When configured, receiving tolerances allow you to define under-receiving and over-receiving limits on quantities and amounts on receipts and corresponding purchase orders.

Typically, the quantities and amounts on a receipt are expected to match the quantities and amounts specified in the corresponding purchase order. However, you may receive fewer quantities, or the same quantities for a lesser amount than what was specified on the purchase order (under-receiving). There may be instances where you may receive higher quantities, or the specified quantity at a higher amount than what was specified in the purchase order (over-receiving).

> **Note**
>
> You can define the separate limits for under and over-receiving.

To configure receiving tolerances, you must define the following attributes:

- Quantity: The difference between the number of items ordered and the number of items received. For example, if you set the under-receiving quantity to 10, and an order had a line item with a quantity of 50, the receipt for that line item must be for 40 items or more to be fully received. Likewise, if you set the over-receiving quantity to 10, the receipt can be for up to 60 items and still be valid.
- Percentage: The same as quantity, except that it is expressed as a percentage and applies to both quantity and total price (amount). Using the previous example, if you set the under-receiving percentage to 10, and the quantity ordered was 50, the receipt must be for at least 45 items (10% less than the quantity ordered) to be fully received. Likewise, if you set the over-receiving percentage to 10, the receipt can be for up to 55 items (10% more than the quantity ordered) and still be valid.
- Value: The difference between the ordered price (amount) and the received price. For example, if you set both the under-receiving value and over-receiving value to 5, and the total price for a line item in the order was $100, the total price of that line item in the receipt must be at least $95 to be fully received and no more than $105 to be valid.

  > **Note**
  >
  > The buying solution exports only the percentage tolerances (configured for purchase order receiving) to the external ERP systems. This is because most external ERP systems are equipped to handle only the percentage value of a PO line quantity or amount. Also, some ERP systems handle tolerance percentages ONLY when the over and under-receiving tolerance percentages are the same. That is, you must configure the same percentage value for both over and under- receiving tolerances.

You must configure the receiving tolerances in the buying solution taking your ERP system's capabilities into consideration (if the purchase orders are integrated with the externals ERPs).

Configuring the receiving tolerances without considering the ERP system's capabilities may result in having a discrepancy in the receiving status between the buying solution and the external ERP system.

Tolerance Operators

If you want to specify more than one of the tolerance attributes, you use the AND or OR operators to describe how they are combined. These operators are case-insensitive.

Use AND to require all three parameters to be true for the receipt to be fully received or valid. In this case, if any of the parameters are false, the receipt is either partial or invalid. If you use AND, be sure to specify all three attributes. If you do not specify a parameter, that tolerance will default to zero, which is usually undesirable when using AND.

Use OR to specify that any of the three attributes must be true for the receipt to be complete or valid. In this case, the receipt is partial or invalid only if all the parameters are false.

The default value for both the `UnderReceivingOperator` and `OverReceivingOperator` parameter is AND.

Configuration of Receiving Tolerance at the Purchasing Unit Level

Receiving tolerances configured for a customer site determines how the buying solution handles the receiving of all purchase orders that are generated for a company irrespective of the subsidiary or purchasing unit to which that PO is associated.

In a complex business scenario where a company has many subsidiaries that function as independent purchasing units, the receiving behavior for such subsidiaries may differ from that of the parent company due to various business requirements. To facilitate such differences at the purchasing unit level, you can now configure the receiving tolerances specific to each purchasing unit. (A purchasing unit is an SAP Ariba concept that lets you keep different sets of data distinct from one another.

Similar to a customer site-level configuration, under and over-receiving tolerances can be defined at the purchasing unit level based on absolute quantity, percentage of quantity or value, and Value (line item amount).

> ### Note
>
> If you choose to configure any one of these tolerance attributes (quantity, percentage, or value) at the purchasing unit level, you must ensure the other receiving tolerance attributes are also configured at that level.

Related Information

Purchasing Unit Administration Guide

Import Receiving Tolerances by PU Task

To configure the receiving tolerances at the purchasing unit level, you must import the required data in the form of a CSV file by executing the **Import Receiving Tolerance By PU** import task.

Following is an example of `ReceivingToleranceByPU.csv`

```
UTF8


ProcurementUnit,OverReceivingOperator,OverReceivingPercentage,OverReceivingQuantity,
OverReceivingValue,UnderReceivingOperator,UnderReceivingPercentage,UnderReceivingQua
ntity,UnderReceivingValue


US005,AND,10.00,1.00,5.00,AND,5.00,1.00,3.00
```

Under and Over-Receiving Attributes

You can specify separate tolerances for under-receiving and over-receiving.

For under-receiving, the tolerances determine when the receipt is considered fully received. If the tolerance is 0, then a receipt is considered a partial receipt until exactly the right number of items or total price (amount) has been received. However, if you specify an under-receiving tolerance, you are saying that the receipt can be considered fully received even if not all items have been received. For example, you could specify that the line items in the receipt can contain up to five items fewer than the quantity ordered and still be considered fully received. When receipts have been submitted within five items for each line item in the original order, the buying solution considers the order fully received.

For over-receiving, the tolerances determine whether the receipt is valid. If the received items exceed the ordered items by more than the over-receiving tolerances, the buying solution returns an error and does not send the receipt. For example, you could specify that the user can receive up to five items more than what was ordered without returning an error.

If the `Application.Procure.ValidateReceivingTolerancesForImportedReceipts` parameter is enabled, the receiving tolerance for receipts imported using the web services task is validated and an error message is displayed to indicate that the receipt can't be created. For example, if the tolerance for over-receiving is configured as 10%, the following error message is displayed for the line items that exceed the tolerance limit.

```
Receipt could not be created as it exceeds the receiving tolerance;

Order line number 1: Cannot receive more than 10 percent beyond amount ordered
```

> **Note**
>
> The validation message isn't displayed for contract-based receipts.

For information about the `Application.Procure.ValidateReceivingTolerancesForImportedReceipts` parameter, see Validate receiving tolerances for imported receipts.

Data Import Tasks for Receiving Types

Receiving types and amount thresholds can be associated with commodity codes or part numbers.

The following table describes the data import tasks and CSV files that associate receiving types with commodity codes and part numbers, and specifies whether the data is required.

Data Import Task	CSV Files Imported	Required
`Import Receiving Types By Commodity Code`	`ReceiptByCommodityCode.csv`	No
`Import Receiving Types By Part Number`	`ReceiptByPartNumber.csv`	No

You manually enter data in the CSV files. If you do not define a receiving type for a commodity code or part number, the default receiving type is manual receiving by quantity (`ByCount`).

For a particular line item, the receiving type associated with the part number takes precedence over the receiving type associated with the commodity code.

Related Information

About Import Receiving Types By Commodity Code [page 284]
About Import Receiving Types By Part Number [page 289]

Receiving Type Values in CSV Files

There are four CSV files that use or include `ReceivingType` information: `ReceiptByCommodityCodeExport.csv`, `ReceiptByCommodityCodeImport.csv`, `ReceiptByPartNumberExport.csv`, and `ReceiptByPartNumberImport.csv`.

The following table lists the supported values for `ReceivingType` in the `ReceiptByCommodityCodeExport.csv, ReceiptByCommodityCodeImport.csv, ReceiptByPartNumberExport.csv` and `ReceiptByPartNumberImport.csv` files:

ReceivingType	Description
`SystemReceived`	Auto-receipt on order date - receive by quantity
`SystemReceivedByAmount`	Auto-receipt on order date - receive by amount
`AutoReceipt`	Auto-receipt on due date - receive by quantity
`AutoReceiptByAmount`	Auto-receipt on due date - receive by amount
`ByCount`	Manual receipt - receive by quantity

ReceivingType	Description
`ByAmount`	Manual receipt - receive by amount
`NoReceipt`	No receipt required - reconcile by quantity
`NoReceiptByAmount`	No receipt required - reconcile by amount
`PCardAutoReceipt`	Auto-receipt on charge reconciliation - reconcile by quantity
`PCardAutoReceiptByAmount`	Auto-receipt on charge reconciliation - reconcile by amount
`InvoiceAutoReceipt`	Auto-receipt on invoice - reconcile by quantity
`InvoiceAutoReceiptByAmount`	Auto-receipt on invoice - reconcile by amount
InvoiceOrPCardAutoReceipt	Auto-receipt on invoice or charge reconciliation – reconcile by quantity
InvoiceOrPCardAutoReceiptByAmount	Auto-receipt on invoice or charge reconciliation – reconcile by amount

Import Receiving Types by Commodity Code

The `Import Receiving Types By Commodity Code` data import task lets you associate receiving types with commodity codes and amount thresholds. specify which commodity codes require asset data information, and configure custom conditions for determining receiving type based on fields on the requisition.

`Import Receiving Types By Commodity Code` reads data from the CSV file `ReceiptByCommodityCode.csv`.

Following is an example of `ReceiptByCommodityCode.csv` that sets amount thresholds for each commodity code.

> ⓘ Note
>
> SystemID values are automatically assigned when you first import or configure the receiving type rules. Leave this field blank when you import the data for the first time.

```
Cp1252


SystemID,Ranking,DefinitionLine,Domain,CommodityCode,ReceivingType,Description,IsAss
etReq,LineLevelThreshold,LineLevelThresholdCurrency

         RBCC20,,,unspsc,44112002,ByCount,Calendars,true,35,USD
         RBCC21,,,unspsc,4411,SystemReceived,Desk Accessories,false,50,USD
         RBCC22,,,unspsc,44,AutoReceiptOnDueDate,Cleaning Services,false,200,USD
         RBCC23,,,unspsc,44101801,AutoReceipt,Calculators or
accessories,false,100,USD
```

How Commodity Code Hierarchy Works

UNSPSC commodity codes are 8-digit numbers structured into the following hierarchical levels:

- Segment (first 2 digits)
- Family (first 4 digits)
- Class (first 6 digits)
- Commodity (all 8 digits)

You can specify commodity code hierarchies in the `ReceiptByCommodityCode.csv` or using the **Receiving Types by Commodity Code** task in the **Procure-to-Pay Manager** workspace. This determines the receiving type for a line item.

In the buying solution, if you have configured commodity codes to be associated with the following receiving types:

- 44112002 - By Count
- 4411 - System Received
- 44111801 - Auto-Received
- 44 - Auto-Receipt On Due Date

When an order is ready to be received, the buying solution checks each line item's commodity code in the purchase order, against the receiving type that has been configured and determines the receiving behavior for that line item.

A more detailed (lower level or longer) commodity code takes precedence over a less detailed (higher level or shorter) code. If both an 8-digit and a 4-digit commodity code are configured, and a line item matches both, the 8-digit code takes precedence as it is the more precise match. The best-matching commodity code determines the receiving type.

For example, consider an order that has the following line items with these commodity codes:

- Line item 1: 44112002
- Line item 2: 4411
- Line item 3: 44111801
- Line item 4: 44111802
- Line item 5: 44221001

The buying solution treats the line items as follows:

- Line item 1: Receiving type `By Count` (exact match)
- Line item 2: Receiving type `System Received` (exact match)
- Line item 3: Receiving type `Auto-Received` (exact match)
- Line item 4: Receiving type `System Received` (Association for 4411 is the lowest level available match)
- Line item 5: Receiving type `Auto-Receipt on Due Date` (Association for 44 is the lowest level available match)

Configuration of Receiving Types Based on Fields in the Requisition

If you configure receiving type rules for commodity codes, you can define up to 10 custom conditions that a line item must meet in order to be assigned the receiving type specified. Custom conditions consist of two parts: a field (any field available from a requisition) and a value.

For example, a receiving type rule can be specific to a specific purchasing unit or cost center.

Following is an overview of the process for configuring receiving conditions.

1. Run the following data export task: **Export Receiving Types By Commodity Code**.
2. Use the resulting export file, `ReceiptByCommodityCode.csv`, to define which requisition fields to use in the conditions. For example, if you want to configure one condition that looks at the purchasing unit of the requisition and another that evaluates whether the line item is a non-catalog item.
3. For each record to which you want the conditions applied, add the values you want to match for the conditions.
4. To import the `ReceiptByCommodityCode.csv` file, run the following data import task: **Import Receiving Types By Commodity Code**.

Related Information

Conditions in the ReceiptByCommodityCode.csv File [page 286]
Common Fields for Configuring Conditions [page 287]
Example: Custom Conditions in ReceiptByCommodityCode.csv [page 288]
About Defining Conditions in the ReceiptByCommodityCode.csv File [page 286]

Conditions in the ReceiptByCommodityCode.csv File

When you define a condition, you specify the requisition field to use and the value to match. To specify which requisition fields to use, insert a definition row to the `ReceiptByCommodityCode.csv` file. Make the definition row the first row below the header row.

The following example has a definition row with three requisition fields specified: purchasing unit, payment terms, and non-catalog items.

```
SystemID,Ranking,DefinitionLine,Domain,CommodityCode,ReceivingType,Description,IsAss
etReq,LineLevelThreshold,LineLevelThresholdCurrency,ConditionValue1,ConditionValue2,
ConditionValue3,ConditionValue4,ConditionValue5,ConditionValue6,ConditionValue7,Cond
itionValue8,ConditionValue9,ConditionValue10

,,Yes,,,,”This is a definition row, see administrator before
editing”,,,,Requisition.ProcurementUnit.UniqueName,Requisition.LineItems.IsAdHoc,Req
uisition.LineItems.PaymentTerms,null,null,null,null,null,null,null
```

Follow these guidelines when adding a definition row:

- Leave `SystemID` blank. The buying solution automatically generates this ID when you import the file. (This is true for any new row you add, not just a definition row.)
- In the `DefinitionLine` column, specify Yes.
- For `Description`, you don't need to include anything. However, you might want to include a note emphasizing that this is a definition row to help prevent inadvertent editing of the row.
- Do not provide values in the following columns: `Ranking`, `Domain`, `CommodityCode`, `ReceivingType`, `IsAssetReq`, `LineLevelThreshold`, `LineLevelThresholdCurrency`.
- In the `ConditionValueX` columns, specify the requisition fields you want to use as conditions. You must specify these fields by providing a requisition-based path to the field, using dot notation. For example, to create a condition based on payment terms, enter `Requisition.LineItems.PaymentTerms`.
- For any `ConditionValueX` columns that you do not use, enter leave them blank. (When the file is exported, these values will be "Undefined.")

After you define the fields to use in your conditions, specify the condition values used to match commodity codes with receiving types.

Notes about condition values:

- In order for a line item to match a receiving type rule defined in `ReceiptByCommodityCode.csv`, the line item must match all of the conditions specified in the row for that commodity code.
- Using the example in the preceding section, if a data row specifies condition values for `Requisition.LineItems.IsAdHoc` and `Requisition.LineItems.PaymentTerms`, line items must match both of those values in order to be assigned the specified receiving type. If the row does not specify a value for the other condition, `Requisition.ProcurementUnit.UniqueName`, the value in the purchasing unit field of the line item does not matter.
- You can configure different receiving type rules for a single commodity code as long as you specify different values for the conditions.

Related Information

About Common Fields for Configuring Conditions [page 287]

Common Fields for Configuring Conditions

The field path for a custom condition must point to a field that is one of the following simple types: String, Boolean, or Number.

The following table lists some requisition fields that are commonly used for receiving conditions:

Field Path	Description
`Requisition.LineItems.ShipTo.UniqueName`	The unique name of the address where ordered items will be shipped. This is a String field.

