

SAP SuccessFactors Employee Central  
Document Version: Q3 2016 – August 5

# Using the Employee Delta Export Add-in for Microsoft Excel

---

# Content

<b>1</b>	<b>Must Read.</b>	<b>5</b>
<b>2</b>	<b>Overview.</b>	<b>6</b>
2.1	Features and Functions.	6
2.2	Target Group.	7
<b>3</b>	<b>Business Context and Use.</b>	<b>8</b>
3.1	Prerequisites.	8
3.2	Limitations.	9
	Limitations caused by the backend.	9
	Limitations caused by the client application.	10
3.3	Security and Session Management.	11
3.4	Role-based permissions.	12
3.5	Application Components.	12
3.6	Employee Delta Export UI.	13
3.7	Version Check and Updating the Add-in.	13
3.8	Installation.	14
3.9	Uninstalling the add-in.	14
<b>4</b>	<b>Working with Employee Delta Export.</b>	<b>15</b>
4.1	Quick Start.	15
	Finding your URL.	16
	Example test run.	17
4.2	Architecture Concept.	17
	Component Structure.	17
	Data Extraction.	19
	Understanding Delta Calculation.	19
4.3	Workbooks.	25
4.4	Worksheets.	26
	Result Sheets.	26
	Selection Sheet.	29
	Summary Sheets.	36
	Systems Sheet.	36
	Config Sheet.	36
	Errors Sheet.	41
	Warnings Sheet.	41
	Internal Sheets.	41
4.5	Tabs on the Ribbon.	43

SuccessFactors Tab. . . . .	43
Developer Tab. . . . .	43
<b>5 Extending Employee Delta Export. . . . .</b>	<b>50</b>
5.1 Column Configuration. . . . .	50
5.2 Adding new columns from the database . . . . .	52
5.3 Adding Calculated Columns. . . . .	53
5.4 Adding user-friendly descriptions for picklist-based columns. . . . .	53
5.5 Adding user-friendly descriptions for FO-based columns. . . . .	54
5.6 Deleting columns. . . . .	54
5.7 Adding personal data for user ID fields. . . . .	55
5.8 Adding information about workflow approvals. . . . .	56
5.9 Adding new filter parameters . . . . .	57
5.10 Sorting in the Result Sheet. . . . .	57
5.11 Using Payment Information . . . . .	58
5.12 Using the Requested Segments Table. . . . .	58
5.13 Dos and Don'ts . . . . .	59
5.14 Tips and Tricks. . . . .	61
<b>6 Trouble Shooting. . . . .</b>	<b>62</b>
<b>7 Appendix. . . . .</b>	<b>63</b>
7.1 Screenshots. . . . .	63

# What's New in this Guide

## Q3 2016

Table 1: The following table summarizes changes to this guide for the Q3 2016 release

What's new	Description	More Info
Correction	Minor correction in text. Removed screenshots to improve accessibility.	<a href="#">Version Check and Updating the Add-in [page 13]</a>

## Q2 2016

Table 2: The following table summarizes changes to this guide for the Q2 2016 release.

What's new	Description	More Info
Permissions	The permissions required to extract data from the backend have changed.	<a href="#">Role-based permissions [page 12]</a>

## Q1 2016

Table 3: The following table summarizes changes to this guide for the Q1 2016 release.

What's new	Description	More Info
<a href="#">Requested Segments</a> table on the <a href="#">Config Sheet</a>	<p>You can now define which segments are included as result sheets and which aren't using the <a href="#">Requested Segments</a> table. You can also request the following segments that aren't available in the standard query:</p> <ul style="list-style-type: none"><li>• Deduction Recurring</li><li>• One Time Deduction</li><li>• Alternative Cost Distribution</li><li>• Personal Documents Information</li></ul>	<a href="#">Using the Requested Segments Table [page 58]</a>

---

# 1 Must Read

The application described in this document is not subject to your license agreement or any other agreement with SAP/SuccessFactors. SAP/SuccessFactors has no obligation to pursue any course of business outlined in this document or to develop or release any functionality mentioned in this document. This document and SAP/SuccessFactors' strategy and possible future developments are subject to change and may be changed by SAP/SuccessFactors at any time for any reason without notice. This document is provided without a warranty of any kind, either express or implied, including, but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. SAP/SuccessFactors assumes no responsibility for errors or omissions in this document.



---

## 2 Overview

Employee Delta Export for Microsoft Excel is an Excel-based application that retrieves employee master data from the Employee Central backend which can then be used for payroll runs in a subsequent system. It provides the new data and the data changed compared to the end time and date of the previous run and thus minimizes the effort of transferring the changed employee data to the payroll system.

Employee Delta Export considers changes on all time slices of effective-dated employee data. In other words, it considers future-dated and retroactive changes of employee data. It indicates data deletions in certain cases, for example, if a pay component is deleted in the backend.

For more information on time slices see section *Understanding Delta Calculation*.

The Employee Delta Export application consists of:

- Add-in for Microsoft Excel
- 2 workbooks

The workbooks provide the layout of the data being retrieved from the backend and some configuration data. The configuration data allows for consultants and administrators to configure the data retrieval process without actually having to program. This means, however, that some advanced knowledge of working with Microsoft Excel and XML is needed.

### Related Information

[Understanding Delta Calculation \[page 19\]](#)

## 2.1 Features and Functions

The main functions of Employee Delta Export for Microsoft Excel are:

- Filter employees who have been changed, hired, or re-hired since the begin of the evaluation period
- Show before image and current image of changed data
- Consider changes along the full history of employees (retroactive changes and future-dated changes)
- Highlight changes for easy manual processing
- Support all selection criteria offered by the underlying API in the backend
- Support additional selection criteria that are evaluated in Excel locally, on your machine
- Lookup user-friendly descriptions for picklist and foundation object based fields
- Lookup additional information for related persons (for example manager's first/last name)
- Include workflow approval information
- Easy configuration of columns shown on the worksheets without programming

- 
- Easy configuration of how the returned changes on each sheet are sorted
  - Support for locally calculated columns derived from backend fields
  - User-defined selection criteria
  - User-defined table styles
  - One click installation of the tool
  - Passwords are not persisted
  - Trouble-shooting support
  - Version check for the add-in and workbook files that are available in the [Employee Delta Export UI](#)
  - Request certain segments to be extracted and processed

## 2.2 Target Group

This guide is intended for the following target groups:

- SAP/SuccessFactors Professional Services and Consultants
- SAP/SuccessFactors Partners
- Customers

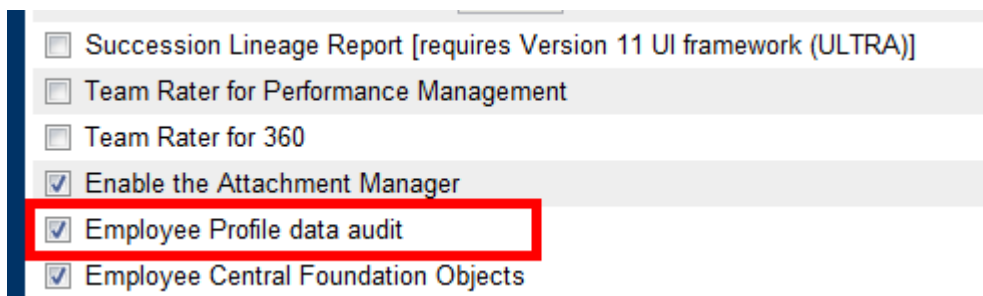
## 3 Business Context and Use

Use Employee Delta Export for Microsoft Excel to extract changes made to employee master data in the SuccessFactors HCM Suite.

### 3.1 Prerequisites

The *Employee Delta Export for Excel* has the following prerequisites:

- One of the following Microsoft Excel for Windows versions
  - Microsoft Excel 2010 (32-bit)
  - Microsoft Excel 2010 (64-bit)
  - Microsoft Excel 2013 (32-bit)
  - Microsoft Excel 2013 (64-bit)
- One of the following Windows versions
  - Windows 7 32-bit or 64-bit
  - Windows 8 32-bit or 64-bit
- Your company policy must allow the execution of macros in Microsoft Excel
- Microsoft XML Core Services version 6 (MSXML 6.0) or newer must be installed on the computer.
- SuccessFactors Employee Central release b1402 or newer
- *Employee Profile data audit* must be turned on in Provisioning.



- Web Service setup for SF APIs
  - See section *Enablement of the Application Programming Interface (API) of the Employee Central Payroll Implementation and Integration Handbook*.
  - Find the most current version of this Handbook on the [SAP Service Marketplace](#) under *Cloud Solutions from SAP > SuccessFactors > Employee Central*



## 3.2 Limitations

### 3.2.1 Limitations caused by the backend

- Role-based permissions are not supported. This means you need to grant the permission [Employee Central HRIS SOAP API](#) to the user who uses Employee Delta Export. This will allow the user to retrieve all kinds of data that is accessible using SOAP APIs.
- [Employee Delta Export](#) can only handle the following entities from the backend
  - Person
  - Personal Information
  - Personal Information Global
  - Address Information
  - Employment Information
  - Job Information
  - Compensation Information
  - Pay Compensation Recurring
  - Pay Compensation Non Recurring
  - Direct Deposit (see Note)
  - PaymentInformationV3 (see Note)
  - National ID Card
  - Email
  - Phone
  - Person Relation
  - Job Relation
  - Global Assignment Information

#### Note

Depending on settings made for Payment Information V3 in Provisioning, only either Direct Deposit or Payment Information V3 can be requested. For more information see [Using Payment Information \[page 58\]](#)

The following segments are not included in the standard query. You can request to have these segments included as result sheets, for more information see section [Using the Requested Segments Table \[page 58\]](#).

