Galvanize

SAP ERP
Data Integration Robot
Implementation Guide

Version 1.0

Document Change History

Date	Version	Description	Author
August 6, 2020	1.0	Initial documentation	Content and Configuration, Galvanize

Contents

1 Overview and Context

This guide describes how to implement the SAP ERP Data Integration Robot. It assumes that you are familiar with ACL Robotics technologies and terminology.

Data integration robots are designed to support pre-built analysis robots currently in development, for release later in 2020. They also serve as a starting point for building custom analysis robots.

The interaction between data integration robots and analysis robots is shown below. The process is to implement the data integration robot and share the resulting tables so they can be used by any analysis robot. The target analysis robot links to the shared tables required to perform the analysis. Resulting exceptions can be exported to HighBond (Results) from within the analysis robot. Data integration robots do not export raw source data to HighBond (Results).
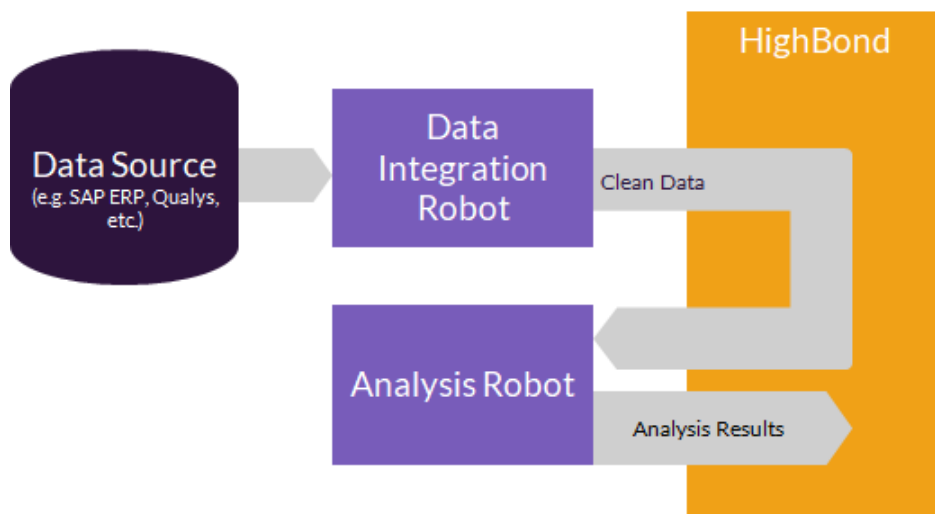
2 Before you Start

Please review the pre-requisites below to confirm your readiness and collect all the appropriate information for the configuration process.

2.1 System Requirements Checklist for Data Integration Robots

Item	Notes	Status
On-premise robot agent 14.2 or higher installed on the server	• Confirm Unicode or non-Unicode version of the agent	
ACL Analytics Version 14.2 or higher installed*	• * A local installation of ACL Analytics can be useful for troubleshooting purposes or for developing custom scripts. If running exclusively as a robot, a local installation may not be required. • Ensure same encoding as the robot agent (Unicode, non-Unicode) • Data integration robots may utilize functionality not available in earlier versions	
Data integration robot is available in the customer organization in HighBond.	• The robot has been successfully deployed into the customer organization via toolkits.highbond.com	

2.2 Source System Access Checklist for Data Integration Robots

Item	Notes	Status
Connection information for SAP ERP is available	• IP address or name of the SAP server • Instance number • Client number	
User credentials are available	• Username • Password • The user account requires Read-only access to the tables to be imported (see available tables below)	
A connection to SAP ERP can be established successfully	• Ensure firewalls/drivers have been installed correctly to connect to SAP ERP from both the development environment (ACL Analytics) and from the Robots Agent	
Error: "You do not have the required entitlement"	• If the robot task fails with this error despite the organization being properly authorized, contact Support.	

3 Implementation Steps

Pre-built robots are delivered into the customer's HighBond organization via toolkits.highbond.com. The process creates the robot in development mode, and provisions the required Default Import Configuration File in the Input / Output tab. To complete the configuration of the robot, follow the steps below.

3.1   Open the Robot in HighBond

1) Log into HighBond Launchpad (www.highbond.com) and browse to the data integration robot.
2) If the robot opens in production mode, toggle the switch to Development mode, and open the Input / Output tab.

3.2   Review the Default Configurations

The robot is delivered with default configurations - stored in the Default Import Configuration File - that are intended to apply to most customers. These configurations include the list of import scripts to run, or filters applicable to certain tables such as document types or date ranges. A customer may require changes to these default settings if they are not applicable to their environment.

The provided default configurations can be reviewed in two locations:

1) Refer to the Reference section of this Implementation Guide, or
2) Download a copy of the Default Import Configuration File from the Input / Output tab.

Do not make any changes to the Default Import Configuration File or overwrite the provisioned file. Do not add, delete, or modify the content or names of any worksheets. Doing so may cause unexpected behavior.

When reviewing the default configurations, take note of any items that require modification, including

1) Import scripts that need to be disabled.
   a) Examples are tables that are not being used by any pre-built or custom analysis robots for the time being, tables to which the user has not been given access, or tables which require customization to align with the customer's needs (refer to section Custom Scripts).
2) Parameter input that requires different values than those provided with the default configuration.

3.3 Configure the User Import Configuration File (optional)

The User Import Configuration File template is available from an implementation consultant. It is not provisioned by toolkits.highbond.com.

Configuration of the User Import Configuration File is only necessary if changes to the default settings are required. If the default settings are acceptable and no changes are required, leave the User Import Configuration File as-is, with its original worksheet names and field names intact.