Field Path	Description
`Requisition.LineItems.Supplier.UniqueName`	The unique name of the company that supplies this product. This is a String field.
`Requisition.LineItems.HasAdhocShipToAddress`	A Boolean value that indicates whether this item has a non-standard (ad-hoc) shipping address.
`Requisition.LineItems.IsAdHoc`	A Boolean value that indicates whether a line contains an item that is neither a customer catalog nor a punchout item.
`Requisition.LineItems.PaymentTerms.UniqueName`	The unique name of the payment terms for this supplier location. This is a String field.
Requisition.ProcurementUnits.UniqueName	The unique name of the purchasing unit of the requisition. This is a String field.
`Requisition.LineItems.Accountings.SplitAccountings.CostCenter.UniqueName`	The unique name of the cost center of the split accounting line. **Note** If one line item has multiple split accountings, all of them must match the condition. For example, if the condition field path is `Requisition.LineItems.Accountings.SplitAccountings.CostCenter.UniqueName`, and the condition field value is CC750, when a line item has two split accountings, both must have Cost Center set to CC750 in order to match to the rule.
`Requisition.LineItems.OriginatingSystem`	This field identifies the external system from which this requisition was imported. The value is a string.

Example: Custom Conditions in ReceiptByCommodityCode.csv

You can set receiving conditions in the `ReceiptByCommodityCode.csv` file.

The following example illustrates how to set a receiving condition based on the purchasing unit for a requisition in the `ReceiptByCommodityCode.csv` file:

```
Cp1252


SystemID,Ranking,DefinitionLine,Domain,CommodityCode,ReceivingType,Description,IsAss
etReq,LineLevelThreshold,LineLevelThresholdCurrency,ConditionValue1,ConditionValue2,
ConditionValue3,ConditionValue4,ConditionValue5,ConditionValue6,ConditionValue7,Cond
itionValue8,ConditionValue9,ConditionValue10

RBCC11,0,
     Yes
    ,,,,”This is a definition row, see administrator before editing”,
,,,
     Requisition.ProcurementUnit.UniqueName
    ,null,null,null,null,null,null,null,null,null
RBCC12,0,,unspsc,90121601,SystemReceived,Office Supplies -
SysReceived,false,100,USD,
     US002
    ,null,null,null,null,null,null,null,null,null
```

```
RBCC13,2,,unspsc,90121601,AutoReceipt,Office Supplies - AutoReceipt,false,300,USD,
     US005
    ,null,null,null,null,null,null,null,null,null
RBCC14,0,,unspsc,901216,InvoiceOrPCardAutoReceipt,Paper Products -
InvPcard,false,500,USD,
     US002
    ,null,null,null,null,null,null,null,null,null
RBCC15,1,,unspsc,901216,InvoiceOrPCardAutoReceiptByAmount,Paper Products -
InvPcardAmount,false,600,USD,
     US100
    ,null,null,null,null,null,null,null,null,null
RBCC16,0,,unspsc,90121602,ByAmount,Computer Hardware - ByAmount,false,,USD,
     US002
    ,null,null,null,null,null,null,null,null,null
RBCC17,1,,unspsc,90121602,ByCount,Computer Hardware - ByCount,false,,USD,
     US005
    ,null,null,null,null,null,null,null,null,null
RBCC18,2,,unspsc,90121602,NoReceipt,Computer Hardware - NoReceipt,false,,USD,
     US100
    ,null,null,null,null,null,null,null,null,null
```

> **Note**
>
> - `SystemID` values are automatically assigned when you first import or configure the receiving type rules. Leave this field blank when you import the data for the first time.
> - The first row of data in the example has a value of Yes in the `DefinitionLine` column, indicating that the row is defining specific fields to be used in receiving conditions. In the example, the only field defined is `Requisition.ProcurementUnit.UniqueName`, the unique name of the purchasing unit on the requisition.
> - The remaining data rows specify the values that satisfy the receiving condition. For example, the second data row says: "For requisitions from purchasing unit US002, auto-receive on order any line items with commodity code 90121601."
> - The example uses `Ranking` values to prioritize receiving type rules for commodity codes that have multiple receiving type rules defined (such as commodity code 90121602).

Importing Receiving Types by Part Number

The `Import Receiving Types By Part Number` data import task associates receiving types and amount thresholds with part numbers. It also specifies which part numbers require asset data information. `Import Receiving Types By Part Number` reads data from the CSV file named `ReceiptByPartNumber.csv`.

Related Information

Using Wildcards for Part Numbers [page 290]
ReceiptByPartNumber.csv: PeopleSoft Example [page 290]
ReceiptByPartNumber.csv: SAP Example [page 290]
ReceiptByPartNumber.csv: Generic Example [page 291]

Using Wildcards for Part Numbers

You can specify a series or group of supplier part numbers by using a wildcard (*) in the `PartNumber` field in the `ReceiptByPartNumber.csv` file.

To use a wildcard for part numbers, follow these guidelines:

- The wildcard must be placed at the end of a string (`abc12*`). It can also be the only character in the field (which indicates all part numbers).
- The wildcard cannot be the first character in a string, nor can it be between characters in this field (for example, `*abc12` or `ab*12`).

ReceiptByPartNumber.csv: PeopleSoft Example

The `ReceiptByParNumber.csv` example illustrates three records, the first of which employs a wildcard in the `PartNumber` column.

```
UTF8


SupplierID,PartNumber,IsAssetReq,ReceivingType,Description,LineLevelThreshold,LineLe
velThreshold
Currency

sid480-be,RAY450*,true,ByCount,RAY45050 - By Count,50,USD
sid480-hq,MON152 25,false,InvoiceAutoReceipt,MON152 25-Invoice Auto Receipt,,
sid481-si,AD981,false,InvoiceAutoReceipt,AD981-InvoiceAutoReceipt,100,USD
```

ReceiptByPartNumber.csv: SAP Example

The `ReceiptByParNumber.csv` example illustrates three records, the first of which employs a wildcard in the `PartNumber` column.

```
UTF8


SupplierID,PartNumber,IsAssetReq,ReceivingType,Description,LineLevelThreshold,LineLe
velThreshold
Currency

0000000001,RAY450*,true,ByCount,RAY45050 - By Count,50,USD
0000000100,MON152 25,false,InvoiceAutoReceipt,MON152 25 - InvoiceAutoReceipt,,
0000000100,AD981,false,InvoiceAutoReceipt,AD981 - InvoiceAutoReceipt,100,USD
```

ReceiptByPartNumber.csv: Generic Example

The `ReceiptByParNumber.csv` example illustrates three records, the first of which employs a wildcard in the `PartNumber` column.

```
UTF8


SupplierID,PartNumber,IsAssetReq,ReceivingType,Description,LineLevelThreshold,LineLe
velThreshold
Currency

         29,RAY450*,true,ByCount,RAY45050 - By Count,50,USD
         11,MON152 25,false,InvoiceAutoReceipt,MON152 25-Invoice Auto Receipt,,
         11,AD981,false,InvoiceAutoReceipt,AD981-InvoiceAutoReceipt,100,USD
```

Managing Receiving Types by Commodity Code

You can view, add, modify, and delete receiving type rules in the **Procure-to-Pay Manager** workspace in Ariba Administrator. You can use the `Receiving Types By Commodity Code` task to manage receiving type rules associated with system commodity codes.

Related Information

How to View Receiving Type Rules for Commodity Codes [page 291]
How to Add or Modify Receiving Type Rules for Commodity Codes [page 292]
How to Delete Receiving Type Rules for Commodity Codes [page 293]
Managing Receiving Type Rules by Part Number [page 294]

How to View Receiving Type Rules for Commodity Codes

You can view your current receiving type rules for commodity codes in Ariba Administrator.

Procedure

1. From the dashboard, click **Manage** > **Core Administration**.
2. Click **Procure-to-Pay Manager**.
3. Click **Receiving Types By Commodity Code**, click **List All** to display all commodity codes, or enter search criteria and click **Search**.

The following table describes the columns on the **Receiving Types By Commodity Codes** page.

Column	Description
Domain	The domain for the system commodity code.
Commodity Code	The system commodity code.
Description	The user-visible description of the system commodity code.
Asset Data Required	Indicates whether asset data is required on receipts for items with this commodity code.
Receiving Type	The receiving type associated with the system commodity code.
Threshold Amount	The amount threshold for auto-receiving. Line items that have an amount less than or equal to the threshold amount (and match all of the conditions for the receiving type rule) are auto-received.

4. To view the any additional conditions that might be defined for the receiving type rule, click the description, then click the **Conditions** tab.

> ### Note
>
> To work with conditions, you must use the **Import Receiving Types by Commodity Code** data import task. You cannot use the **Procure-to-Pay Manager** workspace to add, modify, or delete custom conditions.

How to Add or Modify Receiving Type Rules for Commodity Codes

You can add new receiving type rules for commodity codes or modify existing rules.

Procedure

1. On the dashboard, click **Manage** > **Core Administration**.
2. Click **Procure-to-Pay Manager**.
3. Click **Receiving Types By Commodity Code** and click **Create New** to add a receiving type rule for a commodity code, or select the system commodity code you want to modify and click **Actions** > **Edit**.
4. Enter or modify information on the **Create Receiving Type** or **Edit Receiving Type** page. The following table describes the fields on these pages.

Field	Description
System ID	A unique internal ID for the receiving type rule. This read-only ID is generated automatically when the receiving type rule is created or imported. The field corresponds to the new SystemID column in the `ReceiptByCommodityCode.csv` file.

Field	Description
Adapter Source	The adapter source, which can be one of the following values: • **AribaManaged**: The object is not maintained in an external system. Objects with this data source are maintained in Ariba Administrator. • **External**: The object and all its data fields are maintained in an external system, such as an ERP system or a set of CSV files. Objects with this adapter source can be overwritten or deleted when data is imported from the external system. • **ManuallyMaintained**: The source of the object is an external system, but the object cannot be overwritten or deleted when data is imported from the external system. You cannot modify this field in edit mode.
Commodity Code	The system commodity code. You cannot modify this field in edit mode.
Ranking	This field is available only when there are multiple receiving type rules configured for the commodity code. Determines how to rank this receiving type rule when there are multiple receiving type rules for the same commodity code. If a purchase order line matches two or more rules for the same commodity code, the buying solution selects the rule with the lowest Ranking value greater than 0.
Description	The user-visible description for the system commodity code.
Asset Data Required	Select this check box if you want to require asset data on receipts for this commodity code. Asset data is applicable to the receiving type **Manual-receive by quantity** only. For a particular line item, the asset data setting associated with the part number takes precedence over the asset data setting associated with the commodity code.
Receiving Type	The receiving type you want to associate with the system commodity code.
Threshold Amount	Line-level auto-receiving threshold. If the monetary amount specified in this field is surpassed, the specified auto-receiving type is not used. The field appears if an auto-receiving option is selected.

5. Click **Save** to save your changes, or **Cancel** to return to the previous page without saving your changes.

How to Delete Receiving Type Rules for Commodity Codes

You can delete your current receiving type rules for commodity codes. Deleting a rule only deactivates it. It is still available for reactivation if needed.

Prerequisites

Before deleting the receiving type rule for a commodity code, we recommend that you export receiving type rule data to a CSV file by running the **Export Receiving Types By Commodity Code** data export task.

Procedure

1. On the dashboard, click **Manage** > **Core Administration**.
2. Click **Procure-to-Pay Manager** and click **Receiving Types By Commodity Code**.
3. Search for the receiving type rule you want to delete.
4. In search results, select the check box for the receiving type rule, and click **Delete**.
5. Click **OK** to delete the receiving type rule or **Cancel** to return to the previous page without deleting the receiving type rule.

   If a receiving type is accidentally or intentionally deleted (deactivated), you can still activate it by selecting the receiving type and clicking **Actions** > **Activate**.

Managing Receiving Type Rules by Part Number

You use the `Receiving Types By Part Number` task in the **Procure-to-Pay Manager** workspace to manage receiving types and amount thresholds associated with part numbers.

Related Information

How to View Receiving Type Rules for Part Numbers [page 294]
How to Add or Modify Receiving Type Rules for Part Numbers [page 295]
How to Delete Receiving Type Rules for Part Numbers [page 296]

How to View Receiving Type Rules for Part Numbers

You can view your current receiving type rules for part numbers.

Procedure

1. On the dashboard, click **Manage** > **Core Administration**.
2. Click **Procure-to-Pay Manager**.
3. Click **Receiving Types By Part Number** and click **List All** to display all part numbers, or enter search criteria and then click **Search**.

   The following table describes the columns on the **Receiving Types By Part Number** page.

Column	Description
Supplier ID	The system ID that uniquely describes the supplier organization.
Part Number	The part number of the item.
Description	The user-visible description of the item.
Asset Data Required	Indicates whether asset data is required on receipts for items with this part number.
Receiving Type	The receiving type associated with the part number.
Threshold Amount	Line-level auto-receiving threshold. If the monetary amount specified in this field is surpassed, the specified auto-receiving type is not used. The field appears if an auto-receiving option is selected.

How to Add or Modify Receiving Type Rules for Part Numbers

You can add new receiving type rules for part numbers or modify existing ones in Ariba Administrator.

Procedure

1. On the dashboard, click **Manage** > **Core Administration**.
2. Click **Procure-to-Pay Manager**.
3. Click **Receiving Types By Part Number** and click **Create New** to add a receiving type rule for a commodity code, or select the part number you want to modify and click **Actions** > **Edit**.
4. Enter or modify information on the **Create Receiving Type** or **Edit Receiving Type** page. The following table describes the fields on these pages.

Field	Description
Adapter Source	The adapter source, which can be one of the following values: • **AribaManaged**—The object is not maintained in an external system. Objects with this data source are maintained in Ariba Administrator. • **External**—The object and all its data fields are maintained in an external system, such as an ERP system or a set of CSV files. Objects with this adapter source can be overwritten or deleted when data is imported from the external system. • **ManuallyMaintained**—The source of the object is an external system, but the object cannot be overwritten or deleted when data is imported from the external system. You cannot modify this field in edit mode.
Supplier	The system ID of the supplier organizations that supplies the item. You cannot modify this field in edit mode.

Field	Description
Part Number	The part number of the item. You cannot modify this field in edit mode. When adding a receiving type rule, you can use a wildcard (*) in the **Part Number** field to indicate a group or series of part numbers. The wildcard must be entered at the end of a string (`abc12*`). It can also be the only character in the field (which indicates all part numbers). The wildcard cannot be the first character in a string, nor can it be between characters in the field. (For example, the following entries are invalid: *abc12, ab*12).
Description	The user-visible description of the item.
Asset Data Required	Select this check box if you want to require asset data on receipts for this part number. Asset data is applicable to the receiving type **Manual-receive by quantity** only. For a particular line item, the asset data setting associated with the part number takes precedence over the asset data setting associated with the commodity code.
Receiving Type	The receiving type you want to associate with the part number.

5. Click **Save** to save your changes, or **Cancel** to return to the previous page without saving your changes.

How to Delete Receiving Type Rules for Part Numbers

You can delete your current receiving type rules for part numbers. Deleting a rule deactivates the rule. It is still available for reactivation.

Prerequisites

Before deleting the receiving type rule for a part number, we recommend you export receiving type rule data to a CSV file by running the **Export Receiving Types By Part Number** data export task.

Procedure

1. On the dashboard, click **Manage** > **Core Administration**.
2. Click **Procure-to-Pay Manager** and click **Receiving Types By Part Number**.
3. Search for the part number you want to delete.
4. In search results, select the check box for the part number, and click **Delete**.

   Deletion only deactivates the part number. It does not delete it from the system.
5. Click **OK** to delete the receiving type, or **Cancel** to return to the previous page without deleting the receiving type rule.

   If a receiving type rule is accidentally or intentionally deleted (deactivated), you can still activate it by selecting the receiving type rule and clicking **Actions** > **Activate**.

Configuring Auto-Receiving Lead Times for Non-Catalog Items

Auto-receive rules allow the automatic creation of a receipt for any order or line item amount when an item is ordered, an invoice is received, or when the receipt due date is reached.

Your site can be configured to allow automatic receiving on the due date of the receipt for orders or line item amounts below a specified threshold value. This is useful if you do not want users to spend data entry time for low value receipts, or do not want to risk holding up payment of small invoice amounts waiting for a manual receipt.

Non-catalog items, typically, do not have supplier-furnished attributes because of which, the auto-receiving default lead time parameter value is set to zero (0). This generates the auto-receipt or email notifications immediately (when the order is submitted).