- Deduction Recurring
- One Time Deduction
- Alternative Cost Distribution
- Personal Documents Information
- [Employee Delta Export](#) doesn't consider an employee changed when other entities than the above have changed (for example, foundation objects used for lookup purposes)
- By default, the workbook shows all future time slices of an entity that have been changed, meaning changes that become effective after the payroll period can't be excluded. However, if you use [Period-based Delta](#) you can determine the start and end date of the validity period for which you want time slices to be considered.

---

For more information on [Period-based Delta](#) see section [Period-based Delta \[page 35\]](#).

- Since [Employee Delta Export](#) is primarily based on the Compound Employee API it inherits most of the limitations of the Compound API which are:
  - Event and event reason on Compensation Information is not supported
  - Only filter parameters supported by the Compound Employee API are passed on to the backend. Other filters are applied locally in Excel. This may lead to unexpected high data volumes being retrieved from the backend.
  - Picklists and Foundation Objects assigned to fields returned by the API need to have external codes. Otherwise, no value is returned for those fields.
  - Severe data inconsistencies (for example time slice mismatches) may prevent employees from being shown. They will appear on the [Errors](#) sheet instead.

## Related Information

[Understanding Delta Calculation \[page 19\]](#)

[Using Payment Information \[page 58\]](#)

[Period-based Delta \[page 35\]](#)

## 3.2.2 Limitations caused by the client application

- For performance reasons, [Employee Delta Export](#) should not be used to process more than 1000 employees. Depending on the data volume of the processed employees, the maximum number of processed employees can be substantially smaller. In such cases the data needs to be broken down into smaller parts that can be handled by Excel.
- Initial load scenarios that extract the whole change history for every employee of a target population are not supported. A delta extraction must have a report start date.
- No localization is provided for:
  - Workbook static content (text, messages)
  - Employee data (descriptions)
  - Foundation data (for example, pay component names)
  - Picklist entries (for example, [married](#), [single](#), [divorced](#))
- When you plan to divide your overall employee population into different pieces processed by different processors using different filter criteria, this needs to be planned very thoroughly to prevent loss of information when employees transfer from one sub-population to another.
- You cannot use lists of values for filter parameters in arbitrary combinations.
- Sorting must be done locally in Excel.
- Single sign-on is not supported.
- [Employee Delta Export](#) makes it possible to manually extract data from a system. Even though it is possible to have an automated process read data from the processed workbook, Employee Delta Export doesn't provide a functionality to automatically start the export.

## 3.3 Security and Session Management

### Context

The application calls the Compound Employee API to extract data from the backend. When you trigger the extraction a login dialog pops up asking for user and password for the selected system. Authentication is established through the login operation using user and password. Single Sign-on is not supported for the API calls. A successful login will return a session ID as an HTTP Cookie. This cookie is passed back to all subsequent HTTP Requests that invoke API operations in order to authenticate. When all required API calls have been executed, a logout operation is sent to the backend that closes the session.

The session will also timeout after 10 minutes of inactivity, for example in case of an error that is not properly handled.

The SuccessFactors HCM Suite has the ability to restrict a user account used for API access to certain IP addresses. This also applies to the API used by the Excel application. Proceed as follows:

### Procedure

1. Go to [Admin Tools](#) > [Password & Login Policy Settings](#)
2. Choose the link [Set API login exceptions](#)
3. Choose the [Add](#) button
4. Specify Username, Maximum password age = 0, and your IP addresses

#### Admin Tools

[Back to Admin Tools](#)

#### Password & Login Policy Settings : App

Use this page to set the Password Policy.

Minimum Length

Maximum Length

Maximum Password Age (in days)

Set to -1 keep passwords from expiring (not recommended)

[Set API login exceptions...](#)

Apply different maximum password age for the following users

[Add](#)

Items per page: 10

Displaying 0 records

Create New User Security Setting - Google Chrome

localhost:8080/singleStep?step\_name=generic&stepFB=editUserSecurity&fbStepUserSeci

#### Create New User Security Setting

Use this page to create a new user security setting

Username: *	<input type="text"/>
Maximum password age(days): *	<input type="text"/>
	Set to -1 keep password not expiring (not recommended)
IP address restrictions: *	Apply the maximum password age only when user is access address(es) (for example 10.20.30.41, 10.20.30.40-10.20.30.5)
	<input type="text"/>

## 3.4 Role-based permissions

The following permissions are required to extract data from the backend using the Compound Employee API:

- Manage System Properties
  - Picklist Management and Picklists Mappings Set Up
- General User Permission
  - SFAPI User Login
- Employee Central API
  - Employee Central Foundation SOAP API
  - Employee Central HRIS SOAP API
  - Employee Central Foundation SOAP AP
  - Employee Central HRIS OData API
- Manage Integration Tools
  - Admin access to OData API

The following permission is required to access the [Employee Delta Export UI](#):

- Manage Employee Delta Exporting Templates

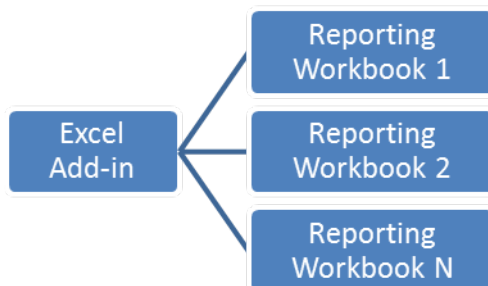
You can set this permission in one of the following permission groups:

- Manage Dashboards / Reports
- Manage User

After setting the permission, logout and login again. You can now find [Employee Delta Export](#) link in Admin Center using the [Admin Tools](#) search field.

## 3.5 Application Components

Employee Delta Export consists of the following components:



There is one Excel add-in that contains the logic to retrieve the data from the backend and perform delta calculation. The add-in adds functions to the Excel Ribbon that allow you to fill one of the reporting workbooks with data fetched from the backend. You can have an arbitrary number of reporting workbooks. The workbook provides the layout for the data to be shown plus configuration information used by the add-in.

## 3.6 Employee Delta Export UI

On the Employee Delta Export UI you can find the Excel Add-in Installer, the sample workbooks, and this user guide for download.

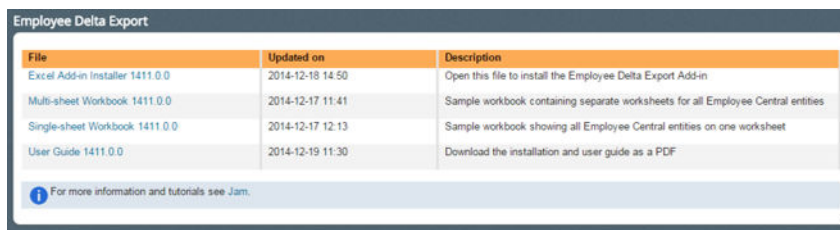
Find the Employee Delta Export UI in Admin Tools under [Employee Files](#) or [Reporting](#).

### Note


You need the [Manage Delta Exporting Templates](#) permission to access the [Employee Delta Export UI](#).

The Employee Delta Export UI has a list of files you can download:

- Excel Add-in Installer (CompEmpDeltaAddIn.xlsm): Use this to install the add-in.
- Multi-sheet Workbook (CompEmpDelta\_multiSheet.xlsx): Contains separate worksheets for all Employee Central entities.
- Single-sheet Workbook (CompEmpDelta\_singleSheet.xlsx): A sample workbook showing all Employee Central entities on one single worksheet.
- User Guide (Employee Delta Excel.pdf): The current *Employee Central Compound Employee API - Employee Delta Export Add-in for Microsoft Excel* guide.



File	Updated on	Description
Excel Add-in Installer 1411.0.0	2014-12-18 14:50	Open this file to install the Employee Delta Export Add-in
Multi-sheet Workbook 1411.0.0	2014-12-17 11:41	Sample workbook containing separate worksheets for all Employee Central entities
Single-sheet Workbook 1411.0.0	2014-12-17 12:13	Sample workbook showing all Employee Central entities on one worksheet
User Guide 1411.0.0	2014-12-19 11:30	Download the installation and user guide as a PDF

 For more information and tutorials see [Jam](#).

Each file has its own version consisting of three numbers. The first number indicates the SuccessFactors build for which the file is provided. Only the files relevant to your SuccessFactors release are listed. The second number represents the patch version of the file. The add-in performs a version check each time you retrieve data from the Excel and give a warning if the add-in or a workbook is outdated. The third number indicates the revision of the file and represents minor changes. You are not forced to update files that have an outdated revision number.

## 3.7 Version Check and Updating the Add-in

Whenever you execute a query request by clicking [Get Data](#) a version check is performed. If there is a version of the add-in with a higher release number or patch version than the file you have installed, you will get the warning: *Your version is outdated and may cause problems. We strongly recommend you download the latest version of the Employee Delta Export Add-in from Admin tools.*

If only the revision number of your installed add-in differs from that of the newest version available, you get a different warning: *A new version of the Employee Delta Export Add-in is available. Please download it from Admin Tools to benefit from the latest features. You can continue working in the older version and download a new version later, if desired.*

If you check the [Do not show this message again](#) check box, the warning won't show up again as long as the revision number of the add-in file in the Employee Delta Export UI isn't changed. Revision changes are minor changes that won't influence existing features. Therefore, updating is optional.

The compatibility of the add-in and the workbook is also checked.

## 3.8 Installation

The Excel Add-in must be installed on your local computer. In order to do so, download the Excel Add-in Installer (CompEmpDeltaAddIn.xlsm) from the [Employee Delta Export UI](#) in Admin Tools. After downloading the installer, open it and follow the instructions shown on the pop-up. Make sure that macro execution is enabled in Microsoft Excel and the workbook is made a trusted document.