1) To make changes, follow these steps as applicable:

    a) In the User_Config_Imports worksheet, disable specific import scripts from running, as follows:

        i) Enter the affected import script name(s) in column Import_Name exactly as they appear in the Default Import Configuration File.
        ii) Enter one script per row and mark column Disable_Import with X (required).

    b) In the User_Config_Params worksheet, apply customized parameter values as follows:

        i) To ensure correct configuration of the new parameter value, it is recommended to download a copy of the Default Import Configuration File and copy the affected parameters to the User Import Configuration File to act as a template for the new input.
        ii) Adjust the copied parameter values as needed, paying attention to the correct formatting as described in the Parameter Description field and in the documentation below. Do not change the variable names.

2) Save any changes made to the User Import Configuration File, if applicable.

Note: Default parameters not overwritten via the User Import Configuration File continue to apply.

3.4 Upload required files to the robot's Input / Output tab

1) Upload the User Import Configuration File to the Input / Output tab, whether it has been changed or not.
2) If used, upload any custom scripts or external source files to the Input / Output tab. For more information, refer to section Custom Scripts.

3.5 Set up the robot task

1) In Robots, switch to the Tasks tab, and click Create task. Configure the task as follows:

   a) Assign a meaningful name to the task.
   b) In the next screen, set the task input parameters.

      i) As a first test run, it may be helpful to select the Sample Data mode parameter and familiarize a new user with the various concepts of the robot and its output, using small sets of fictitious data. For more information on this option, refer to section Sample Data Mode.

      ii) Rerun the task on actual data to confirm that a connection to the data connector can be established.
         (1) If the task was initially created in Sample Data mode, edit the task to set the Sample Data mode parameter to False, or leave blank.
         (2) To test the connection quickly, set the following parameter to import only the first 500 records to True. This option is available only in data integration robots that connect to data connectors. Data integration robots that import external source files do not have this option.
         (3) Populate the task parameters with the user's credentials and other connectivity settings as prompted.
         (4) Run the task.

      iii) Once a successful connection has been confirmed, edit the task to the turn the 500-record limitation off to prevent future imports from receiving limited data.

2) After confirming that a given data integration task runs successfully, share all source tables (S_*) and the robot's Import Error Log table via the Input / Output tab so that they can later be linked by analysis robots.

3.6 Activate the robot in Production Mode

1) After confirming the successful implementation in Development mode, activate the robot in Production mode.

   a) All external files in the Input / Output tab need to be manually uploaded to Production mode. This includes:

      i) Default Import Configuration File (download from Development mode, do not make changes).
      ii) User Import Configuration File (download from Development mode).
      iii) Any custom scripts or external source files, if applicable.

   b) Once the robot has run in Production mode, the resulting source tables need to be manually shared in the Input / Output tab for use by analysis robots also running in Production mode.

4 Sample Data Mode

Sample Data mode can be used temporarily to create source tables populated with fictitious data. The sample data resides in the Default Import Configuration File. This mode provides a method to showcase a robot's components and concepts without the need to connect to an actual data source.

Most task parameters such as user account information can be skipped in Sample Data mode, except for any hashed input such as passwords. Enter a random string of text to be able to save the task. An actual password is not required in Sample Data mode. Parameter input from the Default and User Import Configuration Files is not used in this mode. The 500-record limitation, if selected, does not apply.

5 Custom scripts

Data integration robots import a variety of tables to support analysis robots. A customer may need to import additional tables to support their own custom analytics, or they may need to import an existing table with modifications such as additional filters or fields. Both examples require the use of custom scripts.

The original data integration robot **should not** be modified internally to incorporate new scripts or modify existing scripts. Doing so jeopardizes the future ability to deploy official updates seamlessly into the customer's organization. Custom scripts provide a mechanism for these types of changes, as described in the following scenarios.

5.1 Importing an additional table

1) Create a separate custom script for every additional table.

2) To ensure compatibility with standard data integration and analysis robots processes, source tables must be named as follows, using SAP ERP as an example. Adapt the naming conventions to the data integration robot being built:

    a. Minimum table name: S_<application_system>_<table_name>. For example: S_SAP_LFA1
    b. If there are filters being applied, the table name can be expanded to include a reference to the filter. For example: S_SAP_LFA1_LAND1 if filtered on LAND1.
    c. The table name must not match an existing table name (see next scenario).
    d. The script must be saved as an external aclscript file, with the same name as the table. For example: S_SAP_LFA1_LAND1.aclscript.
    e. The custom script must be uploaded to the Input / Output tab in both Development mode and Production mode.

5.2 Customizing an existing table import

1) If a custom script replaces an original script, disable the original script via the User Import Configuration File.

2) Create a separate custom script for every customized table import.

3) To prevent conflicts with the original table import, the custom table name must be unique.
    a. For example, if the original table S_SAP_LFA1 is to be replaced by a custom script, the new table name could be S_SAP_LFA1_Custom.
    b. The script must be saved as an external aclscript file, with the same name as the table. For example: S_SAP_LFA1_Custom.aclscript.
    c. The custom script must be uploaded to the Input / Output tab in both Development mode and Production mode.

Existing variables from the Default Import Configuration File or User Import Configuration File are available to the custom script. The 500-record limitation is also available to the custom scripts. Copy the corresponding syntax from an original script.

Custom scripts can be developed in an ACL project to facilitate testing or as an external text file. An existing script can be used as a template.

Custom scripts run last, in alphabetical order on script name. They can use the tables created by the enabled original import scripts as input if needed. If a custom script is dependent on another custom script, control the order in which they run via the script names.