You can to choose to configure this parameter value to specify any appropriate lead time (in days). Contact your SAP Ariba Customer Support to configure the receiving lead times for non-catalog items.

Managing Asset Receiving

Asset receiving provides an option to add asset data to the receipt line items that have been identified as assets. A receipt line item is identified as an asset based on the Part Number or Commodity Code.

If Asset Data is enabled on your site, it is mandatory to add asset details to the corresponding asset line items in the receipt before it is approved. Contact SAP Ariba Customer Support to enable this feature on your site.

> **Note**
>
> In a multi-ERP environment, if you have enabled asset data on a parent site, and the associated child sites are configured with the same receiving type as the parent, then asset data becomes mandatory while receiving on all the child sites, by default. Also, asset data on such child sites is enabled automatically.

Any approver in the receipt approval workflow can add this asset data. Typically, the Central Receiver adds asset data at the time of receiving, but this data may not be accurate every time. To ensure accurate information is provided, you can add Asset Receiver Group users to the approval flow. The Asset Receiver can then review the asset information that is specified, change it if required, and approve the receipt.

If your site has been configured to receive orders with asset data, then either the requestor or any one of the approvers must ensure asset data is specified before the receipts are fully approved.

In the default configuration, when asset data is enabled for receiving, an Asset Receiver or the Customer Administrator is automatically added at the end of the receipt approval flow. This is to ensure asset data is specified before the receipt is fully approved.

If the Asset Receiver tries to approve the receipt without adding this asset data, an error message displays asking you to specify this mandatory information before approving.

Related Information

Purchasing Guide for Procurement Professionals
SAP Ariba Procurement Solutions Group Descriptions Table

Imported Receipts

The receipt import feature in SAP Ariba Buying and Invoicing and SAP Ariba Buying (when Receiving is enabled), imports and creates receipt data for requisitions (purchase orders and contract documents) that are created using a third-party ERP system.

You can import receipts into the buying solution by using the Import Receipts integration event in the user interface.

When importing receipts using the user interface in the buying solution, the import is performed in two stages and involves the following tasks:

1. Receipt data is imported from a third-party ERP system and stored as a staged object in the staging table within SAP Ariba Buying and Invoicing. To do this, run the Import Receipt integration event.
2. The receipt is created in SAP Ariba Buying and Invoicing. The `Process Imported Receipts` scheduled task creates the actual receipt from the staged object and links it to the corresponding purchase order in SAP Ariba Buying and Invoicing.
   The scheduled task matches the records in the staging table based on the following parameters (accessible by SAP Ariba Customer Support):
   - Application.Receiving.ReceiptImportReceivableKey: This is a string parameter required for the purchase order or contract match. This parameter value should be `"<Object>,<Object Field path>,<StagedObject Field Path>"`
     where `<Object>` is the object to be queried, `< Object Field path>` is the field path of the `<Object>` to be matched, and `<StagedObject Field Path>` is the field path in the staged object to be matched against the field path of `<Object>.`
     For example, `ReceiptImportReceivableKey = "Contract,UniqueName,ReceivableID";`
     In this example, when the `ReceiptImportReceivableKey` parameter has “Contract” as object value, it means that you can perform a receipt import of Contract documents only.
     To enable receipt import of only purchase orders, specify the following:
     `ReceiptImportReceivableKey = "PurchaseOrder,UniqueName,ReceivableID";`

     > **Note**
     >
     > To enable receipt import of both contracts and purchase orders, SAP Ariba Customer Support configures the parameter as:
     >
     > `ReceiptImportReceivableKey = "ReceivableLineItemCollection,UniqueName,ReceivableID";`

     > **Note**
     >
     > Buyers can choose between the SAP Ariba purchase order ID and the ERP purchase order ID as the field to be matched. Based on the buyer's choice, SAP Ariba Support defines the string parameter.

   - Application.Receiving.ReceiptImportLineItemKeys: This is a vector parameter required for the line item match. The value for each element of this vector parameter should be `"<Line Item Field Path>,<Staged Object Field Path>"`

where "<Line Item Field Path> is the field path of the line item to be matched, and <Staged Object Field Path> is the field path of the staged object to be matched against value of the <Line Item Field Path>.
For example, `ReceiptImportLineItemKeys = ("Supplier.Name,Supplier");`

ReceiptImport.csv File for Receipt Import [page 300]

Receipts Import [page 302]

Export Receipt Status Task [page 306]

Asynchronous Import of Receipts [page 308]

ReceiptImport.csv File for Receipt Import

Before you can import transaction data from your ERP system, you must create the `ReceiptImport.csv` file that contains the details of the receipt information you want to import.

When you run the `Export Receipt Status` integration event, the receipt status is exported in a ZIP format (`StagedReceiptExport.zip`) that contains the `StagedReceiptExport.csv` file.

> **Note**
>
> When the **Restrict receiving against deleted contract line items** (`Application.Receiving.RestrictReceivingAgainstDeletedContractLineItems`) parameter is enabled and if the file channel or web services task is run for receiving, the quantity or amount accepted will be '0' for deleted contract line items in the `ReceiptDetailExport.csv` file. These values will be updated based on the imported receipt for active line items. For more information, see Restrict receiving against deleted contract line items.

ReceiptImport.csv: Generic ERP Example

There are three examples of the `ReceiptImport.csv` file, one each for quantity-based receipt import, one for amount-based receipt import, and one illustrating partial receipt import.

Related Information

Quantity-Based Receipt Import [page 301]
Amount-Based Receipt Import [page 301]
Partial Receipt Import [page 301]

Quantity-Based Receipt Import

To perform quantity-based receipt import, set the `Receiving_By_Count` column in `ReceiptImport.csv` to `True` .

For example:

```
     UTF8


Receivable_Id,LineItem_Num,Supplier,Supplier_Location,Commodity_Code,Quantity,Unit_P
rice,Man_Part_Num,Item_Part_Num,Number_Accepted,Number_Rejected,Amount_Accepted,Amou
nt_Rejected,Received_Date,Comment,Is_Receiving_By_Count,Amount_Currency

PO15,1,JCN Technologies (AN - Fulfillment FT),14,,20,60,,,10,,,,Tue May 27 09:01:17
IST 2010,Partially Received LI1,TRUE,USD
```

Amount-Based Receipt Import

To perform amount-based receipt import, set the `Receiving_By_Count` column in `ReceiptImport.csv` to `False` .

For example:

```
     UTF8


Receivable_Id,LineItem_Num,Supplier,Supplier_Location,Commodity_Code,Quantity,Unit_P
rice,Man_Part_Num,Item_Part_Num,Number_Accepted,Number_Rejected,Amount_Accepted,Amou
nt_Rejected,Received_Date,Comment,Is_Receiving_By_Count,Amount_Currency

PO15,1,JCN Technologies (AN - Fulfillment FT),14,,20,60,,,,,150,,Tue May 27
09:01:17 IST 2010,Partially Received LI1,FALSE,USD
```

Partial Receipt Import

When you perform partial receiving through receipt import, the subsequent import of partial or fully received receipts should have a change in the `Received_Date` column for the receipts to be created successfully.

If the CSV files used in subsequent receipt imports (for the same receivable item) have the same `Received_Date` value, the import is task successful, but the receipts are not created.

Consider the following receivable line item in the `ReceiptImport.csv`:

```
     UTF8
```

```
Receivable_Id,LineItem_Num,Supplier,Supplier_Location,Commodity_Code,Quantity,Unit_P
rice,Man_Part_Num,Item_Part_Num,Number_Accepted,Number_Rejected,Amount_Accepted,Amou
nt_Rejected,Received_Date,Comment,Is_Receiving_By_Count,Amount_Currency

PO15,1,JCN Technologies (AN - Fulfillment FT),14,,20,60,,,10,,,,Tue May 27 09:01:17
IST 2010,Partially Received LI1,TRUE,USD
```

Here, the total quantity of the purchase order is 20, where as you have performed a partial receiving for 10 numbers only.

The next time you do a receipt import for the same receivable line item (that is PO15, in this example), you can choose to fully receive the remaining quantity or do a partial receiving again. In either event, when using the same CSV file again, if you do not change the Received_Date value, the receipts are not created when the Process Imported Receipts scheduled task is run.

For example, the next time you do a partial (or full) receiving of the same receivable line item, the `ReceiptImport.csv` should read as follows:

```
     UTF8



Receivable_Id,LineItem_Num,Supplier,Supplier_Location,Commodity_Code,Quantity,Unit_P
rice,Man_Part_Num,Item_Part_Num,Number_Accepted,Number_Rejected,Amount_Accepted,Amou
nt_Rejected,Received_Date,Comment,Is_Receiving_By_Count,Amount_Currency

PO15,1,JCN Technologies (AN - Fulfillment FT),14,,20,60,,,10,,,,
     Tue May 27 09:11:17 IST 2010
    ,Partially Received LI1,TRUE,USD
```

Receipts Import

You can import receipt information from Ariba Administrator.

In Ariba Administrator, run the `Import Receipt` data import task from the **Integration Manager** workspace. This task creates the receipt data and stores it as a staged object.

The `Process Imported Receipts` scheduled task creates the receipt in the buying solution from the staged object. This scheduled task also links the created receipt to the corresponding purchase order that already exists in the buying solution.

In the default configuration, your buying solution is configured to import receipts information using the file channel method. You can also use web services to import receipts created in the ERP system. To configure this feature, contact SAP Ariba Customer Support.

> ⓘ Note
>
> When the **Restrict receiving against deleted contract line items** (`Application.Receiving.RestrictReceivingAgainstDeletedContractLineItems`) parameter is enabled:
>
> - For the file channel, if both deleted contract line items and active line items are present in the receipt imported using the **Import Receipt** task, the deleted line items are not processed. Instead, in the `StagedReceiptExport.csv`, the reason for failure will be updated as "**ERROR_NO_RI_FOUND**" when

the **Process Imported Receipts** scheduled task is run. Consequently, receipts will be generated only for the active line items.

- For web services, if both deleted contract line items and active line items are present in the receipt imported using the **Import Receipt** task, the receipt is created only for the active line items. Deleted line items are not processed, and the output payload displays the receipt ID for active line items along with a warning message for deleted line items.
- The user cannot view the deleted contract line items in the receipt created after running the file channel or web services task for receiving.

For more information, see Restrict receiving against deleted contract line items.

If your site is enabled for line-level receiving, you can manage receipts at the line-level of an order or a contract using a combination of the following options:

- Manually receive a set of line items in the buying solution.
- Import receipts from external ERP systems for the rest of the items on the order.

If your site is enabled for including subcontractor information in receipts, the receipts data (`ReceiptImport.csv`) imported from external systems can include a field (`DateOfDelivery`) that specifies the date when the items were delivered. If the date of delivery is not specified in a receipt for a subcontractor, an error message is displayed and the receipt import fails. For information on including subcontractor information in receipts, see Receipts for Orders from Subcontractors.

> **Note**
>
> There are no approval rules configured for imported receipts in the default configuration. Approvers are not added to the workflow when they are imported from an external ERP system.

Related Information

Purchasing Guide for Procurement Professionals

Status of Import Receipt

The import receipt transaction provides status information about receipt import.

The following table lists and describes import receipt status messages.

Status	Description
`Waiting`	The Import Receipt integration event is run and the receipt data is successfully created as a staged object.
`Processed`	The **Process Import Receipts** scheduled task is run and the staged object of the receipt goes into the **Processed** state. The receipt is then successfully created in the buying solution and stored in the **Approved** state.
`Failed`	The scheduled task is run, but the receipt is not created.

When the receipt is successfully created, the status of the receipt object changes from `Waiting` to `Processed` . After the final receipt is created from the staged object, the `AribaReceiptID` field in the CSV file is manually updated with the unique ID of the newly created receipt.

If the receipt creation fails, the status changes to `Failed` , and the corresponding `FailedReason` field in the CSV file is updated with the reason for the failure.

> **Note**
>
> If you have performed a partial receiving of an order; the next time you receive for the same order (be it partial or full), and do a receipt import using the same CSV file.

Importing Receipts with Custom Fields

To include custom fields (also called flex fields) in your receipt import, you must ensure that any custom fields you have created for requisitions in the external system are also created and made available for import in the buying solution.

To make custom fields available for import, contact your Designated Support Contact to log a service request so an SAP Ariba Customer Support representative follows up to create these fields accordingly.

> **Note**
>
> When providing the customized fields' information, ensure you give the exact name and type of the fields that you have created in the external system.

After configuration, all flex fields in the receipts are available for import as well as for export when you run the `Receipt Status Export` task and generate a report on the receipt import status is generated.

Conditions When Receipt Import May Fail

When importing receipts information, be aware that some conditions can cause receipts creation to fail.

Conditions that cause receipt creation failure include:

- There is no purchase order in the buying solution that can be mapped to the new receipt.
- There is a mismatch in the line item mapping between the receipt and its corresponding purchase order.
- The purchase order is not in the **Ordered** or the **Receiving** state.
- A manual receipt already exists for the purchase order. If not enabled for your site, you cannot do a receipt import and also create receipts manually for the same receivable.
- There is a mismatch in the receiving type (for example, By Amount and By Quantity).
- The parameter values for the integration task are not configured correctly.

> **Note**
>
> Receipts already created in the buying solution (partially or for all line items) cannot be imported again using the `Import Receipt` integration event.

Common Causes of Errors for Receipts Imported Using Web Services

If you use web services to import receipt data from the ERP system to create receipts in the buying solution, receipts are created only if all items corresponding to an order or a contract are processed successfully in the buying solution. For example, if a receipt imported from the ERP system consists of 10 items, suppose one of the items could not be processed due to errors, the buying solution does not create receipts for the remaining items. This behavior is different in the case of line-level receiving using the CSV file channel method, where the buying solution supports creation of partial receipts for those items for which data is imported successfully in the buying solution.

The following are the common causes of errors when receipts associated with an order or contract are imported from external ERP systems using web services:

- The receipts are incorrectly associated with a service order.
- The receipts are incorrectly associated with an order that is in the **Ordering** state.
- No matching order or contract is available in the buying solution.
- The associated order or contract is closed for receiving.
- The corresponding line item on the associated order or contract is closed for receiving.
- The request message received from the ERP system does not include the `ERPReceiptNumber` for one or more items.
- The quantity (for quantity-based receiving) or amount (for amount-based receiving) for an item on a receipt has been incorrectly specified as zero.
- The external receipts incorrectly include an item that was rejected by the supplier while processing the associated order.
- Unexpected failures while external receipts are processed in the buying solution.
- The receipts incorrectly include an item, which has quantity as zero in the associated order or contract.

Receiving on Imported Receipts

Imported receipts have no restrictions or limits on the receiving amount or quantity. When receiving against orders originating from imported receipts, you can over-receive based on both quantity and/or amount. There are no validation checks performed against receiving on imported receipts.

Negative receiving is supported for imported receipts.

> ### Note
>
> Enable **Restrict receiving against deleted contract line items** (`Application.Receiving.RestrictReceivingAgainstDeletedContractLineItems`) parameter to restrict the receiving based on amount or quantity against deleted contract line items in the receipts imported through web services or file channel when the associated contract version changes. For more information, see Restrict receiving against deleted contract line items.

Related Information

Purchasing Guide for Procurement Professionals

Export Receipt Status Task

You can export receipts information by using the Ariba Administrator.

In Ariba Administrator, run the `Export Receipt Status` data export task. This task is typically run to verify the status of the receipt creation in the buying solution.

The export receipt data shows the number of receipts that were created for the imported records (purchase order line items).

> **Note**
>
> - The `Export Receipt Status` task exports only those receipts that are in the **Failed** and **Processed** states.
> - When the **Remove delimiters from export receipt fields** (`Application.Receiving.RemoveDelimitersFromExportReceiptFields`) parameter is enabled, the delimiters are removed from the `POQuantity` and `TotalNumberAccepted` fields in the CSV file exported when the `Export Receipts` task is run, and from the `Quantity` and `Number_Accepted` fields in the CSV file exported when the `Export Receipt` Status task is run. This prevents integration errors when the buying solution is integrated with an external ERP system. For more information, see Remove delimiters from export receipt fields.