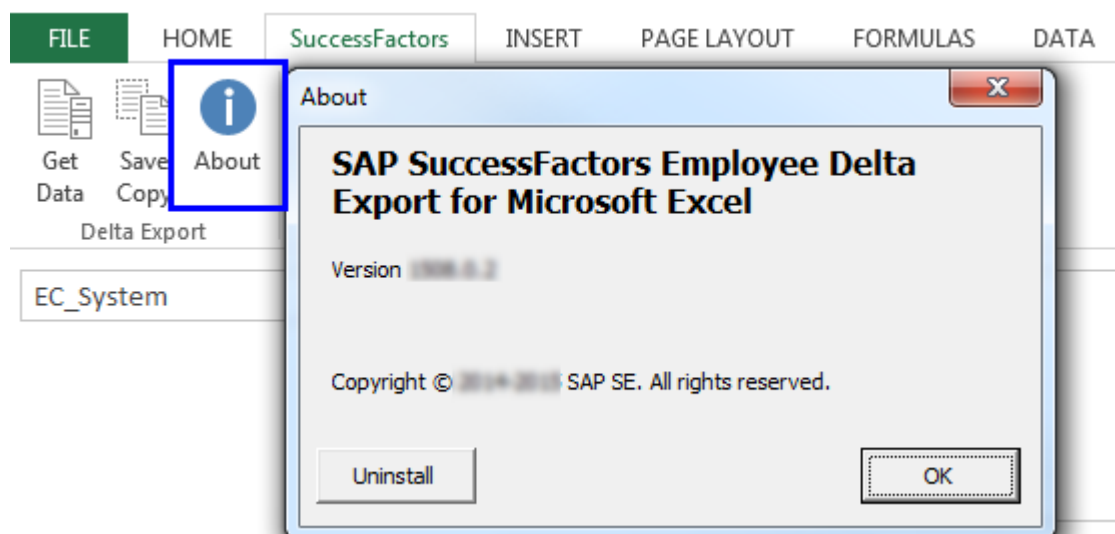
## Related Information

[Role-based permissions \[page 12\]](#)

[Employee Delta Export UI \[page 13\]](#)

## 3.9 Uninstalling the add-in

You can uninstall the Excel add-in by choosing [About](#) from the SuccessFactors ribbon tab and choosing [Uninstall](#) in the upcoming popup window.





## 4 Working with Employee Delta Export

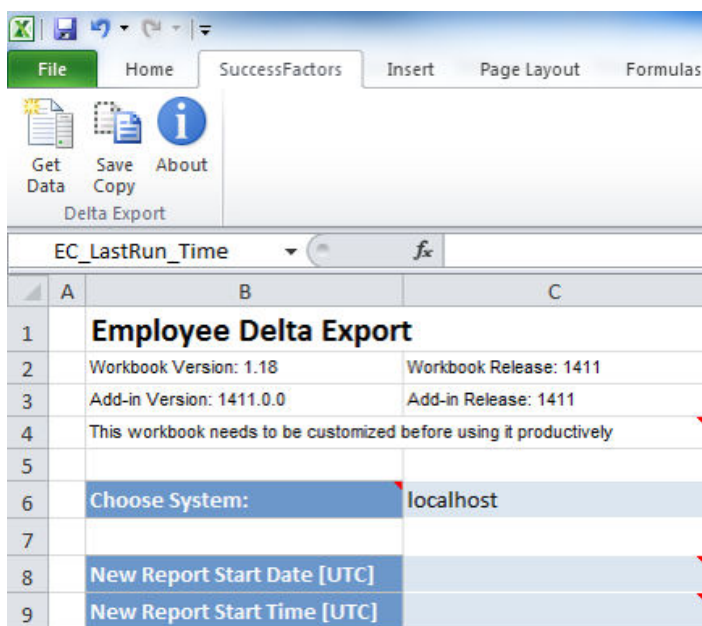
### 4.1 Quick Start

#### Context

This section describes the first steps you need to take to retrieve data using Employee Delta Export.

#### Procedure

1. Install the add-in as described in section [Installation \[page 14\]](#).
2. Download the [Multi-sheet Workbook](#) from the Employee Delta Export UI.
3. Open the [Reporting](#) workbook.
4. Go to the [Systems](#) worksheet and enter the system you want to retrieve data from
  - Name: free text to identify the system
  - URL: enter the relevant URL. For more information see section [Finding your URL](#)
  - Company: the name of the company (case sensitive)
  - User: Login user (if you leave this empty, you will get a pop-up later)
5. Go to the [Selection](#) worksheet.
6. Choose the desired system.



7. Set the other input fields as follows

New Report Start Date [UTC]	01/01/2014
New Report Start Time [UTC]	00:00:00

- Go to the SuccessFactors tab on the Excel ribbon and choose Get Data. This calls the Compound Employee API in the SuccessFactors backend returning all changes that happened between 01.01.2014 and the current time.
- Once the data is read, the view switches to the Summary sheet showing you the selection parameters and the number of records found.



## Related Information

[Finding your URL \[page 16\]](#)

### 4.1.1 Finding your URL

Your API endpoint will depend on where your SuccessFactors instance is located. It can be in one of several data centers. The endpoints by data center are listed below.

Region	Data Center	DC#	Site	URLs
EMEA	Amsterdam (ASM-2)	2	EU Production (Premium)	<a href="https://api.successfactors.eu">https://api.successfactors.eu</a> ➡ <a href="https://api2.successfactors.eu">https://api2.successfactors.eu</a> ➡  (Both <a href="https://api.successfactors.eu">https://api.successfactors.eu</a> ➡ and <a href="https://api2.successfactors.eu">https://api2.successfactors.eu</a> ➡ resolve to the same servers)
US	Arizona	4	AZ Production (Standard)	<a href="https://api4.successfactors.com">https://api4.successfactors.com</a> ➡
EMEA	Amsterdam (ASM-5)	5	EU Production (Standard)	<a href="https://api5.successfactors.eu">https://api5.successfactors.eu</a> ➡
US	Ashburn	8	Ashburn Production (Premium)	<a href="https://api8.successfactors.com">https://api8.successfactors.com</a> ➡

Region	Data Center	DC#	Site	URLs
APJ	Sydney	10	Sydney Production (Premium)	<a href="https://api10.successfactors.com">https://api10.successfactors.com</a> 
EMEA	Rot	12	Rot Production (Standard)	<a href="https://api012.successfactors.eu">https://api012.successfactors.eu</a> 

## 4.1.2 Example test run

### Procedure

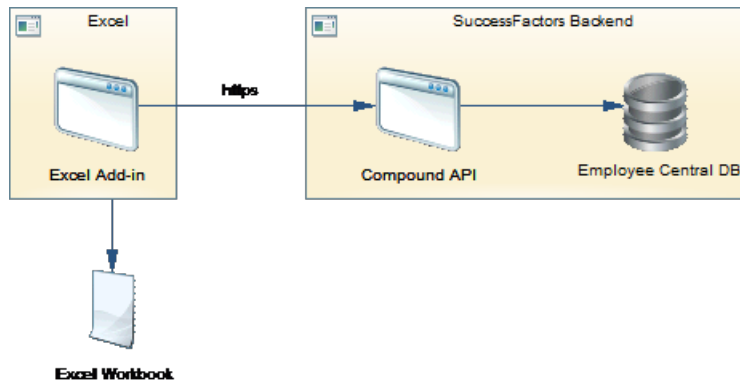
- Check the data sheets in the workbook for new content
- Go to the Employee Central UI and change the address of an employee (for example, change the house number)
- Press the [Get Data](#) button. This time, all changes between 05.09.2014 07:04:17 and the current time are retrieved.
- Check the data sheet for changes and you will see exactly the one address change you keyed in just before.

## 4.2 Architecture Concept

### 4.2.1 Component Structure

The following components are involved in employee delta reporting:

- Microsoft Excel Application
- Excel Add-in
- Excel reporting workbooks containing configuration information
- Compound Employee API
- SuccessFactors Business Suite Database



## Excel Add-in

The Excel add-in is the heart of Employee Delta Export. It contains VBA code which performs the data extraction and delta calculation. The add-in must be installed into the Excel application once and becomes an intrinsic component of Microsoft Excel. However, it remains inactive unless a reporting workbook is opened. The VBA code calls the Compound Employee API to receive the information about new and changed employees. In addition, it uses other direct and OData APIs to retrieve lookup data.

## Reporting Workbooks

The reporting workbooks contain the result sheets that receive the data retrieved from the backend plus configuration data controlling the extraction process.

## Compound Employee API

The Compound Employee API is a SOAP API that extracts the data from the backend. It is called by the VBA code of the add-in multiple times.

## SuccessFactors Business Suite Database

This backend component keeps all the employee data plus historical data about changes. It is accessed by the Compound Employee API.

---

## 4.2.2 Data Extraction

Employee Delta Export is mainly based on the Compound Employee API in the backend, which is a SOAP API. This means that the result returned by the API is XML. This is the reason why many configurations in the reporting workbooks are based on XPath syntax, since XPath is the perfect language to access elements in an XML document.

The Employee Delta Export application calls the Compound Employee API in delta mode to retrieve all creations, changes, and deletions of entities that happened since the start of the evaluation period which is typically the time of the previous extraction. Additional selection parameters going into the WHERE clause of the API call can be specified on the [Selection](#) sheet. In case the first call in delta mode did not retrieve all records matching the selection criteria, subsequent queryMore calls are issued to retrieve the remaining data.

Depending on the version of Employee Delta Export, subsequent calls to the Compound API are performed. For the persons returned by the call in delta mode the API is called again in the so-called snapshot mode. This mode returns the status of all entities for these persons at the time of the previous extraction. After that, a third API call in full transmission mode is performed returning the current status of the persons in question.

The raw data returned by all API calls are stored in the workbook on hidden sheets.

In a second phase the client-side filters defined on the [Config](#) sheet are applied to the result XML returned from the backend in order to exclude more employees from the result. Both filters are applied to the full data of each employee retrieved from the backend including all time slices (filters are not applied to the delta result).

In a third phase the add-in post-processes the data and calculates the changes compared to the previous extraction on field level. Only changed fields are shown on the result sheets of the workbook unless it is enforced to show the current field values.