6 Reference

6.1 Default Parameters

The Default Import Configuration File provides the following parameter input for the import scripts:

Variable Name	Worksheet Name	Type	Applies to	Default Value
v_coa_skat	Default_Config_Params	CHAR	SKAT_KTOPL	Defined as an empty string "". If no user-specified value is found, imports all chart of account entries
v_doc_types_bsis_bsas	Default_Config_Params	CHAR	BSIS_BLART BSAS_BLART	"'AB','AE','SA','JE'"
v_include_vgabe_ekbe	Default_Config_Params	CHAR	EKBE_VGABE	"'1','2'"
v_language_code	Default_Config_Params	CHAR	SPRAS or SPRSL in multiple tables	"E"
v_start_date_bsis_bsas v_end_date_bsis_bsas v_start_date_cdhdr_cdpos_bkg v_end_date_cdhdr_cdpos_bkg v_start_date_cdhdr_cdpos_po v_end_date_cdhdr_cdpos_po v_start_date_eban v_end_date_eban v_start_date_ekbe v_end_date_ekbe v_start_date_ekbn v_end_date_ekbn v_start_date_ekko_ekpo v_end_date_ekko_ekpo	Default_Config_Params	CHAR	BSIS/BSAS: BUDAT CDHDR (banking, po): UDATE EBAN: BADAT EKBE: CPUDT EKBN: ERDAT EKKO/EKPO: AEDAT RBKP: BUDAT	Calculates start and end dates as the first and the last day of the previous month.

Variable Name	Worksheet Name	Type	Applies to	Default Value
v_start_date_rbkp_rseg v_end_date_rbkp_rseg				
v_tcode_list_cdhdr_po	Default_Config_Params	CHAR	CDHDR_TCODE	"'ME28','ME29','E29N'" The table is used in the context of released purchase orders.

6.2 Overwriting default parameters

If any of the provided default parameters do not apply or are incomplete, declare the required custom values in the User Import Configuration File. The format and naming conventions must match the Default Import Configuration File.

Variable Name	Worksheet Name	Type	Applies to	Default Value
v_coa_skat	User_Config_Params	CHAR	SKAT_KTOPL	Insert a single value for filtering field KTOPL, enclosed in double quotes and single quotes, for example "'9999'".
v_doc_types_bsis_bsas	User_Config_Params	CHAR	BSIS_BLART BSAS_BLART	Insert one or more document types, separated by comma. Enclose each individual value in single quotes, and the entire string in double quotes (including when only one value is listed).
v_include_vgabe_ekbe	User_Config_Params	CHAR	EKBE_VGABE	Transaction / Event Type (Purchase Order History). Insert one or more codes, separated by comma. Enclose each individual value in single quotes, and the entire string in double quotes (including when only one value is listed).

Variable Name	Worksheet Name	Type	Applies to	Default Value
v_language_code	User_Config_Params	CHAR	SPRAS or SPRSL in multiple tables	Insert the required language key, enclosed in double quotes. All tables use the same language key.

Variable Name	Worksheet Name	Type	Applies to	Default Value
v_start_date_bsis_bsas v_end_date_bsis_bsas v_start_date_cdhdr_cdpos_bkg v_end_date_cdhdr_cdpos_bkg v_start_date_cdhdr_cdpos_po v_end_date_cdhdr_cdpos_po v_start_date_eban v_end_date_eban v_start_date_ekbe v_end_date_ekbe v_start_date_ekbn v_end_date_ekbn v_start_date_ekko_ekpo v_end_date_ekko_ekpo v_start_date_rbkp_rseg v_end_date_rbkp_rseg	User_Config_Params	CHAR	BSIS/BSAS: BUDAT CDHDR (banking, po): UDATE EBAN: BADAT EKBE: CPUDT EKBN: ERDAT EKKO/EKPO: AEDAT RBKP: BUDAT	Populate each required variable with a start or end date respectively, written in the format "YYYY-MM-DD" as a character string. Enclose the dates in double-quotes. Alternatively, enter valid ACL functions to calculate the start and end dates, such as DATE(TODAY() 'YYYY-MM-DD') or DATE(TODAY() - 365 'YYYY-MM-DD') (no quotes).

Variable Name	Worksheet Name	Type	Applies to	Default Value
v_tcode_list_cdhdr_po	User_Config_Params	CHAR	CDHDR_TCODE	Insert one or more TCODEs for filtering table CDHDR, separated by comma. The imported table is intended to be used in the context of released purchase orders. Enclose each individual value in single quotes, and the entire string in double quotes (including when only one value is listed).

Note: All parameter values are ignored if the sample data approach has been selected.

6.3 S_SAP_ADR6_USR21

General approach
- Imports a selection of fields from the joined USR21 (User Name/Address Key Assignment) and ADR6 (Email Addresses) tables in SAP ERP, sufficient to support all core analytics.
- The tables are joined on field PERSNUMBER.
- Depending on the data volume, this import script can take several minutes to run.

Import Filters
Not applicable

Post Processing Script
The post processing scripts contain code to convert a list of fields from character to numeric or date data type (especially applicable for sample data imports). The scripts run against all imported tables and convert these fields if they are found in the current table.

Error Logging
Not applicable.

6.4   S_SAP_BKPF_BSAS

General approach
Imports a selection of fields that may be most useful in supporting custom analytics.

Import Filters
- Applies a document type filter on field BKPF_BLART.
    - o Variable:
        - ▪ v_doc_types_bsis_bsas
    - o Default value:
        - ▪ "'AB','AE','SA','JE'"
    - o Format
        - ▪ Character
        - ▪ Full string enclosed in double quotes
        - ▪ Individual values enclosed in single quotes
        - ▪ Comma-separated.