Following is an example of the `ReceiptStatusExport.csv` file:

```
UTF-8


Receivable_Id,LineItem_Num,Supplier,Supplier_Location,Commodity_Code,Quantity,Unit_P
rice,
Unit_Price_Currency,Man_Part_Num,Item_Part_Num,Number_Accepted,Number_Rejected,Amoun
t_Accepted,
Amount_Rejected,Amount_Currency,Received_Date,Comment,Is_Receiving_By_Count,Status_S
tring,
Failed_Reason,Ariba_Receipt_ID,External_Receipt_ID

"PO12","1","JCN
Technologies","157",,"10","10","AED","USD","item-1","1","2","10","0","USD",
"Thu Jan 24 00:00:00 IST 2008","Partial Receiving","Yes","Processed",,"RC43"
```

Receipt Import Error Codes Reported by Export Receipt Status

If a receipt import fails, the reason for failure and the corresponding error code is listed in the **Failed Reason** field in the CSV file that you export by running the `Export Receipt Status` export task.

The following table lists reasons for receipt import failure and their corresponding error codes.

Reason for Failure	Error Code
Purchase order can't be mapped to the new receipt.	ERROR_NO_DR_FOUND
Line item mismatch between receipt and purchase order.	ERROR_NO_RI_FOUND
Purchase order not in Ordered or Receiving state.	ERROR_RECEIVING_NOT_ALLOWED
A manual receipt already exists.	ERROR_IMPORT_NOT_ALLOWED
Mismatch is receiving type.	ERROR_INVALID_RECEIVING_TYPE
Parameters in integration task not configured correctly.	ERROR_REQD_FIELDS_NULL
Reason for failure unknown.	ERROR_UNKNOWN_REASON
Parameter values are not of type DirectReceivables or ReceivableLineItem-Collection.	ERROR_OBJECT_NOT_RECEIVABLE
• When receiving by Quantity: Quantity Accepted and Quantity Rejected values are zero • When Receiving by Amount: Amount Accepted and Amount Rejected values are zero	ERROR_ZERO_RECEIPT

Exporting Receipts to SAP ERP

The buying solution integrated with SAP ERP processes approved receipts and exports them in real time.

Receipt export in an SAP ERP environment happens in real time, that is, all approved receipts get exported to the external SAP ERP as soon as they are approved, without having to remain in the **Awaiting Processing** state till a scheduled task is run. In the user interface, when a receipt is approved, it moves from the **Processing** state to the **Processed** state without any delay.

For other ERP systems, in the buying solutions version 10s1 SP7 or higher, partially or fully approved receipts are moved into the **Awaiting Processing** state until a scheduled task runs to export the approved receipts to an external ERP. This task run once a day, meaning that approved receipts are not processed and exported to the external ERP in real time.

Receipt Export Errors

Receipt export can fail due to a system error or an error in the ERP system.

If an error occurs in the buying solution during receipt export, (for example, a network issue), such receipts remain in the **Processing** state.

An exported receipt may not be processed in the external ERP due to various reasons (for example, a duplicate receipt exists, etc.). In such an event, the error status is imported back to the buying solution and the receipt status is set to **Awaiting Processing**.

When a receipt export error occurs, the failed receipts (receipts in the **Processing** and **Awaiting Processing** states) are picked up for processing by an associated scheduled task.

Asynchronous Import of Receipts

You can import goods receipts asynchronously if you have integrated your SAP Ariba Procurement solutions with your SAP back-end system through the SAP Integration Suite, managed gateway for spend management and SAP Business Network. You can import the receipts asynchronously using the new **Import Receipts from an External Application Asynchronously** web service task in the **Integration Manager** workspace. This task can be configured for use with SAP Integration Suite, managed gateway for spend management and SAP Business Network. This is different than the existing web service real-time receipt import task, which imports receipts in a synchronous manner. The end point URL for the new web service task must be configured in the buying solution from SAP Ariba. Customer administrators must enter the URL in the **Integration Manager** page.

Prerequisites

Before using this feature, ensure the following:

- The *Realtime receipts import* feature is enabled on your site.
- Support for the SAP Integration Suite, managed gateway for spend management and SAP Business Network is enabled in your solution.
- Support for asynchronous integration events for the web service channel in SAP Integration Suite, managed gateway for spend management and SAP Business Network is enabled in your solution.

Export of Receipt Data

The receipt export feature in SAP Ariba Buying and Invoicing and SAP Ariba Buying (when receiving is enabled) lets you extract receipt data from the buying solution and export it as comma-separated value (CSV) files.

For sites enabled to allow users to edit the unit price of items on partially or fully received orders, when receipts are exported for change orders in which the unit prices of line items have been changed, the receipt export task also exports the receipts associated with the previous version of the orders.

Receipts are queued for export each time certain status changes occur. You can ask your SAP Ariba Customer Support to configure the export queue statuses for receipts for your site.

> **Note**
>
> Standard export data are enhanced periodically to include more fields. When added, new fields are always inserted at the end of the record in all exported CSV files. Customers that implement further integrations with these exported CSV files must ensure that their integrations are robust enough to ignore these newly added fields if they are not required.

> **Restriction**
>
> SAP Ariba exports receipt line items to SAP S/4HANA regardless of the item price. As a result, receipt line items with a price of zero and a non-zero received quantity are also exported. Exclusion of zero-price receipt line items is not supported because it can result in inconsistencies between the purchase order and receipts in SAP S/4HANA.

Receipt Integration with ERP Systems [page 309]

Workflow of a Failed Receipt Export [page 310]

Exporting Receipts [page 311]

Receipt Data [page 312]

Receipt Integration with ERP Systems

Integration between the buying solutions and third-party systems, such as SAP and PeopleSoft ERPs is supported with both web services and file-based channels..

Receipt Integration with SAP ERP

End-to-end integration is achieved using SAP NetWeaver PI. SAP NetWeaver PI acts as an EAI tool to ensure that SOAP messages (Request and Response) are exchanged between the buying solutions and SAP ERP.

Receipt transactions are supported with both web service and file-based integration.

Receipt Integration with PeopleSoft ERP

To integrate with a PeopleSoft ERP, the buying solution publishes a SOAP message for the required integration event. Customers need to use an EAI tool or application that can consume this SOAP message and send an acknowledgment back as an ID or error.

> **Note**
>
> This transaction can also be configured to use the file system for integration.

Workflow of a Failed Receipt Export

For every purchase order created in the buying solution, there is at least one receipt created. The receipt object is in the **Awaiting Processing** state and moves to the **Processing** state when exported to an external ERP. If the receipt is exported successfully, the receipt state moves to **Processed**.

If there is a receipt export failure, the status moves back to **Awaiting Processing**.

If a receipt export to the external ERP fail because of integration-related issues, an email is sent to the concerned groups notifying them about the failure. The buying solution then retries to send failed receipts utill the task is successful.

The following system behavior is observed when a receipt export fails:

1. The buying solution creates one or more receipts (depending on how the order is received) for every order and attempts to export the receipts to an external ERP. On creation, the receipt moves into the **Awaiting Processing** state.
   This status changes to **Processing** when the buying solution is exporting the receipt to the ERP.

> **Note**
>
> - If you are using the File Channel for communications, you must manually export the receipts.
> - If you are using the Web Services channel, all receipt exports are performed in real time.

1. If the export is successful, the receipt status moves to **Processed**. If the receipt export fails, the receipt status moves back to **Awaiting Processing**.

   > **Note**
   >
   > Everytime the receipt export fails, the buying solution triggers an email notification that is sent to all users belonging to the **Receiving Administrator** group.

2. The **Process Pending Receipts** scheduled task then picks up all receipts that are in the **Awaiting Processing** state.

   > **Note**
   >
   > Receipts in the **Processing** state that have been updated 24 hours prior to the time that the scheduled task runs, are also picked up for export to the external ERP.

> **Note**
>
> There is no permanent failure because the scheduled task continues to resend failed receipts to the external ERP until successful. An email notification is sent the **Receiving Administrator** group every time the receipt export fails.

Exporting Receipts

You can export receipts from Ariba Administrator.

Procedure

1. On the dashboard, click **Manage** > **Core Administration**.
2. Click **Site Manager**.
3. Click **Data Import/Export**, and click the **Export** tab.
4. Search for the **Export Receipts** data export task and click **Export**.

   If your site is integrated with an external ERP, use the **Export Receipts to ERP** export task instead.
5. Select a date range for the export by choosing dates in the **From** and **To** fields.
6. Click **Run** to export the data.

   > **Note**
   >
   > When the **Remove delimiters from export receipt fields** (`Application.Receiving.RemoveDelimitersFromExportReceiptFields`) parameter is enabled, the delimiters are removed from the `POQuantity` and `TotalNumberAccepted` fields in the CSV file exported when the `Export Receipts` task is run, and from the `Quantity` and `Number_Accepted` fields in the CSV file exported when the `Export Receipt` Status task is run. This prevents integration errors when the buying solution is integrated with an external ERP system. For more information, see Remove delimiters from export receipt fields.
7. After the data has finished downloading, click **Open** to view the contents of the ZIP file or **Save** to save the file to your local hard drive.

   > **Note**
   >
   > When suppliers are categorized as subcontractors, the `ReceiptDetailExport.csv` file includes a **DateofDelivery** field. This field contains the delivery date, followed by the time displayed as 00:00:00. For more information, see Receipts for Orders from Subcontractors.
   >
   > > **Example**
   > >
   > > Sun Mar 2 00:00:00 PST 2025

Receipt Data

When you run the `Export Receipts` data export task, Ariba Administrator exports the data to a ZIP files that contains three CSV files.

The CSV files in the data export ZIP file are:

- `ReceiptHeaderExport.csv`
- `ReceiptDetailExport.csv`
- `ReceiptAssetExport.csv`

Related Information

Receipt Header Data [page 312]
Receipt Detail Data [page 313]
Receipt Asset Data [page 314]

Receipt Header Data

The `ReceiptHeaderExport.csv` file contains header information for receipts.

The following table describes the columns in `ReceiptHeaderExport.csv`.

Column	Description	Format
`Id`	The unique reference ID of the receipt. This value is used to match receipts in the header, detail, and asset data files.	String (255)
`EventDate`	The date on which this status change occurred.	Date
`EventStatus`	The status of the receipt.	String (255)
`ReceiptNumber`	The unique internal identifier of the receipt, for example, "RC100."	String (50)
`PONumber`	The system ID of the purchase order with which the receipt is associated, for example, "PO100."	String (255)
`Requester`	The unique internal identifier of the user who submitted the receipt.	String (50)
GoodsReturnTracking Number	A unique alpha-numeric number for that receipt. This is for identification purposes and used when returning the items to the supplier.	String (100)
`Comments`	Receipt comments, include date and creator information. For example: `3/31/2008: 12:08 PM Arnold Davis: comment`	String (2000)

Following is an example of `ReceiptHeaderExport.csv`:

```
"UTF-8"
```

```
Id,EventDate,EventStatus,ReceiptNumber,PONumber,Requester,GoodsReturnTrackingNumber,
Comments

"2","03/26/2008 20:20:59 -0700","Submitted","RC348","PO265","wtalagtag",
"4","03/26/2008 20:20:59 -0700","Composing","RC348","PO265","wtalagtag",
"7","03/26/2008 20:21:19 -0700","Submitted","RC349","PO266","wtalagtag",
"8","03/26/2008 20:21:19 -0700","Composing","RC349","PO266","wtalagtag",
"11","03/26/2008 20:21:39 -0700","Submitted","RC350","PO267","wtalagtag",
```

Receipt Detail Data

The `ReceiptDetailExport.csv` file contains line item information for receipts.

The following table describes the columns in `ReceiptDetailExport.csv`.

Column	Description	Format
`Id`	The unique reference ID of the receipt. This value matches a payment defined in `ReceiptHeaderExport.csv`.	String (255)
`ReceiptNumber`	The unique internal identifier of the receipt, for example, "RC100."	String (50)
`ReceiptLineNumber`	The line number of this line on the receipt.	Numeric
`ReceivableLineNumber`	The line number of this line on the purchase order.	Numeric
`UnitOfMeasure`	The unit of measure for this line.	String (50)
`NumberAccepted`	The number of items for this line that were accepted.	Numeric
`NumberRejected`	The number of items for this line that were rejected.	Numeric
`TotalNumberAccepted`	The total number of items (on all receipts for this purchase order) that have been accepted for this line.	Numeric
`TotalNumberRejected`	The total number of items (on all receipts for this purchase order) that have been accepted for this line.	Numeric
`AmountAccepted`	The amount of this line that has been accepted.	Numeric
`AmountAccepted`_Currency	The currency for the amount accepted.	String (50)
`AmountAccepted_InBaseCurrency`	The approximate amount accepted in the default currency, using the current exchange rate.	BigDecimal
`TotalAmountAccepted`	The total amount of money accepted in a particular expense.	BigDecimal
`AmountRejected`	The amount of this line that has been rejected.	BigDecimal
`TotalAmountRejected`	The total amount of money rejected in a particular expense.	BigDecimal
`TotalAmountRejected_Currency`	The total amount of money, in the base currency, which was rejected.	BigDecimal
`AmountRejected_Currency`	The currency of the amount rejected.	String (50)

Column	Description	Format
`TotalAmountAccepted_Currency`	The total amount of money, in the base currency, which was accepted.	BigDecimal
`AmountRejected_InBaseCurrency`	The approximate amount rejected in the default currency, using the current exchange rate.	BigDecimal
`TotalAmountAccepted_InBaseCurrency`	The total amount accepted in the default currency, using the current exchange rate	BigDecimal
`TotalAmountRejected_InBaseCurrency`	The total amount rejected in the default currency, using the current exchange rate.	BigDecimal
`ReturnBy`	Indicates how the rejected goods are to be returned; whether as a replacement of goods or by credit.	String (20)
`POQuantity`	The quantity of items ordered for this line.	Numeric
`POAmount`	The amount ordered for this line.	Numeric
`POAmount_Currency`	The currency for the amount ordered.	String (50)
`POAmount_InBaseCurrency`	The amount ordered in the default currency using the current exchange rate.	Numeric
`ReceivingDate`	Receiving date (`Date Received`) of line items in a purchase requisition.	Date

Following is an example of `ReceiptDetailExport.csv`:

```
"UTF-8"


IId,ReceiptNumber,ReceiptLineNumber,ReceivableLineNumber,UnitOfMeasure,NumberAccepted,
NumberRejected,TotalNumberAccepted,TotalNumberRejected,POQuantity,POAmount,AmountAccepted,
POAmount_Currency,AmountAccepted_Currency,POAmount_InBaseCurrency,AmountAccepted_InBaseCurrency,
TotalAmountAccepted,AmountRejected,TotalAmountRejected,TotalAmountRejected_Currency,
AmountRejected_Currency,TotalAmountAccepted_Currency,AmountRejected_InBaseCurrency,
TotalAmountAccepted_InBaseCurrency,TotalAmountRejected_InBaseCurrency,ReturnBy,ReceivingDate

"2","RC348","1","1","EA","0","0","0","0","0.0000","USD","0.0000","10","300.0000","USD","300.0000"
"4","RC348","1","1","EA","0","0","0","0","0.0000","USD","0.0000","10","300.0000","USD","300.0000"
"7","RC349","1","1","EA","0","0","0","0","0.0000","USD","0.0000","10","300.0000","USD","300.0000"
"8","RC349","1","1","EA","0","0","0","0","0.0000","USD","0.0000","10","300.0000","USD","300.0000"
```

Receipt Asset Data

The `ReceiptAssetExport.csv` file contains asset data for received items.

The following table describes the columns in `ReceiptAssetExport.csv`.

