



ERP/CRM Integrator

Task Automation Guide

ERP/CRM Integrator Task Automation Guide

Revision: 08 Jun 2023

You can find the most up-to-date technical documentation on our documentation site at

https://productresources.collibra.com/docs/erp-crm-integrator/latest/Content/TaskAutomation/to_task-automation.htm

Contents

Contents	i	
What's new	ii	
Introduction	1	
Overview of the Task Automation approach for ERP/CRM Integrator	2	
What Tasks can be Automated?	2	
What does the ERP/CRM Integrator Task Automation process actually mean?	2	
How to create and use the infrastructure for Task Automation	4	
Hardware and Software Environment	5	
Building the Excel file for the Required Tasks	5	
Building the Command file for the Required Tasks	6	
Excel Property Settings for each Command Type	7	
Extract Task	7	
Subject Area Import Task	8	
Subject Area Expand Task	9	
Subject Area Export Task	12	
Testing & Running ERP/CRM Integrator Task Automation	17	
Appendix A - Running ERP/CRM Integrator in background	18	
Overview	18	
Windows Task Scheduler Configuration	18	
Command File Modification	19	
ERP/CRM Integrator Task Automation Excel Spreadsheet	19	
Possible Problems	20	
Index	21	

What's new

Introduction

Overview of the ERP/CRM Integrator Task Automation Approach

This manual describes how to configure ERP/CRM Integrator to achieve many of the day-to-day task required of the product without the user having to be present and (if desired) as a scheduled process.



Overview of the Task Automation approach for ERP/CRM Integrator

Running ERP/CRM Integrator unattended is achieved by using a command line option in ERP/CRM Integrator. This command line mode uses a .cmd file to describe the Tasks to be automated, and the Command file in turn makes use of settings in an Excel spreadsheet to determine the 'variables' such as:

- Which ERP/CRM Integrator repository the Task is to be run against
- For Export tasks, the various export options
- For Subject Area-related Tasks, the Subject Area involved

What Tasks can be Automated?

Three types of Tasks can be automated:

- Extraction of metadata from the source application (SAP, Salesforce, PeopleSoft...)
- Creation and Expansion of Subject Areas
- Export of Subject Areas (not all export formats supported)

What does the ERP/CRM Integrator Task Automation process actually mean?

When ERP/CRM Integrator is run with the Task Automation options, the program is usually run as a 'foreground' Windows application. The PC on which ERP/CRM Integrator is running is active at the time(s) the Task(s) are run. Most of the discussion below assumes that mode for simplicity. However, the program can also be run as a 'background' application with the PC logged off and potentially sleeping. This mode requires some additional configuration to cope with unattended operation: this is described in Appendix A.

The program will need access to any of the resources required to complete the Task successfully, such as Databases, Shared Folder and Application connections.

Task Automation with ERP/CRM Integrator uses a command line option. In order to schedule the Tasks, a Scheduling program, such as the Windows Task Scheduler, will need to be employed.

How to create and use the infrastructure for Task Automation

Describes what is required to successfully use the Task Automation features

There are two main considerations when creating the infrastructure to enable Task Automation with ERP/CRM Integrator:

- The hardware/software environment that will be used
- The Command and Spreadsheet files and setting that will control the Task Automation process



Hardware and Software Environment

The PC on which ERP/CRM Integrator is running for the Task Automation process needs to have access to any of the resources required to complete the Task(s).

In particular, the Windows User account under which ERP/CRM Integrator is being run needs permissions to access these resources. The recommended approach to achieving this is to run each of the required steps 'manually' (that is, using the normal ERP/CRM Integrator GUI interface), using the same Windows User account that will be used during Task Automation, in order to ensure that the process can be run successfully before creating the automation process.

Building the Excel file for the Required Tasks

A sample Excel file is provided with the ERP/CRM Integrator software. The file is called SafyrTaskAutomationSample.xlsx and is located in the 'Doc' sub folder of the ERP/CRM Integrator installation folder.

The recommended approach is to take a copy of this file and place it in a new folder where the other Task Automation files will be stored.

The required cells for each Excel sheet are dependent on the type of Task being automated. The possible values for these are described in the following sections.

The example below can be used to describe the approach.