- Applies a date range filter on field BKPF_BUDAT.
    - o Variables:
        - ▪ v_start_date_bsis_bsas
        - ▪ v_end_date_bsis_bsas
    - o Default value:
        - ▪ First and last day of the previous month.
    - o Format:
        - ▪ Character
        - ▪ Format "YYYY-MM-DD" or a valid ACL expression returning a character string.
        - ▪ Literal dates are enclosed in double-quotes.
        - ▪ Expressions are not enclosed in double-quotes.

- All filters can be overwritten via the User Import Configuration File.

Post Processing Script
The post processing scripts contain code to convert a list of fields from character to numeric or date data type (especially applicable for sample data imports). The scripts run against all imported tables and convert these fields if they are found in the current table.

Error Logging
Not applicable.

6.5 S_SAP_BKPF_BSIS

General approach
Imports a selection of fields that may be most useful in supporting custom analytics.

Import Filters
- Applies a document type filter on field BKPF_BLART.
  - Variable:
    - v_doc_types_bsis_bsas
  - Default value:
    - "'AB','AE','SA','JE'"
  - Format
    - Character
    - Full string enclosed in double quotes
    - Individual values enclosed in single quotes
    - Comma-separated.

- Applies a date range filter on field BKPF_BUDAT.
  - Variables:
    - v_start_date_bsis_bsas
    - v_end_date_bsis_bsas
  - Default value:
    - First and last day of the previous month.
  - Format:
    - Character
    - Format "YYYY-MM-DD" or a valid ACL expression returning a character string.
    - Literal dates are enclosed in double-quotes.
    - Expressions are not enclosed in double-quotes.

- All filters can be overwritten via the User Import Configuration File.

Post Processing Script
The post processing scripts contain code to convert a list of fields from character to numeric or date data type (especially applicable for sample data imports). The scripts run against all imported tables and convert these fields if they are found in the current table.

Error Logging
Not applicable.

6.6 S_SAP_CDHDR_CDPOS_Banking_Changes

General approach

Imports a selection of fields that may be most useful in supporting custom analytics in the context of examining purchase order related change documents.

- Joins tables CDHDR and CDPOS on fields OBJECTCLAS and CHANGENR as an inner (matched) join.

Import Filters

- To capture entries related to banking changes, the import script has been hardcoded with the following filters:
    - OBJECTCLAS: KRED or KRED_N
    - FNAME: KEY
    - TABNAME: LFBK

- Applies a date range filter on field CDHDR_UDATE.
    - Variables:
        - v_start_date_cdhdr_cdpos_bkg
        - v_end_date_cdhdr_cdpos_bkg
    - Default value:
        - First and last day of the previous month.
    - Format:
        - Character
        - Format "YYYY-MM-DD" or a valid ACL expression returning a character string.
        - Literal dates are enclosed in double-quotes.
        - Expressions are not enclosed in double-quotes.

- The date filters can be overwritten via the User Import Configuration File.

Post Processing Script

The post processing scripts contain code to convert a list of fields from character to numeric or date data type (especially applicable for sample data imports). The scripts run against all imported tables and convert these fields if they are found in the current table.

Error Logging

Not applicable.

6.7   S_SAP_CDHDR_CDPOS_PO

General approach
Imports a selection of fields that may be most useful in supporting custom analytics in the context of examining banking related change documents.
- Joins tables CDHDR and CDPOS on fields OBJECTCLAS and CHANGENR as an inner (matched) join.

Import Filters
- To capture entries related to released purchase orders, the import script has been hardcoded with the following filter:
    - OBJECTCLAS: EINKBELEG

- Applies a filter on field CDHDR_TCODE.
    - Variable:
        - v_tcode_list_cdhdr_po
    - Default value:
        - "'ME28','ME29','E29N'"
    - Format
        - Character
        - Full string enclosed in double quotes
        - Individual values enclosed in single quotes
        - Comma-separated.

- Applies a date range filter on field CDHDR_UDATE.
    - Variables:
        - v_start_date_cdhdr_cdpos_po
        - v_end_date_cdhdr_cdpos_po
    - Default value:
        - First and last day of the previous month.
    - Format:
        - Character
        - Format "YYYY-MM-DD" or a valid expression function returning a character string.
        - Literal dates are enclosed in double-quotes.
        - Expressions are not enclosed in double-quotes.

- The date and TCODE filters can be overwritten via the User Import Configuration File.

Post Processing Script
The post processing scripts contain code to convert a list of fields from character to numeric or date data type (especially applicable for sample data imports). The scripts run against all imported tables and convert these fields if they are found in the current table.

Error Logging
Not applicable.

6.8 S_SAP_EBAN

General approach

Imports a selection of fields that may be most useful in supporting custom analytics.

Import Filters

- Applies a date range filter on field EBAN_BADAT.
    - Variables:
        - v_start_date_eban
        - v_end_date_eban
    - Default value:
        - First and last day of the previous month.
    - Format:
    - Character
    - Format "YYYY-MM-DD" or a valid expression function returning a character string.
    - Literal dates are enclosed in double-quotes.
    - Expressions are not enclosed in double-quotes.

- The date filter can be overwritten via the User Import Configuration File.

Post Processing Script

The post processing scripts contain code to convert a list of fields from character to numeric or date data type (especially applicable for sample data imports). The scripts run against all imported tables and convert these fields if they are found in the current table.

Error Logging

Not applicable.

6.9 S_SAP_EKBE

General approach

Imports a selection of fields that may be most useful in supporting custom analytics.