Column	Description	Format
`Id`	The unique reference ID of the receipt. This value matches a payment defined in `ReceiptHeaderExport.csv`.	String (255)
`ReceiptNumber`	The unique internal identifier of the receipt, for example, “RC100.”	String (50)
`ReceiptLineNumber`	The line number of this line on the receipt.	Numeric
`ReceivableLineNumber`	The line number of this line on the purchase order.	Numeric
`Location`	The location where this item was received.	String (50)
`SerialNumber`	The serial number of the received item.	String (50)
`TagNumber`	The tag number of the received item.	String (50)
UnitNumber	The unit number of the received item.	Numeric

Following is an example of `ReceiptAssetExport.csv`:

```
"UTF-8"


Id,ReceiptNumber,ReceiptLineNumber,ReceivableLineNumber,Location,SerialNumber,TagNum
ber,
UnitNumber

"2","RC348","1","1",,,,
"4","RC348","1","1",,,,
"7","RC349","1","1",,,,
"8","RC349","1","1",,,,
"11","RC350","1","1",,,,
"12","RC350","1","1",,,,
"13","RC319","1","1",,,,
```

Export of Accrual Information

When the accrual accounting methodology is used, transactions are counted when an order is made, the item is delivered, or the service is provided, regardless of when payment is actually made. Expenses are counted when the goods or services are received, and income is counted when the sale occurs.

> **Note**
>
> The Accruals export feature is only available to existing SAP Ariba customers using (SAP Ariba Buying and Invoicing and SAP Ariba Buying Release 10s1).

The accrual export feature provides accrual history information, in CSV file format, for a specified time frame. The exported CSV file contains a variety of transaction information, such as tracking:

- Fully cleared orders with nothing to accrue. An order is considered "cleared" when the associated invoice has been reconciled.
- Uncleared orders that are accrued and where the receipt amount is equal to the invoice amount.
- Uncleared orders that are accrued and where the receipt amount is not equal to the invoice amount.
- Uncleared orders that are accrued because goods are received, but invoices are yet to be received from suppliers.
- Uncleared orders that are accrued because invoices have been received from suppliers, but goods are yet to be received.

Partially cleared orders that have some accrual left over because no more invoices are expected from the supplier for the quantity of goods supplied.

Accrual Information in Accrual Export [page 317]

Considerations for Exporting Accrual Information [page 318]

Export Accrual Transactions Data Export Task [page 318]

Export Transaction Accruals Data Export Task [page 320]

Related Information

Exporting Financial Transaction Information [page 316]

Accrual Information in Accrual Export

The buying solution captures transaction details and posts them to an “export queue” whenever an event occurs that triggers a change to a relevant accumulator for an item on a purchase order, contract, or an invoice.

The relevant accumulators are:

- Amount billed
- Amount cleared
- Amount accepted

The events that trigger changes to the relevant accumulators include:

- An item is received (the first approver approves a receipt).
- An invoice is received and an invoice reconciliation document is created.
- An invoice reconciliation document is fully approved and transitions to Paying.
- An invoice reconciliation document is manually matched to another purchase order or contract, or each line is manually matched to a different line of the purchase order or contract.
- A payment is made and an invoice reconciliation document transitions to Paid.
- An invoice reconciliation document is rejected or canceled.
- A change order is created.

Transaction Amount in Accrual Export

Whenever a triggering event changes the relevant accumulators, the buying solution captures the current transaction details. These details include the transaction amount, which is the net effect of the liability as a result of the triggering event.

The buying solution calculates the transaction amount by subtracting the accrued amount before the transaction from the accrued amount after the transaction:

```
         TransactionAmount = (AccruedAmount_after_transaction) -
(AccruedAmount_before_transaction)
```

The buying solution calculates the accrued amount by comparing the accumulated accepted and billed amounts, finding the maximum of these two accumulated amounts, and subtracting the accumulated amount cleared:

```
AccruedAmount = Max (AmountAccepted, AmountBilled) - (AmountCleared)
```

The net accrual at any point in time is the summation of all the previous transaction amounts posted for the approvable document. The accrual for a purchase order starts with a receipt or invoice, and closes when the invoice is reconciled or cleared for payment.

Encumbrance Transactions in Accrual Export

An encumbrance is an anticipated expenditure that is used to predict future liabilities. When the encumbrance transaction history feature is enabled for your site, the buying solution captures encumbrance transaction information for all relevant transactions.

The buying solution makes encumbrance entries for the estimated amount of future liabilities when an order enters the **Ordered** status. The actual liabilities are recorded and the encumbrance entries are reversed (and thus eliminated) when an order is invoiced or received.

The encumbrance transaction history feature is disabled in the default configuration. If you want to enable this feature, contact SAP Ariba Customer Support.

Considerations for Exporting Accrual Information

When exporting accrual information, be that some charges (for example, taxes) are not considered for accrual calculations.

Specifically, be aware that:

- Taxes, shipment, handling charges, and discounts are not considered for accrual calculation.
- Purchasing card orders, charges based on accruals, expense report-based accruals, credit memos, partial payments, and items on invoices that do not match an existing purchase order or contract are not included in accrual transactions.
- Accrual transactions for manual matches of different document types (for example, a purchase order-based invoice matched to a contract or vice versa) are not supported.

Export Accrual Transactions Data Export Task

You can export accrual information by running the `Export Accrual Transactions` data export task in Ariba Administrator.

You can also export accrual information by using the SAP Ariba data transfer tool.

The buying solution provides three data import tasks to facilitate export of accruals data:

Import Task	Description
Export Accrual Transactions	Exports information on each transaction (exports one line for every transaction) against a PO, contract, and a non-PO invoice.

Import Task	Description
Export Accrual Summary for Non-PO Invoices	Exports data containing a summary (total) of accrual postings (Amount Accepted, Billed, Cleared, Transaction Amount) against a Non-PO invoice. **Note** This export tasks has been deprecated in the current release of SAP Ariba Buying and Invoicing and SAP Ariba Buying. Also, this task will not be available in any buying solution starting from the next major release.
Export Accrual Summary for Orders and Contracts	Exports data containing a summary (total) of accrual postings (Amount Accepted, Billed, Cleared, Transaction Amount) against an Order and a No Release Contract. **Note** This export tasks has been deprecated in the current release of all SAP Ariba solutions. Also, this task will not be available in any buying solution starting from the next major release.

Exporting Accrual Information

You can export accrual information to a zip file.

Procedure

1. On the dashboard, click **Manage** > **Core Administration** > .
2. Click **Site Manager**.
3. Click **Data Import/Export**, and click the **Export** tab.
4. Search for the **Export Accrual Transactions** data export task and click **Export**.
5. Select a date range for the export by choosing dates in the **From** and **To** fields. Ariba Administrator exports all accrual information created during this time period. The first time you export the data, the **From** text box is empty.
6. Click **Run** to export the data.
7. After the data has finished downloading, click **Open** to view the contents of `AccruedTransactionExport.zip`, or click **Save** to save the file to your local hard drive.

Loading Accrual Information into an ERP System Database

You can use the DB Connector to load accrual information into an ERP system database.

Supported databases include:

- Oracle

- DB2
- SQL Server

Additional References

Integration toolkit guide

Export Transaction Accruals Data Export Task

The `Export Transaction Accruals` data export task exports accrual information in CSV format to the file `AccruedTransactionsExport.csv`.

`AccruedTransactionExport.csv` includes common data (data common to PeopleSoft, SAP and Generic ERP systems, and accounting data, which differs according to your ERP system.

Related Information

Common Data in AccruedTransactionExport.csv [page 320]
Accounting Data in AccruedTransactionExport.csv [page 322]

Common Data in AccruedTransactionExport.csv

The columns in the `AccruedTransactionsExport.csv` file are common to PeopleSoft, SAP, and Generic ERP. The information in `AccruedTransactionsExport.csv` should not be considered a report.

Column Name	Description	Format
`TransactionDate`	The date on which the transaction posted in the format `MM/dd/yyyy HH:mm:ss ZZZZ`.	Date
`DocumentNumber`	The number of the approvable document for which the transaction was posted. For example, POnn (purchase order), Cnn (contract), or INVnn (invoice).	String (64)
`LineNumber`	The line number of the line item.	Number
`Requester`	The unique internal identifier of a user. In the case of a purchase order, this is the name of the requester for the line item. Otherwise, it is the name of the requester for the approvable document.	String (50)
`LineText`	The line item description.	String (128)

Column Name	Description	Format
`SupplierID`	The unique internal identifier of the supplier for the line item.	String (50)
`SupplierName`	The user-visible name of the supplier for the line item.	String (128)
`PostingCurrency`	The currency of the transaction amount.	String (50)
`CommodityCode`	The commodity code of the line item.	Number (50)
`Domain`	The domain for the line item's commodity code.	String (50)
`ReceivedAmount`	The amount on the receipt.	Number
`BilledAmount`	The invoiced amount.	Number
`ClearedAmount`	The cleared amount is equivalent to the billed amount. Entries appear in this column when an invoice is reconciled and goes to the Paying or Paid state.	Number
`TransactionAmount`	The transaction amount, which represents the effect of the total liability of this transaction. When calculating real and future liabilities, only consider the values in the `TransactionAmount` and `EncumberedAmount` columns. The values in other amount columns are supporting values only, and indicate how the `TransactionAmount` and `EncumberedAmount` values were calculated.	Number
`OriginatingSystem`	The name of an external system.	String (50)
`OriginatingSystemReferenceID`	The system ID (such as a restock number) on an external system.	String (50)
`OrderedAmount`	The purchase order amount when the order enters the Ordered status.	Number
`EncumberedAmount`	The current encumbered amount. When calculating real and future liabilities, only consider the values in the `TransactionAmount` and `EncumberedAmount` columns. The values in other amount columns are supporting values only, and indicate how the `TransactionAmount` and `EncumberedAmount` values were calculated.	Number
`ProcurementUnit`	The purchasing unit.	String (50)

Related Information

About Encumbrance Transactions [page 318]

Accounting Data in AccruedTransactionExport.csv

`AccruedTransactionExport.csv` includes columns for accounting information. These columns are specific to your ERP system.

Related Information

Accounting Columns in AccruedTransactionExport.csv for PeopleSoft ERP [page 322]
Accounting Columns in AccruedTransactionExport.csv for SAP ERP [page 323]
Accounting Columns in AccruedTransactionExport.csv for Generic ERP [page 322]

Accounting Columns in AccruedTransactionExport.csv for PeopleSoft ERP

The `AccruedTransactionsExport.csv` file includes columns for accounting information for PeopleSoft ERP.

Column	Format
`Account`	String (50)
`BusinessUnit`	String (50)
`GLBusinessUnit`	String (50)
`Location`	String (50)
`Department`	String (50)
`Product`	String (50)
`Project`	String (50)
`StatisticsCode`	String (50)
`PSProfileId`	String (10)
`BusinessUnitAM`	String (5)

Accounting Columns in AccruedTransactionExport.csv for Generic ERP

The `AccruedTransactionsExport.csv` file includes columns for accounting information for Generic ERP.

Column	Format
`Account`	String (50)

Column	Format
Company	String (50)
BusinessUnit	String (50)
CostCenter	String (50)
SubAccount	String (50)
Product	String (50)
Project	String (50)
Region	String (50)
AccountingCombination	String (50)

Accounting Columns in AccruedTransactionExport.csv for SAP ERP

The AccruedTransactionsExport.csv file includes columns for accounting information for SAP ERP.

Column	Format
Account	String (50)
CostCenter	String (50)
GeneralLedger	String (50)
WBSElement	String (50)
Asset	String (50)
InternalOrder	String (50)
SAPSerialNumber	String (100)
SAPLimitSplitValue	String (25)

Support for the Plant Entity from SAP Backend Systems

This functionality is for new customers of buying solutions from SAP Ariba integrated with SAP backend systems. It can be turned on by setting the parameter **Enable plant-based defaulting** (`Application.Common.EnablePlantBasedDefaultingForSAP`).

It adds a new header-level field named **Plant** in purchase requisitions, representing a functional entity corresponding to the Plant field in SAP backend systems, which is distinct from the existing line-level field in SAP Ariba solutions named **Plant**, which represents a shipping address. The new field is present in the user profile and is added to the requisition header by default. This header-level **Plant** value can be used to default delivery addresses, as well as the company code and purchasing organization, when a requisition is created. The **Plant** field is added by default when requisition line items are added.

The new field is required to represent an organizational or business entity like a manufacturing plant. It also ensures parity with SAP ERP systems, where the corresponding field refers to an organizational entity. The header-level **Plant** field can be used to default delivery addresses, as well as the company code and purchasing organization, when a requisition is created. To ensure that the new field can be used alongside the existing **Plant** field, which represents a shipping address, it must be renamed to something appropriate like Delivery Address using standard customization.

Related Information

Workflow for Defaulting Based on the Plant Entity [page 324]
Setting the Plant Field in the User Profile [page 325]
Setting the Value for the Plant Field for a User from the User Manager Workspace [page 325]

Workflow for Defaulting Based on the Plant Entity

The defaulting of values like company code based on the Plant entity from SAP backend systems happens in the following sequence:

Plant-based defaulting using the new header-level Plant field happens in the following sequence:

1. The **Plant** field in the user profile is defaulted with plant data imported from the SAP backend system. The user can choose a new plant value in the user profile or retain the imported plant value.
2. When the user creates a new requisition, this plant value is used to default the **Plant** field in the header.
3. This value in the header-level **Plant** field is used to default the new **SAPPlant** field on the requisition line item.
4. The company code corresponding to the plant value is used to default the requisition's company code.
5. The address corresponding to the header-level **Plant** field will be defaulted in the line-level **Plant** field (the preexisting field used for shipping addresses). The user can change this default setting by choosing another address from the line-level **Plant** field's dropdown.

Setting the Plant Field in the User Profile

Context

Customers must first import plant details and make sure it is available by default in their users' profile.

Users can choose a different plant by editing their profile.

Procedure

1. Click the user name at the top right corner of the dashboard and choose the **Change your profile** option from the dropdown.
2. Click the **Account/Ship** tab.

   The **Plant** field in the **Change your organization information** section shows the default value imported from the SAP ERP system.
3. Click the **Plant** field's dropdown and chose the required value.

   The values in the **Company Code** and **Delivery Address** fields change to values corresponding to the chosen plant value.
4. Click **Submit** to send the user profile change for approval.

   The **Plant** field will be updated with the new value only after the user profile change is approved.

Setting the Value for the Plant Field for a User from the User Manager Workspace

Context

The **Plant** field's value can be set by administrators for other users from the administrator interface.

Procedure

1. Click **Core Administration** > **User Manager**
2. Click **Other User Info** and then search for the user for whom the job profile needs to be set.
3. Click **Actions** for the selected user and choose Edit from the dropdown.

This opens the page to edit the user information.

4. Click the **Accounting** tab.

   This tab displays the **Plant** field along with all the other accounting fields such as **Company Code** and **Purchase Org** among others. The **Plant** field has the default value imported from the SAP ERP system.

5. Click the **Plant** field and choose a value from the dropdown to change the default value.

   The value of the **Company Code** field changes to the code corresponding to the chosen plant value.

6. Click **Save** to save the changes.

Default Purchasing Organization Based on the Selected Plant

In buying solutions from SAP Ariba, filtering of values available for selection in the **Purch Org** chooser can be turned on. Filtering is based on a combination of the selected plant (the ShipTo address) and the corresponding company code. This functionality allows you to set a default purchasing organization in the **Purch Org** corresponding to the selected plant. This ensures the correct purchasing organization is assigned for the selected plant and company code.

This functionality applies only to customers who have integrated their buying solutions from SAP Ariba with SAP ERP. Additionally, this functionality is designed to work with the plant-based defaulting feature. It can be turned on by enabling the parameter **Enable setting the default purchase organization using a combination of plant and company code** (`Application.Procure.DefaultSAPPurchaseOrgUsingPlant`). When the parameter is enabled, the default purchase organization is set based on a combination of plant and company code only when the requester purchase organization is set. To set the purchase organization even when the requester purchase organization is not set, you can enable the **Default purchase organization based on plant and company code even if not set for requester** (`Application.Procure.DefaultPurchaseOrgBasedOnPlantAndCompanyCodeEvenIfNotSetForRequester`) parameter.