	A	B	C
1	#CommandProperties		
2	PropertyName	PropertyValue	Description
3	RepositoryName	SAPBW-1	
4	CommandType	Extract	Type of the Command (limited to the values describe in this sample sheet)
5	MinimizeWindow	Yes	
6	CloseAfterCommand	Yes	
7	LogFileName	c:\temp\Safyr_SAP1-\$DTS-Extract.txt	Option / \$DTS = current date time as 'yyyymmdd-hhmmss'
8	LogFileAppend	No	
9			
10	ExtractSkipTest	Yes	
11	ExtractDataMode	FromLoadFiles	Full ; To LoadFiles ; FromLoadFiles (support depends on source system type)
12	ExtractDetailedLog	c:\temp\Safyr_SAP1-\$DTS-Extract.xml	Option / \$DTS = current date time as 'yyyymmdd-hhmmss'
13			

The main property values are:

Property	Description
RepositoryName	This specifies which ERP/CRM Integrator Repository is the subject of the Task
CommandType	This can be Extract, SubjectAreaImport, SubjectAreaExpand or Export
MinimiseWindow	Is the ERP/CRM Integrator application going to be minimised during execution
CloseAfterCommand	Should ERP/CRM Integrator be closed after the Task has been completed?
LogFileName	The name and location of the log file that is written to show the progress of the Task

Building the Command file for the Required Tasks

A sample Command file is provided with the ERP/CRM Integrator software. The file is called `SafyrTaskAutomationSample.cmd` and is located in the 'Doc' sub folder of the ERP/CRM Integrator installation folder.

As with the Excel file, the recommended approach is to take a copy of this file and place it in a new folder where the other Task Automation files will be stored.

The example below can be used to describe the approach.

```

echo off
cls
set SafyrExe="X:\DP_PRJ\Safyr\Prog\Safyr.exe"

echo SAP1-Extract is running ...
%SafyrExe% /TaskAutomationFile="X:\DP_PRJ\Safyr\Docu\Manual\User\SafyrTaskAutomationSample.xlsx" /TaskAutomationCommand="SAP1-Extract"
if %ERRORLEVEL% GEQ 1 goto :FinishWithError

echo SAP1-Subject Area Import is running ...
%SafyrExe% /TaskAutomationFile="X:\DP_PRJ\Safyr\Docu\Manual\User\SafyrTaskAutomationSample.xlsx" /TaskAutomationCommand="SAP1-SubjectAreaImport"
if %ERRORLEVEL% GEQ 1 goto :FinishWithError

echo SAP1-Subject Area Expand is running ...
%SafyrExe% /TaskAutomationFile="X:\DP_PRJ\Safyr\Docu\Manual\User\SafyrTaskAutomationSample.xlsx" /TaskAutomationCommand="SAP1-SubjectAreaExpand"
if %ERRORLEVEL% GEQ 1 goto :FinishWithError

echo SAP1-Export-JSON is running ...
%SafyrExe% /TaskAutomationFile="X:\DP_PRJ\Safyr\Docu\Manual\User\SafyrTaskAutomationSample.xlsx" /TaskAutomationCommand="SAP1-Export-JSON"
if %ERRORLEVEL% GEQ 1 goto :FinishWithError

goto FinishSuccessful

:FinishWithError
echo Something failed, please check the result logs
pause
rem force exit of higher level command calls
exit /B 1

:FinishSuccessful
echo everything worked
pause
exit /B 0

```

The first step sets a location for the Safyr.exe program

The following steps then run:

- An Extract from SAP
- Creation of a Subject Area via the Excel Import method
- The Expansion of that Subject Area
- An Export to the JSON Data Catalog format

For each Task there are two Command Line options:

- /TaskAutomationFile: The location of the Excel file containing the Task properties
- /TaskAutomationCommand: The Sheet in the Excel file to be used

Running ERP/CRM Integrator with command line options returns an ERRORLEVEL return code. In the example Command file shown, this is used to set the EXIT command exitcode to 0 if the Task has completed successfully and 1 if otherwise.

Excel Property Settings for each Command Type

The following sections describe the four types of Task and their Excel settings.

Extract Task

This Task type deals with the extraction of metadata from the chosen source system type. The Excel Properties are described below.