Import Filters

- Applies a date range filter on field EKBE_CPUDT.
    - o Variables:
        - ▪ v_start_date_ekbe
        - ▪ v_end_date_ekbe
    - o Default value:
        - ▪ First and last day of the previous month.
    - o Format:
    - o Character
    - o Format "YYYY-MM-DD" or a valid expression function returning a character string.
    - o Literal dates are enclosed in double-quotes.
    - o Expressions are not enclosed in double-quotes.

- Applies a filter for transaction / event type on field EKBE_VGABE.
    - o Variable:
        - ▪ v_include_vgabe_ekbe
    - o Default value:
        - ▪ "'1','2'"
    - o Format
        - ▪ Character
        - ▪ Full string enclosed in double quotes
        - ▪ Individual values enclosed in single quotes
        - ▪ Comma-separated.

- All filters can be overwritten via the User Import Configuration File.

Post Processing Script

The post processing scripts contain code to convert a list of fields from character to numeric or date data type (especially applicable for sample data imports). The scripts run against all imported tables and convert these fields if they are found in the current table.

Error Logging

Not applicable.

6.10 S_SAP_EKBN

General approach
Imports a selection of fields that may be most useful in supporting custom analytics.

Import Filters
- Applies a date range filter on field EBKN_ERDAT.
    - Variables:
        - v_start_date_ekbn
        - v_end_date_ekbn
    - Default value:
        - First and last day of the previous month.
    - Format:
        - Character
        - Format "YYYY-MM-DD" or a valid expression function returning a character string.
        - Literal dates are enclosed in double-quotes.
        - Expressions are not enclosed in double-quotes.

- The date filter can be overwritten via the User Import Configuration File.

Post Processing Script
The post processing scripts contain code to convert a list of fields from character to numeric or date data type (especially applicable for sample data imports). The scripts run against all imported tables and convert these fields if they are found in the current table.

Error Logging
Not applicable.

6.11 S_SAP_EKKO

General approach

Imports a selection of fields that may be most useful in supporting custom analytics.

- Table EKPO is imported separately, for joining with table EKKO via the Post Processing Script process.

Import Filters

- Applies a date range filter on field EKKO_AEDAT.
    - Variables:
        - v_start_date_ekko_ekpo
        - v_end_date_ekko_ekpo
    - Default value:
        - First and last day of the previous month.
    - Format:
        - Character
        - Format "YYYY-MM-DD" or a valid expression function returning a character string.
        - Literal dates are enclosed in double-quotes.
        - Expressions are not enclosed in double-quotes.

- The date filter can be overwritten via the User Import Configuration File.

Post Processing Script

- The post processing scripts contain code to convert a list of fields from character to numeric or date data type (especially applicable for sample data imports). The scripts run against all imported tables and convert these fields if they are found in the current table.
- The post processing scripts join table EKKO and EKPO and manage error logging if one or both tables cannot be found.

Error Logging

- The Post Processing Script process checks for the presence of both EKKO and EKPO prior to joining the two tables. If one or both are not found, the Post Processing Script process writes an entry to the Import_Error_Log table.

6.12 S_SAP_EKPO

General approach

Imports a selection of fields that may be most useful in supporting custom analytics.

- Table EKKO is imported separately, for joining with table EKPO via the Post Processing Script process.

Import Filters

- Applies a date range filter on field EKPO_AEDAT.
    - Variables:
        - v_start_date_ekko_ekpo
        - v_end_date_ekko_ekpo
    - Default value:
        - First and last day of the previous month.
    - Format:
        - Character
        - Format "YYYY-MM-DD" or a valid expression function returning a character string.
        - Literal dates are enclosed in double-quotes.
        - Expressions are not enclosed in double-quotes.

- The date filter can be overwritten via the User Import Configuration File.

Post Processing Script

- The post processing scripts contain code to convert a list of fields from character to numeric or date data type (especially applicable for sample data imports). The scripts run against all imported tables and convert these fields if they are found in the current table.
- The post processing scripts join table EKKO and EKPO and manage error logging if one or both tables cannot be found.

Error Logging

- The Post Processing Script process checks for the presence of both EKKO and EKPO prior to joining the two tables. If one or both are not found, the Post Processing Script process writes an entry to the Import_Error_Log table.

6.13 S_SAP_EKKO_EKPO

General approach
This table is created by a post-processing script, joining EKPO and EKKO as described in the individual table overview above.

6.14 S_SAP_LFA1

General approach
Imports a selection of fields sufficient to support all core analytics.

Import Filters
Not applicable

Post Processing Script
The post processing scripts contain code to convert a list of fields from character to numeric or date data type (especially applicable for sample data imports). The scripts run against all imported Oracle tables and convert these fields if they are found in the current table.

Error Logging
Not applicable.

6.15 S_SAP_LFB1

General approach
Imports a selection of fields sufficient to support all core analytics.

Import Filters
Not applicable

Post Processing Script
The post processing scripts contain code to convert a list of fields from character to numeric or date data type (especially applicable for sample data imports). The scripts run against all imported Oracle tables and convert these fields if they are found in the current table.

Error Logging
Not applicable.

6.16 S_SAP_LFBK

General approach
Imports a selection of fields sufficient to support all core analytics.

Import Filters
Not applicable

Post Processing Script
The post processing scripts contain code to convert a list of fields from character to numeric or date data type (especially applicable for sample data imports). The scripts run against all imported Oracle tables and convert these fields if they are found in the current table.

Error Logging
Not applicable.

6.17 S_SAP_MAKT

General approach
Imports a selection of fields that may be most useful in supporting custom analytics.