## 4.2.3 Understanding Delta Calculation

This chapter explains in more detail which results Employee Delta Export returns. This will be helpful to you when you are setting up business processes that post-process the Excel content.

Employee Delta Export returns all changes that have been keyed in to the backend or modified by APIs during the evaluation period of the report. The start of the evaluation period is as you specify in the fields [New Report Start Date \[UTC\]](#) and [New Report Start Time \[UTC\]](#) on the Selection sheet. The end of the evaluation period is the point in time when the report is executed. The report returns all changes within this period no matter when they become effective.

First of all, you need to distinguish between data that is effective dated and data that is not effective dated in Employee Central. Both types of data appear in the Excel reporting workbooks. For example, personal information and job information are effective dated, whereas national ID card information and spot bonus information aren't.

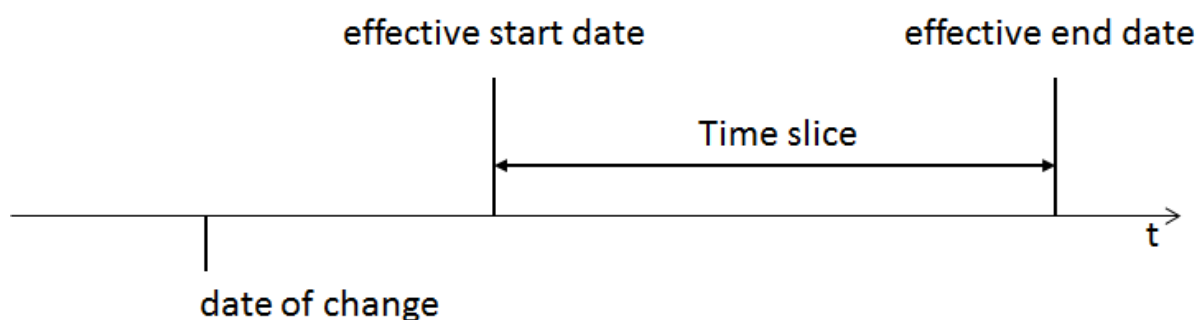
**Non-effective dated information** is easy to understand. Creations, changes, and deletions of such data can't be scheduled and become effective immediately for the whole time bar. Spot bonus (pay component non recurring) is somewhat of an exception, because this is information that applies for one particular date only.

**Effective dated information** may be a bit more difficult to understand. For effective dated information you need to distinguish between the point in time when a change is keyed into the backend and the point in time

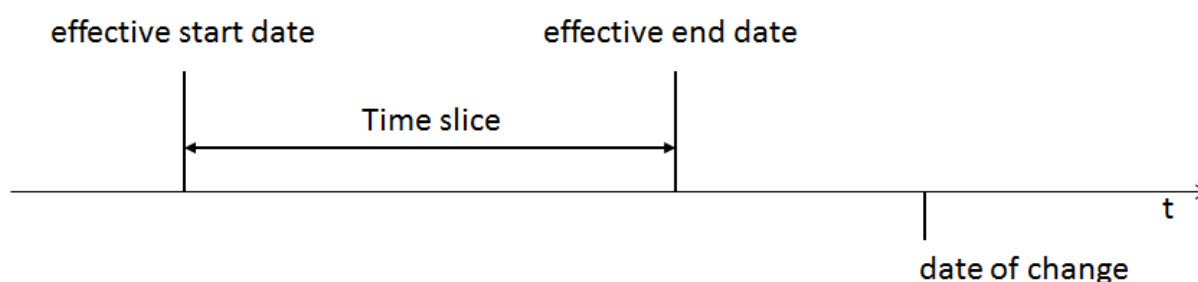
when this change becomes effective. Both points in time are completely independent of each other. A change keyed in today can become effective today, in the future, or in the past. Changes that become effective in the past are called retroactive changes. If a change becomes effective in the past, it may no longer be effective today since the period of effectiveness could also end in the past. The time period during which a set of data is effective is called a time slice. The date when a change becomes effective is called the start date (the beginning of the time slice) and the date when it ends being effective is called the end date (the end of the time slice).

Employee Delta Export returns data that has been keyed in during the evaluation period of the report no matter when these changes become effective. This is important to know because in some cases you might get information that you might not be interested in at the point in time you get it. For example, if the promotion of an employee is keyed in today, that, however, becomes effective at the start of next year, you will see this change in the next Excel report run after the change was made. You need to take this up organizationally, for example by shelving this change.

#### Future Dated change:



#### Retroactive change:



From a business perspective, changes shown in the report can result from the following actions performed in the backend:

- Hire of an employee
- Insertion of a new record for non effective-dated data
- Change of data of a non effective-dated entity
- Change of the key of an entity
- Insertion of a new time slice for effective-dated data
- Change of data in the history of effective-dated data using Make correction
- Change of start date of effective-dated data in the history using Make Correction
- Deletion of a non effective-dated record (for example personal information)



- Deletion of a record in the history
- Deletion of a record starting from a particular date (for example home address)
- Deletion of a record in the course of changing the superordinate

Below, you can see how these changes are shown in the report with examples.

### **i** Note

The highlighting shown in the examples here may differ slightly from the highlighting in the version of the add-in you have installed. For more information on how highlighting is used to determine changes, see section [Highlighting Changes in the Result Sheets \[page 29\]](#).

## Hiring an employee

	A	B	L	M	X	Y
4	Personal Information					
5	Status	User Id	Last Name [New]	Last Name [Old]	Marital Status [New]	Marital Status [Old]
6	New	20010	Kendrick		S	
7	New	20030	Scott		D	
8	New		Scott		M	
9	New	20040	Anderson		D	
10	New	20050	Walker		S	

A hire can be a new hire or a rehire. Both are shown in the same way so that the complete data of all entities and all time slices are shown in the report. In this case, the status columns have the value *New*. The [Old] columns stay empty.

## Inserting a new record for non effective-dated data

New records of non effective-dated data (for example national ID card) are shown as new rows in the Excel report with value *NewSeg* in the status column. The [Old] columns stay empty.

	A	B	D	E	F	G	H	I	J
4	National ID Information								
5	Status	User Id	First Name	Last Name	Country	National ID Type [New]	National ID Type [Old]	National ID [New]	National ID [Old]
6	Changed	20030	John	Scott	USA	ssn		235-20-0030	234-20-0030
7	NewSeg	20050	Patricia	Walker	CAN	sin		123-654-234	
8	New	20120	Robert	Lopez	USA	ssn		234-20-0120	
9	New	20130	Michael	Lee	USA	ssn		234-20-0130	
10	New	20140	Elizabeth	Mitchell	USA	ssn		234-20-0140	

## Changing non effective-dated data

A simple change of data of non effective-dated data (for example national ID card) is shown as new rows in the Excel report with the value *Changed* in the status column. The [New] columns show the latest values whereas the [Old] columns show the values valid before the change happened.

## Changing the key of an entity

If a key field of a non effective-dated entity or effective-dated entity is changed in the backend (for example the country of a national ID card) this leads to the deletion of the old record with the old key and the creation of a new record with the new key. The Excel report will show a row with status *Deleted* for the first and a row with status *NewSeg* for the second action. Rows 7 and 8 of the example below show such a case.

	A	B	D	E	F	G	H	I	J
4	National ID Information								
5	Status	User Id	First Name	Last Name	Country	National ID Type [New]	National ID Type [Old]	National ID [New]	National ID [Old]
6	Changed	20030	John	Scott	USA	ssn		235-20-0030	234-20-0030
7	NewSeg	20050	Patricia	Walker	CAN	sin		123-654-234	
8	Deleted				USA	ssn		***deleted**	234-20-0320
9	New	20120	Robert	Lopez	USA	ssn		234-20-0120	

## Inserting a new time slice for effective-dated data

If you insert a new time slice in the backend using the *Insert Record* function, this is shown in the Excel report as a separate row with the value *Changed* in the status column. The [Old] columns show the values that were valid for the validity period before the new time slice was inserted. A special variant of an insertion where a subordinate entity is deleted during the change is discussed below. Row 6 in the example above shows such a case.

## Changing data in the history using Make Correction

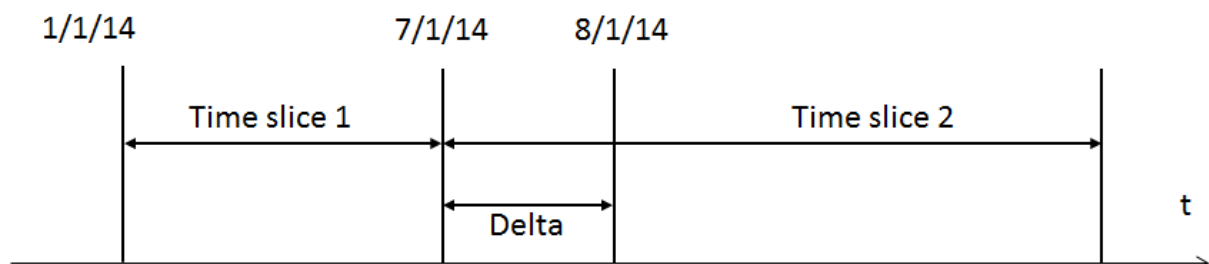
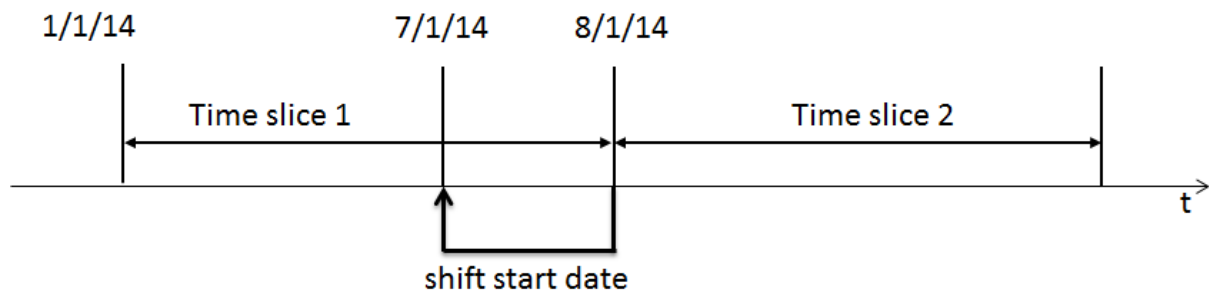
This kind of change is shown in the Excel report just like the insertion of a new time slice. The [Old] columns show the values that were valid before the correction was performed. Row 6 of the example below shows such a case.