Property	Description
RepositoryName	This specifies which ERP/CRM Integrator Repository is the subject of the Task
CommandType	This will be Extract for this Task type
MinimiseWindow	Is the ERP/CRM Integrator application going to be minimised during execution - Yes/No
CloseAfterCommand	Should ERP/CRM Integrator be closed after the Task has been completed? - Yes/No

Property	Description
LogFileName	The name and location of the log file that is written to show the progress of the Task. The parameter \$DTS\$ can be used in the name to insert a date/time string into the file name. e.g. c:\temp\Safyr_SAP1-\$DT\$-Extract.txt
LogFileAppend	If the LogFileName already exists, Append the entries to that file - Yes/No
ExtractSkipTest	An Extract from the source application is normally performed with a 'Test' Extract first. This Property can be used to suppress the Test - Yes/No
ExtractDataMode	Extractions form SAP can be performed as one step, or in two steps using text files. Values can be Full, To LoadFiles,From LoadFiles
ExtractDetailedLog	The name and location of the detailed ERP/CRM Integrator XML log file. The parameter \$DTS\$ can be used in the name to insert a date/time string into the file name. e.g. c:\temp\Safyr_SAP1-\$DT\$-Extract.xml

Subject Area Import Task

This Task type will load ERP/CRM Integrator Subjects areas using the Excel load method (see 'Appendix C - Using an Excel spreadsheet to load Subject Areas' in the ERP/CRM Integrator User Guide). The Properties are described below.

Property	Description
RepositoryName	This specifies which ERP/CRM Integrator Repository is the subject of the Task
CommandType	This will be SubjectAreaImport for this Task type
MinimiseWindow	Is the ERP/CRM Integrator application going to be minimised during execution - Yes/No
CloseAfterCommand	Should ERP/CRM Integrator be closed after the Task has been completed? - Yes/No

Property	Description
LogFileName	The name and location of the log file that is written to show the progress of the Task. The parameter \$DTS\$ can be used in the name to insert a date/time string into the file name. e.g. c:\temp\Safyr_SAP1-\$DT\$-SubjectAreaImport.txt
LogFileAppend	If the LogFileName already exists, Append the entries to that file - Yes/No
SubjectAreaImportFile	The name of the Excel file that contains the Subject area import settings
SubjectAreaImportDetailedLog	The name and location of the detailed ERP/CRM Integrator XML log file. The parameter \$DTS\$ can be used in the name to insert a date/time string into the file name. e.g. c:\temp\Safyr_SAP1-\$DT\$-SubjectAreaImport.xml

Subject Area Expand Task

This Task type will Expand the specified Subject Area(s). See 'Expanding a Subject Area with related Parent or Child tables' in the ERP/CRM Integrator User Guide for details of this feature. The Properties are described below.

Property	Description
RepositoryName	This specifies which ERP/CRM Integrator Repository is the subject of the Task
CommandType	This will be SubjectAreaExpand for this Task type
MinimiseWindow	Is the ERP/CRM Integrator application going to be minimised during execution - Yes/No

Property	Description
CloseAfterCommand	Should ERP/CRM Integrator be closed after the Task has been completed? - Yes/No
LogFileName	The name and location of the log file that is written to show the progress of the Task. The parameter \$DTS\$ can be used in the name to insert a date/time string into the file name. e.g. c:\temp\Safyr_SAP1-\$DTS\$-Expand.txt
LogFileAppend	If the LogFileName already exists, Append the entries to that file - Yes/No
SubjectAreaExpandNames	Name or Names of the Subject areas to be the subject of the Expand

Property	Description	
OTHER THAN SAP BW	SubjectAreaExpandParentPKLevel	Number of levels to include those relationships where the primary key fields of the 'parent' table form part of the primary key of the child table
	SubjectAreaExpandParentNonPKLevel	Number of levels to include Non-Identifying relationships. That is, those relationships where the primary key fields of the 'parent' table do not form part of the primary key of the child table
	SubjectAreaExpandChildLevel	Number of levels to include Child relationships
	SubjectAreaExpandChildOnlyIdentifying	All child relationships or Identifying only child relationships. Yes/No