Import Filters
- Applies a filter on field MAKT_SPRAS.
    - o   Variable:
        - ▪ v_language_code (used by multiple tables).
    - o   Default value:
        - ▪ "E".
    - o   Format
        - ▪ Character
        - ▪ Single value.
        - ▪ Enclosed in double quotes.

- The filter can be overwritten via the User Import Configuration File.

Post Processing Script
The post processing scripts contain code to convert a list of fields from character to numeric or date data type (especially applicable for sample data imports). The scripts run against all imported Oracle tables and convert these fields if they are found in the current table.

Error Logging
Not applicable.

6.18 S_SAP_PA0000

General approach
Imports a selection of fields sufficient to support all core analytics.

Import Filters
Not applicable

Post Processing Script
The post processing scripts contain code to convert a list of fields from character to numeric or date data type (especially applicable for sample data imports). The scripts run against all imported Oracle tables and convert these fields if they are found in the current table.

Error Logging
Not applicable.

6.19 S_SAP_PA0001

General approach
Imports a selection of fields sufficient to support all core analytics.

Import Filters
Not applicable

Post Processing Script
The post processing scripts contain code to convert a list of fields from character to numeric or date data type (especially applicable for sample data imports). The scripts run against all imported Oracle tables and convert these fields if they are found in the current table.

Error Logging
Not applicable.

6.20 S_SAP_PA0002

General approach
Imports a selection of fields sufficient to support all core analytics.

Import Filters
Not applicable

Post Processing Script
The post processing scripts contain code to convert a list of fields from character to numeric or date data type (especially applicable for sample data imports). The scripts run against all imported Oracle tables and convert these fields if they are found in the current table.

Error Logging
Not applicable.

6.21 S_SAP_PA0006

General approach
Imports a selection of fields sufficient to support all core analytics.

Import Filters
Not applicable

Post Processing Script
The post processing scripts contain code to convert a list of fields from character to numeric or date data type (especially applicable for sample data imports). The scripts run against all imported Oracle tables and convert these fields if they are found in the current table.

Error Logging
Not applicable.

6.22 S_SAP_PA0009

General approach
Imports a selection of fields sufficient to support all core analytics.

Import Filters
Not applicable

Post Processing Script
The post processing scripts contain code to convert a list of fields from character to numeric or date data type (especially applicable for sample data imports). The scripts run against all imported Oracle tables and convert these fields if they are found in the current table.

Error Logging
Not applicable.

6.23 S_SAP_PA0105

General approach
Imports a selection of fields sufficient to support all core analytics.

Import Filters
Not applicable

Post Processing Script
The post processing scripts contain code to convert a list of fields from character to numeric or date data type (especially applicable for sample data imports). The scripts run against all imported Oracle tables and convert these fields if they are found in the current table.

Error Logging
Not applicable.

6.24 S_SAP_PA0149

General approach
Imports a selection of fields that may be most useful in supporting custom analytics.

Import Filters
Not applicable

Post Processing Script
The post processing scripts contain code to convert a list of fields from character to numeric or date data type (especially applicable for sample data imports). The scripts run against all imported Oracle tables and convert these fields if they are found in the current table.

Error Logging
Not applicable.

6.25 S_SAP_RBKP_RSEG

General approach
Imports a selection of fields that may be most useful in supporting custom analytics.
- Joins tables RBKP and RSEG on fields BELNR and GJAHR as an inner (matched) join.

Import Filters
- Applies a date range filter on field RBKP_BUDAT.
    - o Variables:
        - ▪ v_start_date_rbkp_rseg
        - ▪ v_end_date_rbkp_rseg
    - o Default value:
        - ▪ First and last day of the previous month.
    - o Format:
        - ▪ Character
        - ▪ Format "YYYY-MM-DD" or a valid expression function returning a character string.
        - ▪ Literal dates are enclosed in double-quotes.
        - ▪ Expressions are not enclosed in double-quotes.

- All filters can be overwritten via the User Import Configuration File.

Post Processing Script
The post processing scripts contain code to convert a list of fields from character to numeric or date data type (especially applicable for sample data imports). The scripts run against all imported tables and convert these fields if they are found in the current table.

Error Logging
Not applicable.

6.26 S_SAP_SKA1

General approach
Imports a selection of fields that may be most useful in supporting custom analytics.

Import Filters
Not applicable

Post Processing Script
The post processing scripts contain code to convert a list of fields from character to numeric or date data type (especially applicable for sample data imports). The scripts run against all imported Oracle tables and convert these fields if they are found in the current table.

Error Logging
Not applicable

6.27 S_SAP_SKAT

General approach
Imports a selection of fields that may be most useful in supporting custom analytics.

Import Filters
- Applies a filter on field SKAT_SPRAS.
    - Variable:
        - v_language_code (used by multiple tables).
    - Default value:
        - "E".
    - Format
        - Character
        - Single value
        - Enclosed in double quotes

- Applies a filter on field SKAT_KTOPL.
    - Variable:
        - v_coa_skat
    - Default value:
        - All chart of account entries will be imported.
    - Format
        - Character
        - Single value
        - Enclosed in single quotes and double-quotes

- All filters can be overwritten via the User Import Configuration File.

Post Processing Script
The post processing scripts contain code to convert a list of fields from character to numeric or date data type (especially applicable for sample data imports). The scripts run against all imported Oracle tables and convert these fields if they are found in the current table.

Error Logging
Not applicable.

6.28 S_SAP_T001

General approach
Imports a selection of fields that may be most useful in supporting custom analytics.