	A	B	D	E	F	G	J	K	L	M
4	Employee Address									
5	Status	User Id	First Name	Last Name	Start Date	End Date	Address Type	Country	Address1 [New]	Address1 [Old]
6	Changed	20030	John	Scott	01.09.2014	31.12.9999	Home		66 Greenway Park Rd	1084 Foster City Blvd
7	NewSeg	20040	Kevin	Anderson	01.09.2014	31.12.9999	Home	USA	66 Greenway Park Rd	
8	Changed	20050	Patricia	Walker	11.07.2014	17.07.2014	Home		3541 Wheeler Rd	3540 Wheeler Rd
9	Changed				18.07.2014	31.12.9999	Home		211 Chambers St	3540 Wheeler Rd
10	Changed	20060	Kimberly	Smith	18.07.2014	17.08.2014	Home		664 Scranton Rd	1111 US Highway 441 N
11	Changed				18.08.2014	31.12.9999	Home		664 Scranton Rd	1111 US Highway 441 N
12	Deleted	20070	Donna	Evans	11.07.2014	30.12.9999	Home	USA	deleted	117 Kite Rd
13	New	20120	Robert	Lopez	11.07.2014	31.12.9999	Home	USA	1234 Creekside Place	

## Changing the start date of effective-dated data in the past using Make Correction

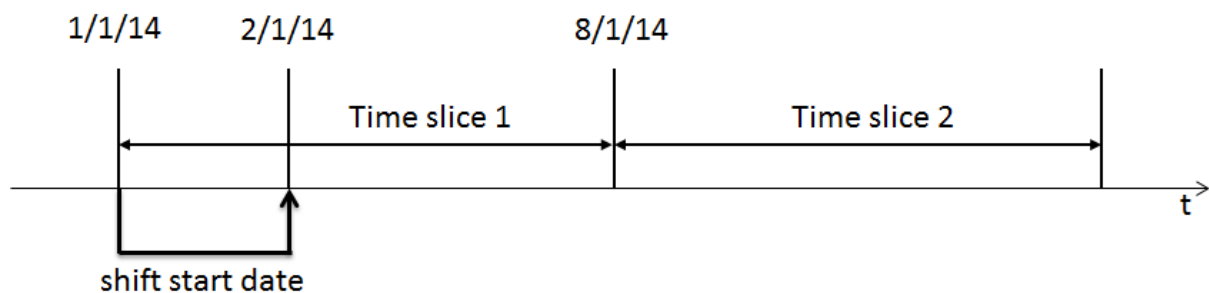
This is the most complex scenario. If the start date of a time slice is shifted to the future or to the past, this is shown in the Excel report as a separate row that shows the differences between the old and the new values for the period that is affected by the shift.

In the graphics below, the start date of Time Slice 2 is changed from 8/1/14 to 7/1/14. As a result, the Excel report shows a new row with start date 7/1/14 and end date 7/31/14 (denoted as Delta in the figure) which shows the values of Time Slice 1 in the [New] columns and the values of Time Slice 1 in the [Old] columns.

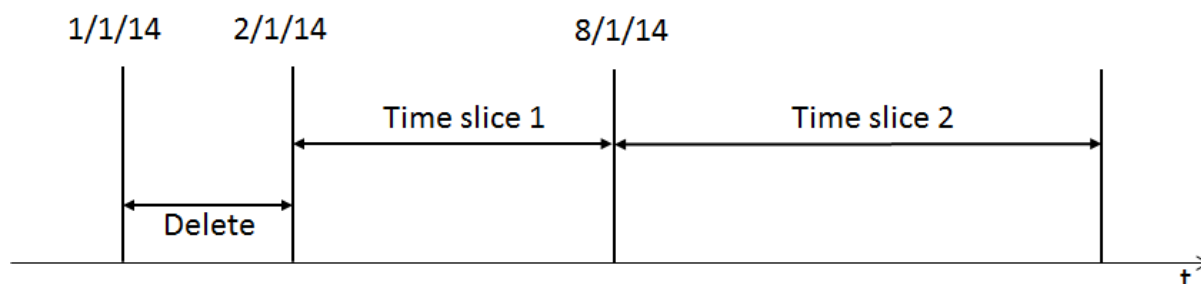


A border case in this category is when the start date of the first time slice is moved to the future. In this case, the Excel reports shows a row denoting a deletion of the delta period.

	A	B	D	E	F	G	L	M	N
4	Employee Address								
5	Status	User Id	First Name	Last Name	Effective As Of (Start Date)	End Date	Country [New]	Country [Old]	Address1 [New]
6	Deleted	20070	Donna	Evans	11.07.2014	30.12.9999	***deleted***	USA	***deleted***



The result of the shift is shown here:



## Deleting a non effective-dated record (for example National ID Card information)

When you delete a non effective-dated record in the backend, this is shown in the Excel report as new rows with the value *Deleted* in the status column. The [Old] columns don't change, in this case. The [New] columns contain the pseudo value *deleted*. Row 8 in the following example shows such a case.

	A	B	D	E	F	G	H	I	J
4	National ID Information								
5	Status	User Id	First Name	Last Name	Country	National ID Type [New]	National ID Type [Old]	National ID [New]	National ID [Old]
6	Changed	20030	John	Scott	USA	ssn		235-20-0030	234-20-0030
7	NewSeg	20050	Patricia	Walker	CAN	sin		123-654-234	
8	Deleted				USA	ssn		***deleted**	234-20-0320
9	New	20120	Robert	Lopez	USA	ssn		234-20-0120	

## Deleting a record in the history

When you delete an effective-dated record in the backend, this is shown in the Excel report, as in the example before (*Deleting a non effective-dated record*).

## Deleting an address as of a specific date

Assume you choose Edit in the address portlet and after entering the effective start date you delete for example the vacation. In the Excel report this will be shown as a new address row with value Deleted in the status column. The [Old] columns show the values just before the deletion. The [New] columns contain the pseudo value deleted. Row 12 of the example above shows such a case.

## Deleting a record in the course of changing the superordinate data

A typical example for this is you create a new compensation time slice with the start date X and in the course of that you delete a pay component from the list. In the Excel report this is shown as a pay component row with value *Deleted* in the status column. The [Old] columns show the values just before the deletion. The [New] columns contain the pseudo value deleted. Row 9 in the example below shows such a deleted pay component.

	A	B	C	D	F	G	H	K	L	M	N	O	P	
4	Pay Component Recurring Information													
5	Status	User Id	First Name	Last Name	Pay Component	Start Date	End Date	Currency [New]	Currency [Old]	Frequency [New]	Frequency [Old]	Amount [New]	Amount [Old]	Last
6	Changed	20220	Thomas	Baker	Base Salary	01.09.2014	31.12.2014	USD	USD	MON	MON	5000,00	5500,00	01.0
7	NewSeg				Monthly Car	01.09.2014	31.12.2014	USD		MON		200,00		01.0
8	Changed				Base Salary	01.01.2015	31.12.9999	USD	USD	MON	MON	5000,00	5500,00	30.0
9	Deleted	20130	Michael	Lee	Monthly Car	01.11.2014	31.12.9999	***deleted	USD	***deleted*	MON	***deleted*	200,00	01.0
10	New	20140	Elizabeth	Mitchell	Bonus	11.07.2014	31.12.9999	USD		MON		20,00		01.0

## Related Information

[Highlighting Changes in the Result Sheets \[page 29\]](#)

## 4.3 Workbooks

*Employee Delta Export* requires specific Excel workbooks to fill in the data. You can download the workbook files from the *Employee Delta Export UI*. Such a workbook has a specific structure and contains configuration information that controls the behavior of the add-in. SAP delivers two different workbook templates you can use as a starting point for customization:

- **CompEmpDelta\_multiSheet.xlsx**  
Contains one sheet for each entity that is supported by the application. Each sheet shows new and changed data in combination.
- **CompEmpDelta\_singleSheet.xlsx**  
Contains three data sheets,
  - New employees (hired, re-hired, or transferred employees)
  - Changed employee data
  - Leaving employees

## 4.4 Worksheets

### 4.4.1 Result Sheets

The result sheets are populated by the data retrieved from the backend by the Compound Employee API. The two sample reporting workbooks [CompEmpDelta\\_singleSheet.xlsx](#) and [CompEmpDelta\\_multiSheet.xlsx](#) have different sets of result sheets. In both cases the result sheets have a common layout in the first 4 columns:

Column 1: Status reflecting the reason why the row appears in the workbook

Column 2: User ID of the employee

Column 3: First name of the employee no matter if it was changed

Column 4: Last name of the employee no matter if it was changed

The result sheets have two columns for data fields that may change. The column with suffix [New] contains the current value and the columns with suffix [Old] contains the value the field had at the start of the evaluation period. Note, however, that if the data field belongs to an effective-dated entity the [Old] value is subject to the start and end date in the same row.

#### CompEmpDelta\_singleSheet.xlsx

This sample workbook has different result sheets for hires, changes, and leaves. On each of the result sheets you find the data of all Employee Central entities supported by Employee Delta Export.

Sheet	Description
Hires	The <a href="#">Hires</a> sheet always shows the complete set of data for the hired or rehired employee. All time slices are shown, time slices in the future as well.