Property		Description
SAP BW	SubjectAreaExpandUpStreamLineageLevel	Number of 'Upstream' levels to include when Expanding to include BW lineage
	SubjectAreaExpandDownStreamLineageLevel	Number of 'Downstream' levels to include when Expanding to include BW lineage
	SubjectAreaExpandSapBwInfoObjectLevel	Number of InfoObject levels to include when Expanding to include BW lineage
SubjectAreaExpandNoOfEntities		This will limit the number of Entities. '0' denotes no limit
SubjectAreaExpandToTablesWithData		Only include tables containing data: The set of tables resulting will be restricted to those that have a Row Count greater than zero. Yes/No

Subject Area Export Task

This Task type will Export the specified Subject Area(s). The properties will be slightly different depending on the 'target' tool (not all Export targets are available). The table below describes the properties for JSON, Erwin, ER/Studio and PowerDesigner exports. Not all properties are relevant to all export types and this is identified in the table.

Property	Description
RepositoryName	This specifies which ERP/CRM Integrator Repository is the subject of the Task
CommandType	This will be Export for this Task type
MinimiseWindow	Is the ERP/CRM Integrator application going to be minimised during execution - Yes/No
CloseAfterCommand	Should ERP/CRM Integrator be closed after the Task has been completed? - Yes/No
LogFileName	The name and location of the log file that is written to show the progress of the Task. The parameter \$DTS\$ can be used in the name to insert a date/time string into the file name. e.g. c:\temp\ERP/CRM Integrator_SAP1-\$DT\$-Export.txt
LogFileAppend	If the LogFileName already exists, Append the entries to that file - Yes/No
ExpType	Data Catalog JSON/Erwin/ERStudio or PowerDesigner, as required
ExpSubjectAreas	Name or Names of the Subject areas to be Exported

Property		Description
ExpDetailedLog		The name and location of the detailed ERP/CRM Integrator XML log file. The parameter \$DTS\$ can be used in the name to insert a date/time string into the file name. e.g. c:\temp\ERP/CRM Integrator_SAP1-\$DTS\$-Export.xml
JSON	ExpDataCatalogJsonSchemaPrefix	Prefix that is added to exported Assets
	ExpDataCatalogJsonIncludeErpHierarchy	Currently only available for SAP systems. Nodes from the SAP Application Component//Package tree, or the SAP BW Info Area tree will be exported, where those nodes are attached to Assets being exported.
	ExpDataCatalogJsonFilePath	Path for the JSON Asset & Relation files created during the Export process
	ExpDataCatalogJsonAddLngIds	A comma delimited list of additional export language Ids
	ExpDataCatalogJsonFilePathClear	Remove any JSON files in the specified folder that already exist

Property		Description
ERWIN	ExpErwinFileName	Name and folder where the Erwin file will be written
	ExpErwinNotationIdef1x	If the Erwin model will be written using the IDEF1X Notation - Yes/No. If 'No' the Information Engineering notation will be used
ERSTUDIO	ExpERStudioFileName	Name and folder where the ERX file will be written
	ExpERStudioFileUTF8Encoded	If the file will be written using UTF-8 encoding - Yes/No.
P'DESIGNER	ExpPDFFileName	Name and folder where the PowerDesigner file will be written
OTHER THAN SAP BW	ExpLimitParentPKLevel	Number of levels to include those relationships where the primary key fields of the 'parent' table form part of the primary key of the child table
ExportLimitParentNonPKLevel		Number of levels to include Non-Identifying relationships. That is, those relationships where the primary key fields of the 'parent' table do not form part of the primary key of the child table

Property		Description
ExportLimitChildLevel		Number of levels to include Child relationships
ExpLimitChildOnlyIdentifying		All child relationships or Identifying only child relationships. Yes/No
SAP BW	ExpLimitUpStreamLineageLevel	Number of 'Upstream' levels to include when Expanding to include BW lineage
	ExpLimitDownStreamLineageLevel	Number of 'Downstream' levels to include when Expanding to include BW lineage
	ExpLimitBwInfoObjectLevel	Number of InfoObject levels to include when Expanding to include BW lineage
ExpLimitNoOfEntities		This will limit the number of Entities. '0' denotes no limit
ExpLimitToTablesWithData		Only include tables containing data: The set of tables resulting will be restricted to those that have a Row Count greater than zero. Yes/No
ExpLimitToMkFields		The set of fields exported for each table will be limited to the Marked Fields if this option is selected, plus the Primary Key fields of the table. Yes/No

Testing & Running ERP/CRM Integrator Task Automation

After creating the Excel and Command files described in the previous chapter, the Task Automation process can be tested.