Prerequisites for setting the default purchasing organization based on the plant

- Turn on the plant-based defaulting feature along by setting the parameter **Enable plant-based defaulting** (`Application.Common.EnablePlantBasedDefaultingForSAP`)
- Import plant information with a one-to-one mapping between plant and commodity code.
- Import plant and purchasing organization combination with one-to-many mappings between plant and purchasing organization.
- Optional. To filter the purchasing organizations shown in the **Purch Org** chooser, ensure that the **Filter SAP purchasing organizations on requisitions** (`Application.Procure.FilterSAPPurchaseOrgsUsingPlants`) site configuration option is enabled. For more information, see Filter SAP purchasing organizations on requisitions.

Related Information

Enable plant-based defaulting
Enable setting the default purchase organization using a combination of plant and company code
Default purchase organization based on plant and company code even if not set for requester

Integration of Orders and Return Receipts for Goods Receipt-Based Invoice Verification

Buying organizations can integrate purchase orders and return receipts for goods receipt-based invoice verification with external ERP systems. The data import and export events related to purchase orders and receipts include fields to support the integrations.

- To enable the integration of goods receipt-based invoicing, the field that indicates whether a line item is enabled for goods receipts-based invoicing is sent during purchase order export.
- To enable the integration of return receipts,
  - The ship notice ID is received in the SAP Ariba solution during receipt import.
  - The ERP receipt number of the reference receipt and the ship notice ID are sent during receipt export.

The following integration events include the flag (`GRBasedInvoice`) that indicates whether a line item is enabled for goods receipts-based invoicing:

Web services	File
Export Purchase Orders	**Export Purchase Orders**
Export Purchase Orders Asynchronously (for SAP ERP or SAP S/4HANA)	**Export Purchase Order to ERP**
Export Change Purchase Orders	**Export Change Purchase Order**
Export Change Purchase Orders Asynchronously (for SAP ERP or SAP S/4HANA)	
Fetch Purchase Order Details	

> **Note**
>
> When the **Include currency of charge line item in PurchaseOrderExportRequest for purchase orders and change orders** (`Application.Procure.IncludeCurrencyOfChargeLineItemInPOExportRequestForPurchaseOrders`) parameter is enabled, the currency of the charge line item is included in the **PurchaseOrderExportRequest** files when purchase orders or change orders are generated. For more information, see Include currency of charge line item in PurchaseOrderExportRequest for purchase orders and change orders.

The following integration events include the ERP receipt number (`ERPReferenceReceiptNumber` for file events, `ERPReceiptNumber` for web services events) and the ship notice ID (`ASNReference`):

Web services	File
Export Receipts	**Export Receipts**
Export Receipts Asynchronously (for SAP ERP only)	**Export Receipts to ERP**

The following integration events include the ship notice ID (`ASNReference`) in the header details of receipts:

Web services	Cloud Integration Gateway
Import Receipts from an External Application	**Import Receipts from an External Application Asynchronously** (for SAP ERP only)
Import Receipts from an External Application Asynchronously (for SAP ERP only)	

Prerequisites

- For SAP ERP, SAP Integration Suite, managed gateway for spend management and SAP Business Network must be set up for the integration between the ERP and the SAP Ariba solution.
- The goods receipt-based invoice verification feature must be enabled for your site.
  Your administrator must enable the **Enable goods receipt-based invoice verification** (`Application.Procure.EnableGRBasedInvoice`) parameter.

Restrictions

The capabilities of this feature aren't supported for importing receipts in the file format.

> **Note**
>
> Customizations to simulate the feature capabilities with the file format may lead to incorrect association of the ship notice ID with the receipt.

Related Information

Goods Receipt-Based Invoice Verification
Enable goods receipt-based invoice verification
Export of Purchase Orders [page 156]
Imported Receipts [page 299]

Data Import Tasks Reference

The data import tasks reference provides tables that list data import task for procurement data, data import order for SAP Ariba Buying solutions by ERP system (PeopleSoft, SAP, and Generic), data import tasks for SAP Ariba Buying and Invoicing, and data import order for SAP Ariba Buying and Invoicing by ERP system.

Data Import Tasks for Procurement Data [page 330]

About SAP Ariba Buying solutions Data Import Order [page 331]

Data Import Tasks for SAP Ariba Buying [page 339]

Data Import Order for SAP Ariba Buying [page 343]

Related Information

Data Import Tasks for Procurement Data [page 330]
About SAP Ariba Buying solutions Data Import Order [page 331]
Data Import Tasks for SAP Ariba Buying [page 339]
Data Import Order for SAP Ariba Buying [page 343]

Data Import Tasks for Procurement Data

A table describes the data import tasks that import procurement data.

Data Import Task	CSV File(s) Imported
Import Batch Data (Batch data import tasks import CSV files from a ZIP file)	
Import Category Drivers (For SAP Ariba internal use only)	
Import Catalog View (For SAP Ariba internal use only)	
Import CSV Lookup Tables Used By Approval Rules (Batch data import tasks import CSV files from a ZIP file)	
Import L2 Charge	`Charge.csv`
Import Charge Exception Types Import Charge Exception Type Translation	`ChargeExceptionType.csv`
Import Charge Exception Type Validation Data	`ChargeExceptionTypeValidation.csv`
Import Charge Tax Code Mapping	`ChargeTaxCodes.csv`
Import Equivalent Words (For SAP Ariba internal use only)	

Data Import Task	CSV File(s) Imported
Import PCards	`PCard.csv`
Import Purchase Order Conditions	`POConditions.csv`
Import Receiving Types By Commodity Code	`ReceiptByCommmodityCode.csv`
Import Receiving Types By Part Number	`ReceiptByPartNumber.csv`
Import Remittance Data	`Remittance.csv` `RemittanceDetails.csv`
Import Transform Using Catalog Map (For SAP Ariba internal use only)	

Related Information

Administration and Data Maintenance Fundamentals

About SAP Ariba Buying solutions Data Import Order

Tables for PeopleSoft, SAP, and Generic list the order in which the `Import Batch Data` data import task imports data for the buying solutions. You can use this information as a guide when running individual data import tasks.

Related Information

Data Import Order for PeopleSoft [page 331]
Data Import Order for SAP ERP [page 334]
Data Import Order for Generic [page 337]

Data Import Order for PeopleSoft

A table shows the `Import Batch Data` data import sequence for data for PeopleSoft ERP.

	Data Import Task	CSV File(s)	SAP Ariba Buying and Invoicing
1	**Import Currency Conversion Rates**	`CurrencyConversionRate.csv`	X
2	**Import Expense Header Category Mappings**	`ExpenseHeaderCategoryMap.csv`	
3	**Import Expense Auto Types**	`AutoType.csv`	

	Data Import Task	CSV File(s)	SAP Ariba Buying and Invoicing
4	**Import ERP Commodity Codes**	`ERPCommodityCode.csv`	
5	**Import CSV Lookup Tables Used By Approval Rules**	`RuleCSVHelperFiles.zip`	X
6	**Import Per Diem Limits**	`LocationType.csv`	
7	**Import Purchase Order Conditions**	`POConditions.csv`	X
8	**Import Payment Method Types**	`PaymentMethodType.csv`	X
9	**Import Bank Identifier Types**	`BankIDType.csv`	X
10	**Import Bank Account Types**	`BankAccountType.csv`	X
11	**Import Expense Code Mappings**	`ExpenseCodeMapping.csv`	
12	**Import Account Types**	`AccountType.csv`	X
13	**Import Payment Terms**	• `PaymentTerms.csv` • `PaymentTermStepDetails.csv` • `PaymentTermSteps.csv`	X
14	**Import Accounting Combination Groups**	`AccountingCombinationGroup.csv`	
15	**Import Allowable Hierarchy Choices**	`AllowableHierarchyChoice.csv`	
16	**Import Tax Type Mappings**	`TaxTypeMap.csv`	X
17	**Import SetIDs**	`SetID.csv`	X
18	**SharedUserPSCSVPullDisplayName**	`SharedUserPSCSV.csv`	
19	**Import Bank Account Number Types**	`BankAccountIDType.csv`	X
20	**Import Receiving Types By Commodity Code**	`ReceiptByCommodityCode.csv`	X
21	**Import Accounts**	`Account.csv`	X
22	**Import Products**	`Product.csv`	X
23	**Import Tax Codes**	`TaxCode.csv`	X
24	**Import ERP Commodity Codes From PeopleSoft**	`ERPCommodityCode.csv`	X
25	**Import GL Business Units**	`GLBusinessUnit.csv`	X
26	**Import Addresses From PeopleSoft**	`Address.csv`	X
27	**Import Departments**	`Department.csv`	X
28	**Import Projects**	`Project.csv`	X
29	**Import Supplier Bank Payment Locations**	`SupplierPaymentBankLocation.csv`	X
30	**Import Buyer Bank Payment Locations**	`BuyerPaymentBankLocation.csv`	X
31	**Import Expense Tax Rate Lookup By Tax Code**	`ExpenseTaxRateLookupByTaxCode.csv`	
32	**Import Tax Code Lookups**	`TaxCodeLookup.csv`	X

	Data Import Task	CSV File(s)	SAP Ariba Buying and Invoicing
33	**Import Business Units**	`BusinessUnit.csv`	X
34	**Import PCards**	`PCard.csv`	X
35	**Import Tax Rates By Tax Code Lookup**	`TaxRateLookupByTaxCode.csv`	X
36	**Import Remittance Locations**	• `RemittanceLocation.csv` • `RemittanceLocationDetails.csv`	X
37	**Import User/Ship To Address Relationships**	`SharedUserShipToAddressMap.csv`	X
38	**Import Users Consolidated CSV File**	• `User.csv` • `UserPCardMap.csv`	X
39	**Import Expense Types**	`ExpenseType.csv`	
40	**Import ERP Commodity to Commodity Mapping**	`ERPCommodityCodeMap.csv`	X
41	**Import Accounting Combinations**	`AccountingCombination.csv`	X
42	**Import Adjustment Types**	`AdjustmentType.csv`	
43	**Import Custom Groups**	`CustomGroup.csv`	X
44	**Import Supplier Organizations**	• `SupplierIDs.csv` • `SupplierOrganizations.csv`	X
45	**Import User Approval Limits**	`UserApprovalLimit.csv`	X
46	**Import Expense Tax Code Lookup**	`ExpenseTaxCodeLookup.csv`	
47	**Import Responsible User Assignments**	`ResponsibleUser.csv`	
48	**Import Group To Child Group Mapping**	• `GroupParentGroup.csv` • `GroupChildGroupMap.csv`	X
49	**Import User Group Mapping**	• `UserParentGroup.csv` • `GroupSharedUserMap.csv`	X
50	**Import Approval Queues**	`ApprovalQueue.csv`	
51	**Import Suppliers**	• `Supplier.csv` • `SupplierLocation.csv`	X
52	**Import Receiving Types By Part Number**	`ReceiptByPartNumber.csv`	X
53	**Import Supplier Supplement Data**	`SupplierSupplement.csv`	
54	**Import Level2 Charge**	`Charge.csv`	X
55	**Import Enterprise User/Supervisor Relationships from PeopleSoft**	`SharedUserSupervisor.csv`	X
56	**Import Supplier Location Remittance Information**	• `SLRemittanceInformation.csv` • `SLRemittanceInformationDetails.csv`	X

	Data Import Task	CSV File(s)	SAP Ariba Buying and Invoicing
57	**Import Scheduled Tasks**	`ScheduledTaskData.zip`	

Data Import Order for SAP ERP

A table shows the `Import Batch Data` data import sequence for data for SAP ERP.

	Data import task	CSV file(s)	SAP Ariba Buying and Invoicing
1	**Import Currency Conversion Rates**	`CurrencyConversionRate.csv`	X
2	**Import Expense Header Category Mappings**	`ExpenseHeaderCategoryMap.csv`	
3	**Import Expense Auto Types**	`AutoType.csv`	
4	**Import CSV Lookup Tables Used By Approval Rules**	`RuleCSVHelperFiles.zip`	X
5	**Import Per Diem Limits**	`LocationType.csv`	
6	**Import Purchase Order Conditions**	`POConditions.csv`	X
7	**Import Payment Method Types**	`PaymentMethodType.csv`	X
8	**Import Purchasing Units**	`PurchasingUnit.csv`	X
9	**Import Bank Identifier Types**	`BankIDType.csv`	X
10	**Import Bank Account Types**	`BankAccountType.csv`	X
11	**Import Item Categories**	`ItemCategory.csv`	X
12	**Import Payment Terms**	• `PaymentTerms.csv` • `PaymentTermStepDetails.csv` • `PaymentTermSteps.csv`	X
13	**Import Expense Code Mappings**	`ExpenseCodeMapping.csv`	
14	**Import Allowable Hierarchy Choices**	`AllowableHierarchyChoice.csv`	
15	**Import Tax Type Mappings**	`TaxTypeMap.csv`	X
16	**Import Company Codes**	`CompanyCode.csv`	X
17	**Import Enterprise Users**	`SharedUser.csv`	X
18	**Import Bank Account Number Types**	`BankAccountIDType.csv`	X
19	**Import Receiving Types By Commodity Code**	`ReceiptByCommodityCode.csv`	X
20	**Import Internal Order Codes**	`InternalOrder.csv`	X
21	**Import Plants Data**	`Plant.csv`	X
22	**Import Purchase Organizations**	`PurchaseOrg.csv`	X

	Data import task	CSV file(s)	SAP Ariba Buying and Invoicing
23	**Import Assets**	`Asset.csv`	X
24	**Import Cost Centers**	`CostCenter.csv`	X
25	**Import Tax Codes**	`TaxCode.csv`	X
26	**Import ERP Commodity Codes**	`ERPCommodityCode.csv`	X
27	**Import Account Categories**	`AccountCategory.csv`	X
28	**Import Account Types**	`AccountType.csv`	X
29	**Import WBS Elements**	`WBSElement.csv`	X
30	**Import Purchase Groups**	`PurchaseGroup.csv`	X
31	**Import General Ledgers**	`GeneralLedger.csv`	X
32	**Import User/Billing Address Relationships**	`SharedUserBillingAddressMap.csv`	
33	**Import User/Ship To Address Relationships**	`SharedUserShipToAddressMap.csv`	X
34	**Import Supplier Bank Payment Locations**	`SupplierPaymentBankLocation.csv`	X
35	**Import Buyer Bank Payment Locations**	`BuyerPaymentBankLocation.csv`	X
36	**Import Company Code Internal Order Mapping**	`CompanyCodeIOCombo.csv`	X
37	**Import Plant To Purchase Organization Mapping**	`PlantPurchaseOrgCombo.csv`	X
38	**Import Expense Tax Rate Lookup By Tax Code**	`ExpenseTaxRateLookupByTaxCode.csv`	
39	**Import Tax Code Lookups**	`TaxCodeLookup.csv`	X
40	**Import Account Category Field Status Combinations**	`AccCategoryFieldStatusCombo.csv`	X
41	**Import Company Code WBS Mapping**	`CompanyCodeWBSCombo.csv`	X
42	**Import Custom Groups**	`CustomGroup.csv`	X
43	**Import Remittance Locations**	• `RemittanceLocation.csv` • `RemittanceLocationDetails.csv`	X
44	**Import PCards**	`PCard.csv`	X
45	**Import Supplier Organizations**	• `SupplierIDs.csv` • `SupplierOrganizations.csv`	X
46	**Import Tax Rates By Tax Code Lookup**	`TaxRateLookupByTaxCode.csv`	X
47	**Import Expense Types**	• `ExpenseType.csv` • `ExpenseTypeGLMapping.csv`	
48	**Import ERP Commodity to Commodity Mapping**	`ERPCommodityCodeMap.csv`	X

	Data import task	CSV file(s)	SAP Ariba Buying and Invoicing
49	**Import Adjustment Types**	`AdjustmentType.csv`	
50	**Import Group To Child Group Mapping**	• `GroupParentGroup.csv` • `GroupChildGroupMap.csv`	X
51	**Import User Group Mapping**	• `UserParentGroup.csv` • `GroupSharedUserMap.csv`	X
52	**Import Approval Queues**	`ApprovalQueue.csv`	
53	**Import Users**	• `User.csv` • `UserPCardMap.csv`	X
54	**Import Suppliers**	• `Supplier.csv` • `SupplierLocation.csv`	X
55	**Import Expense Tax Code Lookup**	`ExpenseTaxCodeLookup.csv`	
56	**Import Enterprise User/Supervisor Relationships**	`SharedUserSupervisor.csv`	X
57	**Import Responsible User Assignments**	• `User.csv` • `ResponsibleUser.csv`	X
58	**Import User Approval Limits**	`UserApprovalLimit.csv`	X
59	**Import Minority Vendor Data**	`MinorityVendor.csv`	X
60	**Import Supplier Supplement Data**	`SupplierSupplement.csv`	
61	**Import Supplier Location Contacts**	`SupplierLocationSupplement.csv`	X
62	**Import Receiving Types By Part Number**	`ReceiptByPartNumber.csv`	X
63	**Import L2 Charge**	`Charge.csv`	
64	**Import Purchase Organization to Supplier Mapping**	`PurchaseOrgSupplierCombo.csv`	X
65	**Import Supplier Location Remittance Information**	• `SLRemittanceInformation.csv` • `SLRemittanceInformationDetails.csv`	X
66	**Import Scheduled Tasks**	`ScheduledTaskData.zip`	
67	**Import Asset Classes**	`AssetClass.csv`	
68	**Import Asset Line Types**	`AssetType.csv`	
69	**Import Storage Location**	`StorageLocation.csv`	X

Data Import Order for Generic

A table shows the **Import Batch Data** data import sequence for data for Generic.