Sheet	Description
Changes	<p>The <a href="#">Changes</a> sheet shows employees who have been changed in the backend during the evaluation period. Only time slices that have been changed are shown in this case.</p> <p>Depending on the configuration parameter <a href="#">Show Unchanged Values</a> it shows all values of a changed entity or only those that have actually been modified. The status column distinguishes between the different events causing the change. It contains one of the following values:</p> <ul style="list-style-type: none"> <li>• Changed</li> </ul> <p>Indicates a change during the evaluation period. This can be caused by either a correction or by adding a new time slice.</p> <ul style="list-style-type: none"> <li>• NewSeg</li> </ul> <p>Indicates that the entity record was created during the evaluation period. For example, a new national ID card entry was added or a new pay compensation entry was added. NewSeg is not used, however, if a new time slice was added to an existing entry, like personal information. This case is portrayed as a change instead.</p> <ul style="list-style-type: none"> <li>• Deleted</li> </ul> <p>Indicates that the time slice or an entity record was deleted during the evaluation period.</p>
Leavers	<p>The <a href="#">Leavers</a> sheet shows only those employees for whom a leave has been keyed during the evaluation period. Only little data is shown for these employees.</p>

Each worksheet contains the data of all Employee Central entities related to the employee. Each entity is shown in a separate row. For effective-dated entities each time slice is shown in a separate row. Rows belonging to the same employee are grouped by color. The column [Start Date](#) in the column section for a particular effective-dated entity shows the start date of the time slice.

## CompEmpDelta\_multiSheet.xlsx

This sample reporting workbook contains separate result worksheets for all Employee Central entities supported by Employee Delta Export:

- Personal Information
- National ID Information
- Address Information
- Employment Details
- Job Information
- Pay Compensation Recurring

- Pay Compensation Non-Recurring (Spot Bonus)
- Direct Deposit
- Payment Information
- Deduction Recurring
- One Time Deduction
- Alternative Cost Distribution
- Personal Documents Information

Each sheet contains data for hires, rehires, changes, and leaves. For effective-dated entities each time slice is shown in a separate row. Rows belonging to the same employee are grouped by color.

The different events are distinguished by the status column. The status column contains one of the values below:

New	A hire or rehire has been entered during the evaluation period. The full set of data including all time slices is returned.
Changed	This can be caused by either a correction or by adding a new time slice.
NewSeg	A new entity record was created during the evaluation period. For example, a new national ID card entry was added or a new pay compensation entry was added. NewSeg is not used, however, if a new time slice was added to an existing entry like for example personal information. This case is denoted as a change instead.
Deleted	The time slice was deleted during the evaluation period
Transfer	The employee has entered the population specified by the selection criteria during the evaluation period. This means that the employee didn't fulfill the selection criteria at the start of the evaluation period. In this case, the full set of data including all time slices is.

	A	B	D	E	F	G	J	L	M	N	O	
4	Employee Address											
5	Status	User Id	First Name	Last Name	Start Date	End Date	Address Type	Country [New]	Country [Old]	Address1 [New]	Address1 [Old]	Ad [Ne
6	Changed	20030	John	Scott	01.09.2014	31.12.9999	Home			66 Greenway Park Rd	1084 Foster City Blvd	
7	NewSeg	20040	Kevin	Anderson	01.09.2014	31.12.9999	Home	USA		66 Greenway Park Rd		
8	Changed	20050	Patricia	Walker	11.07.2014	17.07.2014	Home			3541 Wheeler Rd	3540 Wheeler Rd	
9	Changed				18.07.2014	31.12.9999	Home			211 Chambers St	3540 Wheeler Rd	
10	Changed	20060	Kimberly	Smith	18.07.2014	17.08.2014	Home					
11	Changed				18.08.2014	31.12.9999	Home			664 Scranton Rd	1111 US Highway 441 N	
12	Deleted	20070	Donna	Evans	11.07.2014	30.12.9999	Home	deleted	USA	deleted	117 Kite Rd	
13	New	20120	Robert	Lopez	11.07.2014	31.12.9999	Home	USA		1234 Creekside Place		

## 4.4.1.1 Highlighting Changes in the Result Sheets

Employee Delta Export uses highlighting to make it easier for you to see what has changed.

- If changes are on the field level: one or more single cells are highlighted.
- If changes affect the whole record on a result sheet: the status cell is highlighted.

In this example, the whole national ID record was deleted, which you can see here because the status cell is highlighted. In order to identify the deleted record, the business key stays unchanged and fields that don't belong to the business key are shown as **\*\*\*deleted\*\*\***.

	A	B	C	D	E	F	G	H	I
4	National ID Information								
5	Status	User Id	First Name	Last Name	Country	National ID Type	National ID [New]	National ID [Old]	Is Primary [New]
6	Deleted	person_id_external	first_name	last_name	country	card_type	***deleted***	national_id	***deleted***

The following example shows another deletion, however, this time, only one value of the record was deleted. The changed value is highlighted.

	A	B	C	D	E	F	G	J	K
4	National ID Information								
5	Status	User Id	First Name	Last Name	Country	National ID Type [New]	National ID Type [Old]	Is Primary [New]	Is Primary [Old]
6	Changed	test01	first_name	last_name	USA	card_type		TRUE	FALSE

## 4.4.2 Selection Sheet

The [Selection](#) sheet contains parameters used to filter data. These are:

Choose System	The system to which the http request goes. The dropdown list shows the entries found on the Systems sheet.
New Report Start Date/Time [UTC]	<p>The start date/time of the evaluation period. All creations, changes, and deletions of entities applied in the backend since this point in time will go into the result sheets. Once the data has been retrieved from the backend completely, these two fields are automatically updated with the server time when you started the recent extraction. Unless you change these, this date/time is used for the next extraction run.</p> <p>When no changes are detected in the extraction run, the message <i>No Changes Found</i> is issued and the fields remain unchanged.</p>

Last Report Start Date/Time [UTC]	<p>This is the start date/time of the previous extraction run. It is automatically updated after the extraction is performed.</p> <p>When no changes are detected in the extraction run, the message <i>No Changes Found</i> is issued and the fields remain unchanged.</p> <p>This field is read-only.</p>
Last Report End Date/Time [UTC]	<p>This is the end date/time of the previous extraction. It is in fact the date/time when you pressed the <i>Get Data</i> button the last time (in UTC time zone). It is typically identical with the <i>New Report Start Date/Time</i> mentioned above.</p> <p>This field is read-only.</p>
Period Start Date	Marks the start of the validity period for which data is considered.
Period End Date	Marks the end of the validity period for which data is considered.
First Run in Period	Indicates whether this is the first report run in that respective period.

## Filter Parameters

The remaining parameters are used for filtering the employees when retrieving them from the backend or for post-processing. You can either specify a single value or a comma-separated list. For details please see the section Filter Logic .

Example:

The following parameter specification

Division	ABC
----------	-----

results in a where clause WHERE division = 'ABC' being sent to the API in the backend

The following parameter specification

Division	ABC, BCD
----------	----------

results in a where clause WHERE division IN ('ABC','BCD') being sent to the API in the backend

The following parameter specification

Division	ABC
Department	BCD

results in a where clause WHERE division = 'ABC' AND Department = 'BCD'.

The following parameter specification

Cost Center	ABC
-------------	-----

results in filtering for cost center ABC in job information during post-processing.

## Related Information

[Period-based Delta \[page 35\]](#)

### 4.4.2.1 Filter Logic

The [Selection](#) sheet contains multiple fields which are used for filtering employees. There are two different kinds of filter parameter:

1. Filter parameters that are passed through to the Compound Employee API call.

2. Filter parameters that are applied during post-processing in Excel locally.

In order to understand the result of the report you need to understand how the filtering works for both cases.

## Filter parameters passed through to the Compound Employee API

There is a limited set of filter parameters provided by the Compound Employee API. These are:

- BUSINESS\_UNIT
- COMPANY\_TERRITORY\_CODE
- COMPANY
- DEPARTMENT
- DIVISION
- EMPLOYEE\_CLASS
- LOCATION
- JOB\_CODE
- PAY\_GROUP
- PERSON\_ID
- PERSON\_ID\_EXTERNAL

These parameters are described in detail in the Employee Central Compound Employee API Reference Guide.

Find the most current version of this Handbook on the [SAP Service Marketplace](#) under *Cloud Solutions from SAP > SuccessFactors > Employee Central*

For these parameters you can either specify a single value or a comma-separated list of values on the Selection sheet. A single values results in the WHERE <field> = '<value>' condition passed to the API and a list results in a WHERE <field> IN ('value1', 'value2', ...) to be passed to the API.

If you specify values for multiple parameters on the Selection sheet the filter conditions are logically ANDed in the API call.

### Note

Selection in the Compound Employee API works as follows: an employee counts as selected if each sub-condition is met by at least one of the segments. This may be unexpected since the sub-conditions don't need to be matched in the same segment.

### Example

Assume you specify the following filter values

Department: abc

Division: xyz

Both fields are contained in the job information segment. These conditions are met by an employee who has a single job info time slice that matches both conditions but also by an employee who has two job time slices where one meets the department condition and the other meets the division condition.

Filter parameters belonging to this category have an Excel name along this pattern *EC\_Filter\_<parameter name>*, for example EC\_Filter\_Company.



## Filter parameters applied during post-processing

Filter parameters belonging to this category have an Excel name matching the pattern *EC\_FilterAtClient\_<parameter name>*, for example *EC\_FilterAtClient\_CostCenter*.

In order to make these kinds of parameters effective, you need to include the filters into the sections *Additional Filters (ANDed)* and *Additional Filters (ORed)* on the *Config* sheet. The logic for the section *Additional Filters (ANDed)* is the same as for the backend-related filter parameters. This means, filter expressions specified here and logically ANDed with the backend-related filters and they apply if any of the segments meets the filter condition. As an example, the cost center filter is considered in the *Additional Filters (ANDed)* section.