Import Filters
Not applicable

Post Processing Script
The post processing scripts contain code to convert a list of fields from character to numeric or date data type (especially applicable for sample data imports). The scripts run against all imported Oracle tables and convert these fields if they are found in the current table.

Error Logging
Not applicable

6.29 S_SAP_T001K

General approach
Imports a selection of fields that may be most useful in supporting custom analytics.

Import Filters
Not applicable

Post Processing Script
The post processing scripts contain code to convert a list of fields from character to numeric or date data type (especially applicable for sample data imports). The scripts run against all imported Oracle tables and convert these fields if they are found in the current table.

Error Logging
Not applicable

6.30 S_SAP_T001W

General approach
Imports a selection of fields that may be most useful in supporting custom analytics.

Import Filters
Not applicable

Post Processing Script
The post processing scripts contain code to convert a list of fields from character to numeric or date data type (especially applicable for sample data imports). The scripts run against all imported Oracle tables and convert these fields if they are found in the current table.

Error Logging
Not applicable

6.31 S_SAP_T052

General approach
Imports a selection of fields that may be most useful in supporting custom analytics.

Import Filters
Not applicable

Post Processing Script
The post processing scripts contain code to convert a list of fields from character to numeric or date data type (especially applicable for sample data imports). The scripts run against all imported Oracle tables and convert these fields if they are found in the current table.

Error Logging
Not applicable

6.32 S_SAP_T052U

General approach
Imports a selection of fields that may be most useful in supporting custom analytics.

Import Filters
Not applicable

Post Processing Script
The post processing scripts contain code to convert a list of fields from character to numeric or date data type (especially applicable for sample data imports). The scripts run against all imported Oracle tables and convert these fields if they are found in the current table.

Error Logging
Not applicable

6.33 S_SAP_T500P

General approach
Imports a selection of fields sufficient to support all core analytics.

Import Filters
Not applicable

Post Processing Script
The post processing scripts contain code to convert a list of fields from character to numeric or date data type (especially applicable for sample data imports). The scripts run against all imported Oracle tables and convert these fields if they are found in the current table.

Error Logging
Not applicable

6.34 S_SAP_T528T

General approach

Imports a selection of fields that may be most useful in supporting custom analytics.

Import Filters

- Applies a filter on field T528T_SPRSL
    - o Variable:
        - ▪ v_language_code (used by multiple tables).
    - o Default value:
        - ▪ "E"
    - o Format
        - ▪ Character
        - ▪ Single value
        - ▪ Enclosed in double quotes

Post Processing Script

The post processing scripts contain code to convert a list of fields from character to numeric or date data type (especially applicable for sample data imports). The scripts run against all imported Oracle tables and convert these fields if they are found in the current table.

Error Logging

Not applicable.

6.35 S_SAP_TBSLT

General approach
Imports a selection of fields that may be most useful in supporting custom analytics.

Import Filters
- Applies a filter on field TBSLT_SPRAS.
    - o   Variable:
        - ▪   v_language_code (used by multiple tables).
    - o   Default value:
        - ▪   The Default Import Configuration File sets the default to "E".
    - o   Format
        - ▪   Character
        - ▪   Single value
        - ▪   Enclosed in double quotes

- The filter can be overwritten via the User Import Configuration File.

Post Processing Script
The post processing scripts contain code to convert a list of fields from character to numeric or date data type (especially applicable for sample data imports). The scripts run against all imported Oracle tables and convert these fields if they are found in the current table.

Error Logging
Not applicable.

6.36 S_SAP_TGSBT

General approach
Imports a selection of fields that may be most useful in supporting custom analytics.

Import Filters
- Applies a filter on field TGSBT_SPRAS.
  - Variable:
    - v_language_code (used by multiple tables).
  - Default value:
    - The Default Import Configuration File sets the default to "E".
  - Format
    - Character
    - Single value
    - Enclosed in double quotes

- The filter can be overwritten via the User Import Configuration File.

Post Processing Script
The post processing scripts contain code to convert a list of fields from character to numeric or date data type (especially applicable for sample data imports). The scripts run against all imported Oracle tables and convert these fields if they are found in the current table.

Error Logging
Not applicable.

6.37 S_SAP_TSTCT

General approach
Imports a selection of fields that may be most useful in supporting custom analytics.

Import Filters
- Applies a filter on field TSTCT_SPRSL.
    - o Variable:
        - ▪ v_language_code (used by multiple tables).
    - o Default value:
        - ▪ "E"
    - o Format
        - ▪ Character
        - ▪ Single value
        - ▪ Enclosed in double quotes

- The filter can be overwritten via the User Import Configuration File

Post Processing Script
The post processing scripts contain code to convert a list of fields from character to numeric or date data type (especially applicable for sample data imports). The scripts run against all imported Oracle tables and convert these fields if they are found in the current table.

Error Logging
Not applicable.

6.38 S_SAP_USER_ADDRP

General approach
Imports a selection of fields sufficient to support all core analytics.

Import Filters
Not applicable

Post Processing Script
The post processing scripts contain code to convert a list of fields from character to numeric or date data type (especially applicable for sample data imports). The scripts run against all imported Oracle tables and convert these fields if they are found in the current table.

Error Logging
Not applicable

6.39 S_SAP_USR02_USTYP

General approach

Imports a selection of fields sufficient to support all core analytics.

- Converts logical field IsActive to character to align with the sample data import (all character).

Import Filters

- Field USR21_USTYP is filtered for user types A (Dialog) and C (Communication User).
- The filter is hardcoded in the ACCESSDATA command.