	Data import task	CSV file(s)	SAP Ariba Buying and Invoicing
1	**Import Currency Conversion Rates**	`CurrencyConversionRate.csv`	X
2	**Import Expense Header Category Mappings**	`ExpenseHeaderCategoryMap.csv`	
3	**Import Expense Auto Types**	`AutoType.csv`	
4	**Import CSV Lookup Tables Used By Approval Rules**	`RuleCSVHelperFiles.zip`	X
5	**Import Per Diem Limits**	`LocationType.csv`	
6	**Import Purchase Order Conditions**	`POConditions.csv`	X
7	**Import Purchasing Units**	`PurchasingUnit.csv`	X
8	**Import Payment Terms**	• `PaymentTerms.csv` • `PaymentTermStepDetails.csv` • `PaymentTermSteps.csv`	X
9	**Import Payment Method Types**	`PaymentMethodType.csv`	X
10	**Import Bank Account Types**	`BankAccountType.csv`	X
11	**Import Account Types**	`AccountType.csv`	X
12	**Import Expense Code Mappings**	`ExpenseCodeMapping.csv`	
13	**Import Allowable Hierarchy Choices**	`AllowableHierarchyChoice.csv`	
14	**Import Tax Type Mappings**	`TaxTypeMap.csv`	X
15	**Import Customization AML**	Customizations.aml	
16	**Import Accounts**	`Account.csv`	X
17	**Import Tax Codes**	`TaxCode.csv`	X
18	**Import Companies**	`Company.csv`	X
19	**Import Addresses**	`Address.csv`	X
20	**Import Projects**	`Project.csv`	X
21	**Import Enterprise Users**	`SharedUser.csv`	X
22	**Import Subaccounts**	`SubAccount.csv`	X
23	**Import Expense Tax Rate Lookup By Tax Code**	`ExpenseTaxRateLookupByTaxCode.csv`	
24	**Import Business Units**	`BusinessUnit.csv`	X
25	**Import Receiving Types By Commodity Code**	`ReceiptByCommodityCode.csv`	X
26	**Import Supplier Bank Payment Locations**	`SupplierPaymentBankLocation.csv`	X
27	**Import Buyer Bank Payment Locations**	`BuyerPaymentBankLocation.csv`	X

	Data import task	CSV file(s)	SAP Ariba Buying and Invoicing
28	**Import Cost Centers**	`CostCenter.csv`	X
29	**Import Tax Code Lookups**	`TaxCodeLookup.csv`	X
30	**Import User/Ship To Address Relationships**	`SharedUserShipToAddressMap.csv`	X
31	**Import Remittance Locations**	• `RemittanceLocation.csv` • `RemittanceLocationDetails.csv`	X
32	**Import PCards**	`PCard.csv`	X
33	**Import Accounting Combinations**	`AccountingCombination.csv`	X
34	**Import Tax Rates By Tax Code Lookup**	`TaxRateLookupByTaxCode.csv`	X
35	**Import Custom Groups**	`CustomGroup.csv`	X
36	**Import Supplier Organizations**	`SupplierOrganization.csv`	X
37	**Import Users**	• `User.csv` • `UserPCardMap.csv`	X
38	**Import ERP Commodity to Commodity Mapping**	`ERPCommodityCodeMap.csv`	X
39	**Import AdjustmentType**	`AdjustmentType.csv`	
40	**Import Group To Child Group Mapping**	• `GroupParentGroup.csv` • `GroupChildGroupMap.csv`	X
41	**Import User Group Mapping**	• `UserParentGroup.csv` • `GroupSharedUserMap.csv`	X
42	**Import Suppliers**	• `Supplier.csv` • `SupplierLocation.csv`	X
43	**Import Responsible User Assignments**	• `User.csv` • `ResponsibleUser.csv`	X
44	**Import Expense Tax Code Lookup**	`ExpenseTaxCodeLookup.csv`	
45	**Import VAT Defaults**	`VATDefaults.csv`	
46	**Import Enterprise User/Supervisor Relationships**	`SharedUserSupervisor.csv`	X
47	**Import Receiving Types By Part Number**	`ReceiptByPartNumber.csv`	X
48	**Import Supplier Supplement Data**	`SupplierSupplement.csv`	
49	**Import L2 Charge**	`Charge.csv`	
50	**Import Supplier Location Contacts**	`SupplierLocationSupplement.csv`	X
51	**Import Supplier Location Remittance Information**	• `SLRemittanceInformation.csv` • `SLRemittanceInformationDetails.csv`	X

	Data import task	CSV file(s)	SAP Ariba Buying and Invoicing
52	**Import Scheduled Tasks**	`ScheduledTaskData.zip`	

Data Import Tasks for SAP Ariba Buying

A table describes the data import tasks that import SAP Ariba Buying data.

Data Import Task	CSV File(s) Imported
Import (Queue) Cancel Purchase Order Status	`PurchaseOrderCancel.csv`
Import Accounting Combination Groups	`AccountingCombinationGroup.csv`
Import Accounting Combinations	`AccountingCombination.csv`
Import Accounts	`Account.csv`
Import Account Types	`AccountType.csv`
Import Account Type Translations	`AccountTypeLanguage.csv`
Import Addresses	`Addresses.csv`
Import Adjustment Types	`AdjustmentType.csv`
Import Adjustment Type Translations	`AdjustmentTypeLanguage.csv`
Import Approvable Types	`ApprovableType.csv`
Import Bank Account Number Types	`BankAccountIDType.csv`
Import Bank Account Number Type Translations	`BankAccountIDType.csv`
Import Bank Account Types	`BankAccountType.csv`
Import Bank Account Type Translations	`BankAccountType.csv`
Import Bank Identifier Types	`BankIdentifierType.csv`
Import Bank Identifier Type Translations	`BankIdentifierType.csv`
Import Batch Data (Batch data import tasks import CSV files from a ZIP file)	
Import Bidding Types	`BiddingType.csv`
Import Bidding Type Translations	`BiddingType.csv`
Import Business Units	`BusinessUnits.csv`
Import Buyer Bank Payment Locations	`BuyerPaymentBankLocation.csv`
Import Cancel Purchase Order Error	`PurchaseOrderCancelErrorImport.csv`
Import Cancel Purchase Order ID	`PurchaseOrderCancelIDImport.csv`
Import Catalog Views	`CustomCatalog.csv`

Data Import Task	CSV File(s) Imported
Import Category Drivers	CategoryDriver.csv
Import Change Purchase Order Error	PurchaseOrderChangeErrorImport.csv
Import Change Purchase Order ID	PurchaseOrderChangeIDImport.csv
Import Change Purchase Order Status	PurchaseOrderChangeErrorImport.csv
Import Charge Exception Types	ChargeExceptionType.csv
Import Charge Exception Type Translations	ChargeExceptionType.csv
Import Charge Exception Type Validation Data	ChargeExceptionTypeValidation.csv
Import Classification Code Relationships	ClassificationCodeMap.csv
Import Collaboration Reasons	CollaborationReasons.csv
Import Collaboration Reason Translations	CollaborationReasons.csv
Import Collaboration Types	CollaborationType.csv
Import Collaboration Type Translations	CollaborationType.csv
Import Commodity Code Hierarchy	CommodityCode.csv
Import Commodity Codes	CommodityCode.csv
Import Contractor Expense Types	ContractorExpenseType.csv
Import Contractor Expense Type Translations	ContractorExpenseType.csv
Import Contractor Performance Ten Point Values	ContractorPerformanceTenPointValue.csv
Import Contractor Performance Ten Point Value Translations	ContractorPerformanceTenPointValue.csv
Import Contractor Performance Three Point Values	ContractorPerformanceThreePointValue.csv
Import Contractor Performance Three Point Value Translations	ContractorPerformanceThreePointValue.csv
Import CSV Budget in 4 file format	BudgetPeriods.csv BudgetLineItems.csv BudgetAllocation.csv BudgetHeader.csv
Import CSV Lookup Tables Used By Approval Rules	RuleCSVHelperFiles.zip
Import Currency Conversion Rates	CurrencyConversionRate.csv
Import Currency Information	CurrencyInfo.csv
Import Currency Mappings	CurrencyMap.csv
Import Custom Groups	CustomGroup.csv
Import Departments	Department.csv
Import Enterprise User/Supervisor Relationships from PeopleSoft HRMS	SharedUserSupervisorPShr.csv
Import Enterprise User/Supervisor Relationships	SharedUserSupervisorPSCSV.csv
Import Enterprise Users With Alternate Email Addresses	SharedUserAlternateEmail.csv

Data Import Task	CSV File(s) Imported
Import Equivalent Words (For SAP Ariba internal use only)	`EquivalentWords.csv`
Import ERP Commodity Codes	`ERPCommodityCode.csv`
Import ERP Commodity to Commodity Mapping	`ERPCommodityCodeMap.csv`
Import Escalation By Approvable Type	`EscalationsByApprovableType.csv`
Import GL Business Units	`GLBusinessUnit.csv`
Import Group Management Information	`ApprovalQueue.csv`
Import Group To Child Group Mapping	`GroupChildGroupMap.csv`
Import Initial Data	
Import Labor Pay Codes	`LaborPayCode.csv`
Import Labor Pay Code Translations	`LaborPayCode.csv`
Import Level2 Charge	`Level2Charge.csv`

Import Payment Terms	`PaymentTermStepDetails.csv`
	`PaymentTermSteps.csv`
	`PaymentTerms.csv`

Import Payment Term Translations	`PaymentTermStepDetails.csv`
	`PaymentTermSteps.csv`
	`PaymentTerms.csv`

Import PCards	`PCard.csv`
Import Products	`Product.csv`
Import Projects	`Project.csv`
Import Punchin Groups	`PunchinGroup.csv`
Import Purchase Order Conditions	`POConditions.csv`
Import Purchase Order ERP Header Status	`PurchaseOrderERPHeaderStatusImport.csv`

Import Purchase Order Status	`PurchaseOrderERPHeaderStatusImport.csv`
	`PurchaseOrderErrorImport.csv`
	`PurchaseOrderIDImport.csv`

Import Purchasing Units	`PurchasingUnit.csv`
Import Receipts	`ReceiptImport.csv`

Import Receipt Status	`ReceiptErrorImport.csv`
	`Receipt IDImport`
	`ReceiptErrorImport.csv`

Import Receiving Types By Commodity Code	`ReceiptByCommodityCode.csv`
Import Receiving Types By Part Number	`ReceiptByPartNumber.csv`
Import Relation Entries	`RelationEntry.csv`
Import Relations	`RelationType.csv`

Data Import Task	CSV File(s) Imported

Import Requisitions	`RequisitionSplitAccounting.csv`
	`RequisitionDetail.csv`
	`RequisitionHeader.csv`

Import Responsible User Assignments	`ResponsibleUser.csv`
Import Scheduled Tasks	`ScheduledTaskData.zip`
Import SetIDs	`SetId.csv`
Import Statistics	`Statistics.csv`
Import Supplier Location Contacts	`SupplierLocation.csv`
Import Supplier Organizations	`SupplierOrganizations.csv`
Import Supplier Related Data	
Import Suppliers	`SupplierIDs.csv`
Import Supplier Supplement Data	`SupplierSupplement.csv`
Import TransformUsing Catalog Map (For SAP Ariba internal use only)	`TransformMap.csv`
Import Unit Of Measure CIF Mapping	`UnitsOfMeasureCIFMap.csv`
Import Unit Of Measure Mapping	`UnitsOfMeasureMap.csv`
Import Units Of Measure	`UnitsOfMeasure.csv`
Import User/Ship To Address Relationships	`SharedUserShipToAddressMapPSCSV.csv`
Import User Approval Limits	`UserApprovalLimitCSV.csv` `UserApprovalLimit.csv`
Import User Group Mapping	`GroupSharedUserMapUpdate.csv`
Import User Group Translations	`GroupSharedUserMapUpdate.csv`
Import Value Qualifiers	`ValueQualifier.csv`
Import Value Qualifier Translations	`ValueQualifier.csv`
SharedUserPSCSVPullDisplayName	`SharedUserPSCSV.csv`
Import Incoterms	`IncoTerms.csv`
Import Incoterms Translations	`IncoTerms.csv`

Data Import Order for SAP Ariba Buying

Tables for PeopleSoft, SAP, and Generic list the order in which the **Import Batch Data** data import task imports data for SAP Ariba Buying. You can use this information as a guide when running individual data import tasks.

Related Information

PeopleSoft Data Import Order for SAP Ariba Buying [page 343]
SAP Data Import Order for SAP Ariba Buying [page 345]
Generic Data Import Order for SAP Ariba Buying [page 346]

PeopleSoft Data Import Order for SAP Ariba Buying

A table shows the SAP Ariba Buying **Import Batch Data** data import sequence for data when integrated with PeopleSoft ERP.