Filter expressions in section *Additional Filters (ORed)* are applied to the employee who passed the previous checks. An employee meets the set of conditions in this section if any of the conditions are met.

The filter expressions in both *Additional Filter* sections may be complex XPath expressions containing multiple filter parameters. This allows for filters that check complex conditions on the same segment and so overcome the restriction mentioned above.

### Note

There is a limitation for this category of filter parameters: you can use a comma-separated value list only with one parameter of a complex condition due to complexity reasons. An error will be raised if you ignore this limitation.

### Example 1

Assume you specify the following filter values:

Department: abc

Cost Center: xyz

This filters employees who have department abc in one job segment and cost center xyz in the same or a different job segment.

### Example 2

Assume you specify the following filter values:

Department: abc

Cost Center: xyz, uvw

This filters employees who have department abc in one job segment and cost center xyz or uvw in the same or a different job segment.

### Example 3

Assume you specify the following filter values:

Department: abc,def

Cost Center: xyz, uvw

This filters employees who have department abc or def in one job segment and cost center xyz or uvw in the same or a different job segment.

### Example 4

---

Assume you specify the following filter values, where Region is a *custom\_string1* of job info and the Excel name for the filter parameter in the *Selection* sheet is EC\_FilterAtClient\_Region.

Region: abc

Cost Center: xyz, uvw

Additional filter (ANDed): //job\_information[cost\_center='%EC\_FilterAtClient\_CostCenter%']

Additional filter (ANDed): //job\_information[custom\_string1='%EC\_FilterAtClient\_Region%']

This filters employees who have region abc in one job segment and cost center xyz or uvw in the same or a different job segment.

#### Example 5

Assume you specify the following filter values where Region is a *custom\_string1* of job information and the Excel name for the filter parameter in the *Selection* sheet is EC\_FilterAtClient\_Region.

Region: abc

Cost Center: xyz, uvw

Additional filter (ANDed): //job\_information[cost\_center='%EC\_FilterAtClient\_CostCenter%' AND custom\_string1='%EC\_FilterAtClient\_Region%']

This filters employees who have region abc in one job segment and cost center xyz or uvw in the job segment.

#### Example 6

Assume you specify the following filter values, where Region is a *custom\_string1* of job information and the Excel name for the filter parameter in the *Selection* sheet is EC\_FilterAtClient\_Region.

Region: abc,def

Cost Center: xyz,uvw

Additional filter (ANDed): //job\_information[cost\_center='%EC\_FilterAtClient\_CostCenter%' AND custom\_string1='%EC\_FilterAtClient\_Region%']

This combination of filter and filter values is invalid since both parameters in the filter expression are multi-valued.

#### Example 7

Assume you specify the following filter values, where Region is a *custom\_string1* of job info and the Excel name for the filter parameter in the *Selection* sheet is EC\_FilterAtClient\_Region.

Region: abc

Cost Center: xyz

Additional filter (ORed): //job\_information[cost\_center='%EC\_FilterAtClient\_CostCenter%']

Additional filter (ORed): //job\_information[custom\_string1='%EC\_FilterAtClient\_Region%']

This filters employees who have a job info segment with region abc or a job segment with cost center xyz or both.

## 4.4.2.2 Period-based Delta

With period-based delta you can restrict reporting to those changes to data that are relevant for a given evaluation period, for example the time period to be considered in a payroll run.

The following kinds of changes are considered:

- Changes to data where the validity is in the past or where validity overlaps with the given evaluation period.
- Changes that were made in the past and the start date lies in the given evaluation period.

### **i** Note

Changes to future-dated records are only reported if their validity period is inside the evaluation period.

The following fields show period-based delta information on the [Selection](#) and [Summary](#) sheets:

- [Period Start Date](#)  
Marks the start of the validity period for which data is considered.
- [Period End Date](#)  
Marks the end of the validity period for which data is considered.
- [First Run in Period](#)  
Indicates that this is the first report run in the respective period.

### **i** Note

Leave all three fields empty to run Employee Delta Export without period-based delta.

## Use Case - Preparing upcoming payroll runs

A typical use case for period-based delta is when you are using Employee Delta Export to prepare employee data for an upcoming payroll run, in which case you would use [Period Start Date](#) and [Period End Date](#) to mark the relevant start and end dates of the payroll run in question.

The payroll run is scheduled for June 25 and the data to be considered for this run is all data effective between June 1 and June 30.

On June 20, take a snap shot of the data in the Employee Central system that you want to update in the payroll system. In this example, the last snapshot was taken on May 20 at 9:30:40 UTC. Enter the following data on the [Selection](#) sheet:

New Report Start Date [UTC]	20.05.2015
New Report Start Time [UTC]	09:30:40
Period Start Date	01.06.2015
Period End Date	30.06.2015

First Run in Period	Yes
---------------------	-----

You execute this run on June 20, 2015 at 09:40:50 UTC and you want to perform a run for a second update of data in the payroll system for the same period. Enter the following data on the [Selection](#) sheet:

New Report Start Date [UTC]	21.06.2015
New Report Start Time [UTC]	09:40:50
Period Start Date	01.06.2015
Period End Date	30.06.2015
First Run in Period	No

Note that [First Run in Period](#) is set to [No](#) this time.

## 4.4.3 Summary Sheets

The Summary sheet gives a summary of the last extraction run. It contains:

- The period for which changes were considered
- Filter values that were used for the extraction
- The number of employees that fulfilled the specified filter criteria

## 4.4.4 Systems Sheet

The [Systems](#) sheet contains all backend systems you can connect to. The add-in connects to a SuccessFactors backend using https and retrieves data by calling the Compound Employee API and other APIs. You may connect to different backend systems, for example a test system and production system by choosing the desired system in the [Choose System](#) field on the [Selection](#) sheet. The systems on the dropdown list are specified on the [Systems](#) sheet, which is typically hidden. Maintain your favorite systems on this sheet. The columns [Name](#), [URL](#), and [Company](#) are mandatory. Column [User](#) is optional. If User is not entered a login dialog pops up when the [Get Data](#) button is pressed.

It is not recommended to specify the password on the [Systems](#) sheet in productive use.

## 4.4.5 Config Sheet

The Config sheet contains configuration parameters that control how the extraction is performed and how the results are displayed.

### Note

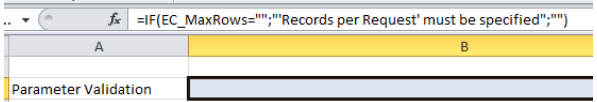
Some sections of this sheet are locked to prevent unintended changes. If you want to change these sections anyway you need to unlock the sheet on the Review tab.

### Note

Some of the parameters are Excel tables. Extending Excel tables needs to be done carefully. Make sure you don't enter data outside the table range. Otherwise, data may not be considered during processing.

Table 4: Parameters

Parameter	Description
Show Unchanged Values	This parameter controls whether unchanged values will be shown in the result sheets. As a standard, a value in a <i>New</i> column (for example <i>Address 1 [New]</i> ) is only then shown if it differs from the values in the corresponding <i>Old</i> column or if it is explicitly configured to always be shown. However, when this parameter is set to <i>True</i> , the <i>New / Old</i> pairs are always filled, if there are values in the backend.
Hire/Rehire Conditions	This field contains an XPath expression which is applied to the result of the Compound Employee API call in order to figure out whether an employee is rehired. This condition typically checks whether the event has changed in a certain way. If an employee is recognized as rehired, the full set of data is shown for this employee - not just the changed data.
File Name Pattern	This cell contains a formula that calculates the file name under which the workbook is saved when you use the <i>Save Copy</i> button on the <i>SuccessFactors</i> tab on the Excel ribbon. You may include any of the selection parameters in the file name. In addition, the placeholders <i>%path%</i> and <i>%date%</i> may be used. <i>%path%</i> is replaced by the folder in which the active workbook resides. <i>%date%</i> is replaced by the current date/time in the format YYYY-MM-DD_HH-MM-SS.

Parameter	Description
Parameter Validation	<p>This cell contains a formula that checks if the parameters in the <a href="#">Selection</a> sheet are consistent. The report will not start unless this formula returns blank. In the case of inconsistencies, the formula will return an error message that is presented to the user when extraction is started. As an example the sample workbook contain the formula shown in the graphic below, which checks if the <a href="#">Records per Request</a> parameter is filled. If not, the message '<a href="#">Records per Request</a>' must be specified is brought up and the report is aborted.</p>  <p>In a customer scenario the formula may be used to enforce that one or more filter values are specified on the <a href="#">Selection</a> sheet.</p>
Update New Report Start Date	<p>This parameter controls whether the <a href="#">New Report Start Date/Time</a> fields on the <a href="#">Selection</a> sheet are updated automatically after an extraction run. The standard setting for this parameter is <a href="#">Yes</a>. This ensures that the user running the report doesn't need to worry about manually updating the extraction start date/time. The user simply needs to press the <a href="#">Get Data</a> button to get the changes that have taken place in the backend since the last extraction run.</p> <p>However, there are cases where it makes sense to set the parameter to <a href="#">No</a>. For example when you want to test the extraction with the same base date/time over and over again.</p>
Records per request	<p>The number of records to be retrieved with one API call. If there are more records that meet the selection parameter, subsequent queryMore requests are sent to retrieve more records. The greater this parameter is, the less overhead is caused, however, the longer it takes to process one chunk of data.</p> <p><b>Recommendation</b></p> <p>200 records per request.</p>
QueryMore Iterations	<p>Maximum number of queryMore calls.</p> <p>Set this parameter to <a href="#">Unlimited</a> in production environments.</p>
Use PaymentInformationV3	<p>Set this parameter to <a href="#">Yes</a> to enable Payment InformationV3 for the single-sheet workbook instead of Direct Deposit.</p>