Post Processing Script

The post processing scripts contain code to convert a list of fields from character to numeric or date data type (especially applicable for sample data imports). The scripts run against all imported Oracle tables and convert these fields if they are found in the current table.

Error Logging

Not applicable

7 Frequently Asked Questions / Troubleshooting

7.1 The parameter values or disabled scripts in the User Import Configuration File are not being applied

There are multiple situations in which the user configurations are not applied.

1) Check the Import Error Log table (uploaded as part of the result files of the robot). The error log reports issues such as a missing or incorrectly configured User Import Configuration File. In these cases, the Default Import Configuration File is used.
    a. Ensure the file has been uploaded to the Input / Output tab in the mode the robot is running in (Development and / or Production).
    b. Ensure the worksheet names and field names match the expected conventions. To confirm the expected worksheet names and field names, refer to the Default Import Configuration File, worksheet Expected_User_Config_Fields.

2) An invalid code has been entered in field Disable_Import.
    a. Ensure that the codes entered match those specified in variable **v_main_disabled_codes** in the start script (_<source system>_Imports). Codes not specified in the variable do not disable the import script.

3) If the robot is running in Sample Data mode, the parameter configurations from the User Import Configuration File are not being used.
    a. The Import Error Log has a note stating the robot is running in sample data mode.

7.2 Error message "Enter the file name" causes the process to fail

This error message can be caused by a misspelled import script name in the Import_Name column in the User Import Configuration File.

7.3 The record count of imported tables appears to be limited to 500 records

1) Edit the task to make sure the 500-record limit has been turned off.
2) Review any scripts, including custom scripts, to make sure the 500-record limitation is not hard-coded, for example, when copying syntax from the log to create the script.

7.4   In the Input / Output tab, a disabled script is showing a record count

1) This scenario is possible when a robot has been run once before with the table enabled, and the table has subsequently been disabled.
   a. The record count and update date reflect the last time the table was populated.
   b. Any analytics still using this table are using the available data as of the last run date.
   c. To prevent this situation, click and delete the table from the Input / Output tab.
   d. Ensure that all analytics requiring this table in any analysis robots are disabled via the analysis robots' User Analytic Configuration Files.

7.5   What is "post processing"?

Post processing refers to data cleansing taking place within the data integration robot. The tasks performed by post processing scripts vary by data integration robot. Most commonly, there may be a difference between data types imported from the production source system and in sample data mode. Post processing scripts align the data types so that any dependent analytics have a consistent data set to work with.

Other post processing may involve renaming sample data tables imported from the Default Import Configuration File if the table name was truncated in Excel. To ensure the source tables have the expected table names regardless of whether they came from a data connector or the sample data process, post processing checks the names of these tables for truncation and applies the expected name.

Post processing is an optional task in a data integration robot, and its use is determined by the developer. If a post processing script is present, it runs last, after all imports are completed. Post processing scripts are limited to original scripts, they are not intended to be used with custom scripts. Ensure all custom scripts perform their own data cleansing as needed.

7.6 What is the difference between data cleansing and data preparation?

You may hear the terms "data cleansing" in the context of a data integration robot, and "data preparation" in the context of an analysis robot.

Data cleansing is generally defined as harmonization or other processing – such as creation of computed fields – that apply to a table for most or all use cases. For example, a transaction date field is being imported as character. Every conceivable use of this field would require a DATE data type. Therefore, converting the data from character to date is a suitable application of data cleansing to be performed in the data integration robot. By performing the task in data integration, the correct data type is available to all dependent analysis robots. Joining of header and line item tables (if imported separately) is another application of data cleansing. These types of tasks are performed in post processing scripts that apply to data either imported via the data connector or sample data.

Data preparation is generally defined as harmonization or other processing limited to a small number of specialized analytics. For example, to obtain a quarterly summarization of data, a computed field showing only the year and the month of the transaction date is required. This type of task, which supports only a small number of analytics, is considered data preparation and is performed only in the analysis robots that need this field.

The line between data cleansing and data preparation can be blurry. If a decision to include a given task in either data cleansing or data preparation is causing issues, please report the scenario via the standard support / reporting channels for evaluation.

7.7 The default parameter values, imported tables, or imported fields do not match most clients' needs

Please report these findings via the standard support / reporting channels for evaluation. If the case is compelling and affects most customers, the master default configuration and master list of imported scripts may be updated in a new release to eliminate the need for customization.

8 Running a data integration robot in ACL Analytics

Although data integration robots have been optimized for use as a robot, they can also be run in ACL Analytics.

1) If not previously tested, ensure that a connection to the source system can be established from within ACL Analytics.

2) Ensure that all expected files such as Default Import Configuration Files, User Import Configuration Files, and any custom scripts or external source files are in the same folder as the ACL project. Ensure all files are closed.

3) Within the correct version of the ACL project (Unicode, non-Unicode), open the start script, named _<source_system>_Imports, for example _SAP_Imports.

4) Review the analytic header for any changes that need to be made to the parameter prompts.
    a. Default values have been provided for some parameters such as sample data mode, the 500-records limitation, or some common connectivity settings. Update the default values as needed.
    b. Other parameters require custom input, such as user account names or customer specific connectivity settings specific to the source systems.

5) Run the start script to kick off the process.

6) If any of the source tables are required by an analysis robot, save the data integration robot to update the project with the latest working copy.
    a. In the analysis robot, copy the table layouts from the data integration robot.
        i. If the table layouts are not available for selection, it is likely that the data integration project was not saved.
    b. Copy the .fil files to the analysis robot project folder or manually link the copied tables to the source files, if needed.
    c. Repeat these steps if toggling between sample data mode and production mode as the layouts will differ, resulting in skewed data.