	Data Import Task	CSV File(s)
1	**Import Currency Conversion Rates**	`CurrencyConversionRate.csv`
2	**Import ERP Commodity Codes**	`ERPCommodityCode.csv`
3	**Import CSV Lookup Tables Used By Approval Rules**	`RuleCSVHelperFiles.zip`
4	**Import Account Types**	`AccountType.csv`

5	**Import Payment Terms**	`PaymentTerms.csv`
		`PaymentTermStepDetails.csv`
		`PaymentTermSteps.csv`

6	**Import Accounting Combination Groups**	`AccountingCombinationGroup.csv`
7	**Import SetIDs**	`SetID.csv`
8	**SharedUserPSCSVPullDisplayName**	`SharedUserPSCSV.csv`
9	**Import Bank Account Number Types**	`BankAccountIDType.csv`
10	**Import Receiving Types By Commodity Code**	`ReceiptByCommodityCode.csv`
11	**Import Accounts**	`Account.csv`
12	**Import Products**	`Product.csv`
13	**Import Tax Codes**	`TaxCode.csv`
14	**Import ERP Commodity Codes From PeopleSoft**	`ERPCommodityCode.csv`
15	**Import GL Business Units**	`GLBusinessUnit.csv`
16	**Import Addresses From PeopleSoft**	`Address.csv`

	Data Import Task	CSV File(s)
17	**Import Departments**	`Department.csv`
18	**Import Projects**	`Project.csv`
19	**Import Business Units**	`BusinessUnit.csv`
20	**Import PCards**	`PCard.csv`
21	**Import User/Ship To Address Relationships**	`SharedUserShipToAddressMap.csv`
22	**Import Users Consolidated CSV File**	`User.csv` `UserPCardMap.csv`
23	**Import ERP Commodity to Commodity Mapping**	`ERPCommodityCodeMap.csv`
24	**Import Accounting Combinations**	`AccountingCombination.csv`
25	**Import Adjustment Types**	`AdjustmentType.csv`
26	**Import Custom Groups**	`CustomGroup.csv`
27	**Import Supplier Organizations**	`SupplierIDs.csv` `SupplierOrganizations.csv`
28	**Import User Approval Limits**	`UserApprovalLimit.csv`
29	**Import Responsible User Assignments**	`ResponsibleUser.csv`
30	**Import Group To Child Group Mapping**	`GroupParentGroup.csv` `GroupChildGroupMap.csv`
31	**Import User Group Mapping**	`UserParentGroup.csv` `GroupSharedUserMap.csv`
32	**Import Suppliers**	`Supplier.csv` `SupplierLocation.csv`
33	**Import Receiving Types By Part Number**	`ReceiptByPartNumber.csv`
34	**Import Supplier Supplement Data**	`SupplierSupplement.csv`
35	**Import Supplier Location Contacts**	`SupplierLocation.csv`
36	**Import Enterprise User/Supervisor Relationships from PeopleSoft HRMS**	`SharedUserSupervisorPShr.csv`
37	**Import Scheduled Tasks**	`ScheduledTaskData.zip`
38	**Import Incoterms**	`IncoTerms.csv`

SAP Data Import Order for SAP Ariba Buying

A table shows the SAP Ariba Buying `Import Batch Data` data import sequence for data when integrated with SAP ERP.

	Data Import Task	CSV File(s)
1	**Import Currency Conversion Rates**	`CurrencyConversionRate.csv`
2	**Import Purchase Order Conditions**	`POConditions.csv`
3	**Import Purchasing Units**	`PurchasingUnit.csv`
4	**Import Bank Identifier Types**	`BankIDType.csv`
5	**Import Bank Account Types**	`BankAccountType.csv`
6	**Import Item Categories**	`ItemCategory.csv`

7	**Import Payment Terms**	`PaymentTerms.csv`
		`PaymentTermStepDetails.csv`
		`PaymentTermSteps.csv`

8	**Import Company Codes**	`CompanyCode.csv`
9	**Import Bank Account Number Types**	`BankAccountIDType.csv`
10	**Import Internal Order Codes**	`InternalOrder.csv`
11	**Import Plants Data**	`Plant.csv`
12	**Import Purchase Organizations**	`PurchaseOrg.csv`
13	**Import Assets**	`Asset.csv`
14	**Import Cost Centers**	`CostCenter.csv`
15	**Import Tax Codes**	`TaxCode.csv`
16	**Import ERP Commodity Codes**	`ERPCommodityCode.csv`
17	**Import Account Categories**	`AccountCategory.csv`
18	**Import Account Types**	`AccountType.csv`
19	**Import WBS Elements**	`WBSElement.csv`
20	**Import Purchase Groups**	`PurchaseGroup.csv`
21	**Import General Ledgers**	`GeneralLedger.csv`
22	**Import User/Billing Address Relationships**	`SharedUserBillingAddressMap.csv`
23	**Import User/Ship To Address Relationships**	`SharedUserShipToAddressMap.csv`
24	**Import Buyer Bank Payment Locations**	`BuyerPaymentBankLocation.csv`
25	**Import Company Code Internal Order Mapping**	`CompanyCodeIOCombo.csv`
26	**Import Plant to Purchase Organization Mapping**	`PlantPurchaseOrgCombo.csv`
27	**Import Account Category Field Status Combinations**	`AccCategoryFieldStatusCombo.csv`
28	**Import Company Code WBS Mapping**	`CompanyCodeWBSCombo.csv`

	Data Import Task	CSV File(s)
29	**Import Custom Groups**	`CustomGroup.csv`
30	**Import PCards**	`PCard.csv`
31	**Import Supplier Organizations**	`SupplierIDs.csv` `SupplierOrganizations.csv`
32	**Import ERP Commodity to Commodity Mapping**	`ERPCommodityCodeMap.csv`
33	**Import Adjustment Types**	`AdjustmentType.csv`
34	**Import Group To Child Group Mapping**	`GroupParentGroup.csv` `GroupChildGroupMap.csv`
35	**Import User Group Mapping**	`UserParentGroup.csv`
36	**Import Approval Queues**	`ApprovalQueue.csv`
37	**Import Users**	`User.csv`
38	**Import Suppliers**	`Supplier.csv` `SupplierLocation.csv`
39	**Import Enterprise User/Supervisor Relationships**	`SharedUserSupervisor.csv`
40	**Import Responsible User Assignments**	`ResponsibleUser.csv`
41	**Import User Approval Limits**	`ApprovalQueue.csv`
42	**Import Minority Vendor Data**	`MinorityVendor.csv`
43	**Import Supplier Supplement Data**	`SupplierSupplement.csv`
44	**Import Supplier Location Contacts**	`SupplierLocation.csv`
45	**Import Purchase Organization to Supplier Mapping**	`PurchaseOrgSupplierCombo.csv`
46	**Import Incoterms**	`IncoTerms.csv`
47	**Import Asset Classes**	`AssetClass.csv`
48	**Import Asset Line Types**	`AssetType.csv`

Generic Data Import Order for SAP Ariba Buying

A table shows the SAP Ariba Buying `Import Batch Data` data import sequence for data when integrated with Generic ERP.

	Data Import Task	CSV File(s)
1	**Import Currency Conversion Rates**	`CurrencyConversionRate.csv`
2	**Import Purchase Order Conditions**	`POConditions.csv`
3	**Import Purchasing Units**	`PurchasingUnit.csv`

	Data Import Task	CSV File(s)

4	**Import Payment Terms**	`PaymentTerms.csv`
		`PaymentTermStepDetails.csv`
		`PaymentTermSteps.csv`

5	**Import Bank Account Types**	`BankAccountType.csv`
6	**Import Account Types**	`AccountType.csv`
7	**Import Accounts**	`Account.csv`
8	**Import Tax Codes**	`TaxCode.csv`
9	**Import Addresses**	`Address.csv`
10	**Import Companies**	`Company.csv`
11	**Import Projects**	`Project.csv`
12	**Import Subaccounts**	`SubAccount.csv`
13	**Import Business Units**	`BusinessUnit.csv`
14	**Import Buyer Bank Payment Locations**	`BuyerPaymentBankLocation.csv`
15	**Import Cost Centers**	`CostCenter.csv`
16	**Import User/Billing Address Relationships**	`SharedUserBillingAddressMap.csv`
17	**Import User/Ship To Address Relationships**	`SharedUserShipToAddressMap.csv`
18	**Import PCards**	`PCard.csv` UserPCardMap.csv
19	**Import Accounting Combinations**	`AccountingCombination.csv`
20	**Import Custom Groups**	`CustomGroup.csv`
21	**Import Supplier Organizations**	`SupplierIDs.csv` `SupplierOrganizations.csv`
22	**Import Users**	`User.csv`
23	**Import ERP Commodity to Commodity Mapping**	`ERPCommodityCodeMap.csv`
24	**Import Adjustment Types**	`AdjustmentType.csv`
25	**Import Group To Child Group Mapping**	`GroupParentGroup.csv` `GroupChildGroupMap.csv`
26	**Import User Group Mapping**	`UserParentGroup.csv`
27	**Import Approval Queues**	`ApprovalQueue.csv`

28	**Import Suppliers**	`Supplier.csv`
		`SupplierIDs.csv`
		`SupplierLocation.csv`

29	**Import Responsible User Assignments**	`ResponsibleUser.csv`
30	**Import Enterprise User/Supervisor Relationships**	`SharedUserSupervisor.csv`
31	**Import Supplier Supplement Data**	`SupplierSupplement.csv`

	Data Import Task	CSV File(s)
32	**Import Supplier Location Contacts**	`SupplierLocation.csv`
33	**Import Incoterms**	`IncoTerms.csv`

CSV Files Reference

Procurement data used in SAP Ariba Buying and Invoicing and SAP Ariba Buying solutions use CSV files for import and export. A CSV file is required if the data it contains is required for an SAP Ariba solution to function properly. Unless otherwise noted, required files should not be empty.

The following table lists each CSV file, describes it, and indicates whether it is required by SAP Ariba Buying solutions and SAP Ariba Buying and Invoicing:

File	Description	SAP Ariba Buying solutions	SAP Ariba Buying and Invoicing
`Charge.csv`	Contains charge statement information in CSV format. When your site uses purchasing cards, your bank sends regular statements, showing the charges made on those cards. Banks provide Level 2 (header level) charge in CSV format.	`Charge.csv` is required only if you import Level 2 charges	
`ChargeExceptionType.csv`	Defines charge exception types. A charge exception represents a single discrepancy between a bank's charge data and the data in the buying solution.	No	
`ChargeExceptionTypeValidation.csv`	Defines validation data for the charge exception types defined in `ChargeExceptionType.csv`. Charge exception type validation data defines exactly which fields to compare and what tolerances to allow for each type of charge exception.	No	
`ChargeTaxCodes.csv`	Maps the common tax names found on charge statements to procurement line types.	`ChargeTaxCodes.csv` is required only if your site receives charges from SAP Business Network.	
`CommodityApprovers.csv`	An approver lookup table that specifies approvers (by group) based on commodity codes. This file is ready by the default requisition approval rule called Commodity Approvers.	`CommodityApprovers.csv`, `CommodityCentralReceivers.csv`, and `CommodityReceivers.csv` are used by the default approval rules Commodity Approvers, Commodity Central Receiving, and Commodity Receiving, and are required when these approval rules are used.	No
`CommodityCentralReceivers.csv`	An approver lookup table that specifies central receiving approvers (by group) based on commodity codes. This file is read by the default receipt approval rule called CommodityCentralReceiving.		No

File	Description	SAP Ariba Buying solutions	SAP Ariba Buying and Invoicing
`CommodityReceivers.csv`	An approver lookup table that specifies receipt approvers (by group) based on commodity codes. This file is read by the default receipt approval rule called Commodity Receivers.	These approval rules are used in the default configuration. Receiving is also enabled in the default configuration. To modify the default approval rules, see the *Approval process management guide*. To disable receiving, you must contact SAP Ariba Customer Support.	No
`CustomCatalog.csv`	For SAP Ariba internal use only.		
`CustomGroup.csv`	Defines custom groups.	No	
`EquivalentWords.csv`	For SAP Ariba internal use only.		
`ExpenseCodeMapping.csv`	Expense card charge statements include an expense code for each charge to indicate how that charge must be classified. This file maps expense codes to expense types.	`ExpenseCodeMapping.csv` is required only if users use expense cards.	
`IncoTerms.csv`	Defines the Incoterms code and the location applicable for the incoterms specified for your site.	Yes	Yes
`PCard.csv`	Defines your company's list of valid purchasing cards. A purchasing card is a corporate credit card.	`PCard.csv` is required only if your users use purchasing cards.	
`POConditions.csv`	Contains terms and conditions text for purchase orders.	No	
`ReceiptByCommmodityCode.csv`	Associates receiving types with commodity codes. Receiving types determine whether a receipt is required for an item on a purchase order or contract.	No	No
`ReceiptByPartNumber.csv`	Associates receiving types with part numbers. Receiving types determine whether a receipt is required for an item on a purchase order or contract.	No	No
`Remittance.csv`	Contains general remittance transaction information. Remittance transaction information is a summary statement that provides details about a payment transaction that has been made. For example, a remittance statement might list all invoices included in a payment, the payment method used, bank information, discount information, and amount paid.	No	No

File	Description	SAP Ariba Buying solutions	SAP Ariba Buying and Invoicing
`RemittanceDetails.csv`	Contains details of the payment documents listed in `Remittance.csv`.	No	No
`RequisitionDetail.csv`	Contains line item detail information for the requisitions in `RequistionHeader.csv`.	No	
`RequisitionHeader.csv`	Contains header information for requisitions retrieved from your ERP system.	No	
`RequisitionLimit.csv`	An approver lookup table that associates groups and requisition limits with either purchasing units or two ERP-specific "rule units." This file is read by the default requisition approval rule called Requisition Cap Rule.	No	
`RequisitionSplitAccounting.csv`	Contains line item split accounting information for the requisitions in `RequisitionHeader.csv`.	No	
`ResponsibleManagers.csv`	An approver lookup table that specifies responsible managers (by user name) for departments (PeopleSoft) or cost centers (SAP or Generic). This file is read by the default invoice reconciliation approval rule called Department Head and the default requisition approval rule called First_DepartmentManagerNotRequester.	Yes	
`StorageLocation.csv`	Contains data related to inventory storage locations at different plants within a buyer organization. Includes storage location details such as a unique identifier, name, address, and the corresponding plant name.		
`UserPCardMap.csv`	Contains user-to-purchasing card mappings. If you do not want to assign purchasing cards to users, create this file with column headings, but without data.	Yes	

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.
About the icons:

- Links with the icon : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
  - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
  - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon : You are leaving the documentation for that particular SAP product or service and are entering an SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.
The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Bias-Free Language

SAP supports a culture of diversity and inclusion. Whenever possible, we use unbiased language in our documentation to refer to people of all cultures, ethnicities, genders, and abilities.

www.ariba.com

Copyright © 2026 Ariba, Inc. All rights reserved.

This documentation, as well as the Ariba solutions, software and/or services described in it, contain proprietary information. They are provided under a license or other agreement containing restrictions on use and disclosure and are also protected by copyright, patent and/or other intellectual property laws. Except as permitted by such agreement, no part of the document may be reproduced or transmitted in any form by any means, electronic, mechanical or otherwise, without the prior written permission of Ariba, Inc.

Ariba, Inc. assumes no responsibility or liability for any errors or inaccuracies that may appear in the documentation. The information contained in the documentation is subject to change without notice.

Ariba and Ariba products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Ariba, Inc. in the United States and other countries. Please see http://www.ariba.com/legal/trademarks for additional trademark information and notices.

Ariba Sourcing solutions (On Demand and software) are protected by one or more of the following patents, including without limitation: U.S. Patent Nos. 6,199,050; 6,216,114; 6,223,167; 6,230,146; 6,230,147; 6,285,989; 6,408,283; 6,499,018; 6,564,192; 6,871,191; 6,952,682; 7,010,511; 7,072,061; 7,130,815; 7,146,331; 7,152,043;7,225,152; 7,277,878; 7,249,085; 7,283,979; 7,283,980; 7,296,001; 7,346,574; 7,383,206; 7,395,238; 7,401,035; 7,407,035; 7,444,299; 7,483,852; 7,499,876; 7,536,362; 7,558,746; 7,558,752; 7,571,137; 7,599,878; 7,634,439; 7,657,461; 7,693,747; 8,364,577; and 8,392,317. Patents pending.

Other Ariba product solutions are protected by one or more of the following patents:

U.S. Patent Nos. 6,199,050, 6,216,114, 6,223,167, 6,230,146, 6,230,147, 6,285,989, 6,408,283, 6,499,018, 6,564,192, 6,584,451, 6,606,603, 6,714,939, 6,871,191, 6,952,682, 7,010,511, 7,047,318, 7,072,061, 7,084,998; 7,117,165; 7,225,145; 7,324,936; 7,536,362; 8,364,577; and 8,392,317. Patents pending.

Certain Ariba products may include third party software or other intellectual property licensed from a third party. For information regarding software or other intellectual property licensed from a third party, go to http://www.ariba.com/copyrights.cfm.