Running each of the steps to be automated in 'interactive' mode (i.e. not using Automation), is recommended first, in order to ensure that there are no obvious issues that might be encountered during the 'unattended' execution of ERP/CRM Integrator.

This should be done using the same Windows user as that is to be used when running the 'unattended' Task(s).

Appendix A - Running ERP/CRM Integrator in background

Overview

This appendix describes how to run ERP/CRM Integrator as a 'background' application with the PC logged off and potentially sleeping.

The description assumes the task scheduler and the ERP/CRM Integrator automation files have already been configured and tested in foreground operation. Background operation requires some additional configuration settings to cope with unattended operation. The recommended settings are described below.

Windows Task Scheduler Configuration

This section assumes the Windows Task Scheduler is being used. If another task scheduler is used different settings will be required. Please note that background operation has not been tested with any other task scheduler.

General Tab

Select Run whether user is logged on or not.

Select Run with highest privileges.

Conditions Tab

Select Wake the computer to run this task. This should allow the Windows Task Scheduler to run even if the machine is sleeping. However, this may not work on some systems (see Possible Problems below).

Optionally, select Start the task only if the computer is on AC power and Stop if the computer switches to battery power.

Settings Tab

Select Stop the task if it runs longer than and set an expected time limit for the task.

Optionally, select If the running task does not end when requested force it to stop.

Select If the task is already running, then the following rule applies: Do not start a new instance.

Command File Modification

Remove or comment-out any pause commands (or equivalent) in the Command File or make sure they are only called by the command script in foreground operation.

The sample Command File includes pause commands to allow the user to see what has happened. This is fine when running in foreground. Anything that waits indefinitely for user input in background will cause the scheduled task to hang. This in turn makes the Task Scheduler think the task is still running, so it never properly completes. It can also stop the task from being rerun subsequently because the Task Scheduler has an option in its Settings tab about what to do if a task is already running and the best option is usually not to start a new instance.

ERP/CRM Integrator Task Automation Excel Spreadsheet

Select the appropriate Task worksheet.

Set the MinimizeWindow property to Yes.

Set the CloseAfterCommand property to Yes to tell ERP/CRM Integrator to exit after the task has completed.

Set the LogFileName property to a path and filename for the log file and set the LogFileAppend property to Yes or No as desired.

Possible Problems

Scheduled Task Hanging

If the scheduled task is shown as running but Task Manager shows the ERP/CRM Integrator.exe process is idle check the Command File does not do or call anything that waits indefinitely on user (or other) input.

A hang may also be caused by another copy of ERP/CRM Integrator running (either an interactive one or a previous hanging scheduled one). This might be causing a lock on a file (eg: a SQLite repository file). Use Task Manager to check there is no other copy of the ERP/CRM Integrator.exe process running. If there is, stop it and try running the scheduled task again.

Wakeup Does Not Work On Some Sleeping Machines

In the Windows Task Scheduler Conditions tab the option Wake the computer to run this task should be selected. This should allow the task to run even if the system is sleeping. However, this is dependent on support for Wake Timers and may not work on some physical or virtual machines.

Check the Wake Timers setting in the Power Plan. Select Control Panel / Power Options / Change plan settings / Change advanced power settings. Expand Sleep / Allow wake timers and confirm it is set to Enabled.

Also check the machine is actually sleeping and not hibernating. Wake Timers may not work for some hibernating machines.

If Wake Timers still do not work the problem can be avoided by configuring the machine to never sleep, but sleep is a useful feature if they do.

Index

C

Command File for Task
Automation 6

E

Excel file for Task
Automation 5

Extract Task 7

H

Hardware & Software
Requirements 5

S

Subject Area Expand Task 9

Subject Area Export Task 12

Subject Area Import Task 8

T

Testing Task Automation 17