Parameter	Description
Build Dynamic Request	Set this parameter to <a href="#">Yes</a> to use the <a href="#">Requested Segments</a> table as the source of the query call.
Table Format	<p>The area labelled <a href="#">Table Format</a> specifies the formatting which is used for the result sheets. Note that this part includes named cells that should not be deleted. Otherwise the rendering code will no longer run.</p> <p>To change the formatting of the table in the data sheets change the formatting of this table accordingly using Excel standard formatting options. Make sure you don't deleted or add cells in this table.</p>
Format for Changed Fields	The cell right to the label provides the formatting which is applied to changed fields in the result sheets. Only the values in the <a href="#">New</a> columns will be formatted like that.
Additional Filters (ANDed)	This table defines filter expressions (XPath) that are applied locally in Excel to the response XML of the Compound Employee API to check whether an employee is in . An employee is in if he matches all XPath expression in the list. For more information see section <a href="#">Filter Logic</a> .
Additional Filters (ORed)	This table defines filter expressions (XPath) which are applied in Excel locally to the response XML of the Compound Employee API to check whether an employee is in . An employee is in if he matches all conditions of the <a href="#">Additional Filters (ANDed)</a> table and he matches any of the XPath expression in this list. For more information see section <a href="#">Filter Logic</a> .
Keep Intermediate Results	<p>This parameter controls whether intermediate results are kept in the Excel workbook beyond the end of the extraction run. The intermediate results (kept on hidden sheets of the workbook) are required for calculating the delta information and contain more data than are shown on visible sheets. Therefore, for security reasons, this option should be set to No in productive environments. This ensures that intermediate results are deleted once the visible sheets are filled.</p> <p>You have to set this to Yes when you want to send the filled report to SAP for troubleshooting purposes.</p>

Parameter	Description
Test Mode	<p>This parameter controls whether data is fetched from the backend or whether the intermediate results contained in the workbook are used. This mode is typically used for trouble-shooting when the person debugging the code has no access to the backend system. Prerequisite is that the intermediate results have been kept in the workbook by retrieving with <a href="#">Keep Intermediate Results = Yes</a> before.</p> <p>This parameter must be set to <a href="#">Off</a> in productive environments.</p>
Change Sheets	<p>This table contains the names of the sheets that are intended to receive changed data in contrast to the table receiving new data for hire and rehires. If you don't need one of these sheets to be filled, you may hide or delete it and remove the corresponding entry from this list. In case you introduce a new sheet containing changes, you have to add the sheet name to this table. Otherwise it won't be filled with data. The multi-sheet workbook contains a <a href="#">Payment Info</a> sheet that is not a part of this table. For more information see <a href="#">Using Payment Information [page 58]</a></p>
Hire Sheet	<p>This field contains the name of the sheet to receive new data for hired or rehired employees. You do not necessarily need such a sheet. For example, the workbook template CompEmpDelta_multiSheet.xlsx has no <a href="#">Hire Sheet</a> because new data goes into the entity sheets.</p>
Requested Segments	<p>This table contains the technical segment names used to query data from the backend using the Compound Employee API. All segments included in this table are supported by the Compound Employee API. Remove the segments from this table that you don't want to use for data processing in the Delta Export workbook.</p> <p>The result sheets for <a href="#">Recurring Deduction</a>, <a href="#">One Time Deduction</a>, <a href="#">Personal Documents Information</a>, and <a href="#">Alternative Cost Distribution</a> are only filled with data if their data segments are requested in this table. They are not part of the standard query. For more information see <a href="#">Using the Requested Segments Table [page 58]</a></p>

## Related Information

[Filter Logic \[page 31\]](#)



---

## 4.4.6 Errors Sheet

The Errors sheet shows all employees who could not be replicated for whatever reason. These employees will not appear on any of the results sheets. You need to take this very seriously and handle these employees manually.

The reasons are typically severe inconsistencies in the time slices of effective-dated entities for this employee. For example, if a compensation time slice starts at a time for which no job information time slices exists.

## 4.4.7 Warnings Sheet

The Warnings sheet shows employees for which certain fields could not be retrieved due to picklist or FO configuration issues. Those employees will show up on the result sheets as usual. However, the fields indicated in the warning will remain empty.

A very common example is following warning:

*"custom-string1" can't be returned. Please make sure that the external code for picklist "currency" with option ID "12722" is valid.*

This means that the field *custom-string1* on the entity which is shown right to the warning cannot be filled for the employee in question since the data model has a picklist "currency" configured for this field which is not properly configured. The Compound Employee API and as a result also the Excel application require the external code to be filled for all picklist options. Otherwise such a warning appears.

Another reason could be that the picklist assignment was changed in the data model so that a picklist was assigned to a formed free-text field or the picklist assignment was changed to a different picklist. In this case the valued assigned to employees based on the formed picklist is no longer valid for the new picklist.

## 4.4.8 Internal Sheets

There are a couple of worksheets in the reporting workbook that are used for internal purposes and that are not shown to the end user unless you make them visible using the [Show Internal Sheets](#) function on the [Developer](#) tab. You shouldn't change or remove these sheets. This is a short overview, for your information:

### Lookup

This sheet contains a couple of tables with foundation data that are used to show the user-friendly descriptions of fields on the result sheets that are based on foundation objects. For example, to show the description of a pay component. This is sometimes desired because the Compound Employee API only returns the external codes of these fields. See section *Extending Employee Delta Export* for more information on how to make use of the foundation data on this sheet.

---

The data on this sheet is refreshed from the backend automatically when an unresolved reference to lookup data is detected after filling the result sheets.

## LookupPerson

This sheet contains additional data for persons who are referred to on the result sheets. This is useful for cases where the Compound Employee API only provides the user ID of these persons. The job information, for example, contains the field *Manager* which is in fact the user ID of the manager. If you want to show the first and last name of this manager you can make use of the LookupPerson sheet. See section *Extending Employee Delta Export* for more information.

## Picklists

This sheet contains all picklist entries for all picklists found in the backend. This is useful when you want to show the user-friendly description of a field on the result sheets, which is based on a picklist. In the sample workbook CompEmpDelta\_multiSheet.xlsx the *Address Type* field on the *Address* sheet is such a case. This is sometimes desired since the Compound Employee API only returns the external codes of these fields. See section *Extending Employee Delta Export* for more information on how to make use of the foundation data on this sheet.

## Sheets DeltaResult, SnapshotResult, FullResult, Responses

These sheets are used for internal processing. During a report run they receive the raw data returned by the API calls to the backend. As a default, the content of the sheets is deleted at the end of a report run. For trouble-shooting purposes, however, it makes sense to keep the data on these sheets. This can be controlled by the configuration parameter *Keep Intermediate Results*.

## Scrambling

This sheet contains a table of XPath expressions that are used by the *Scramble Data* function on the *Developer* tab. These XPaths specify the fields that will be affected by the scrambling algorithm.

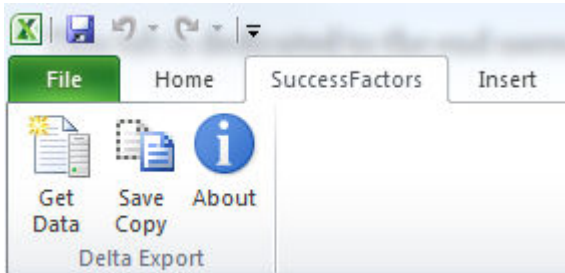
## Related Information

[Extending Employee Delta Export \[page 50\]](#)

## 4.5 Tabs on the Ribbon

### 4.5.1 SuccessFactors Tab

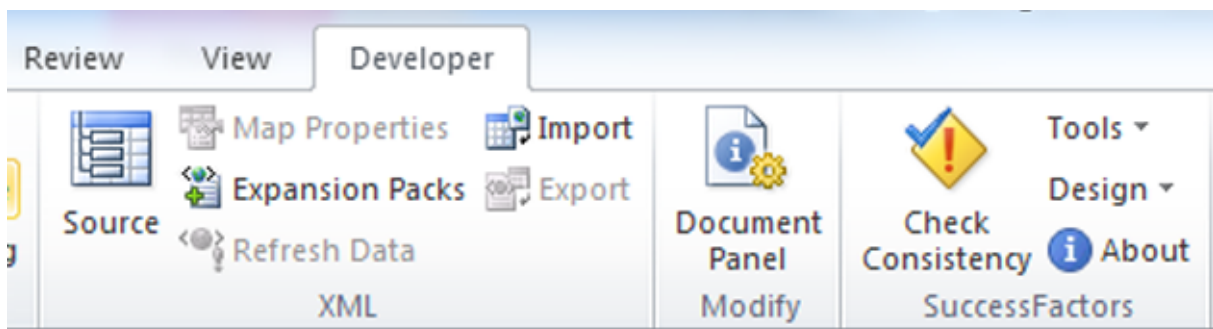
Once you've installed the add-in, the SuccessFactors tab will appear in the ribbon:



This tab is dedicated to the end users of Employee Delta Export.

- [Get Data](#): starts the extraction of data from the backend and subsequent post-processing
- [Save Copy](#): saves a copy of the Excel which is intended for send out to other people in or outside your organization. This copy doesn't contain any internal sheets or hidden business data.
- [About](#): gives you the version information for the installed add-in

### 4.5.2 Developer Tab



The Employee Delta Export for Excel add-in adds a group to the Developer tab. The functions in this group are useful especially for an admin who needs to customize the sample workbooks.

#### **i** Note

The Developer tab is only visible if you have enabled it in Excel.

## 4.5.2.1 Consistency Check

Execute this function after you change the configuration part of a workbook. It checks if the configuration is consistent.

### Caution

Performing a consistency check will delete the data on the result sheets.

In particular the following checks are performed:

- Are the XPath paths contained in rows 1 and 2 of the Result sheets correct?  
If the rows contain only simple segment and field XPath paths these are cross-checked against the signature of the Compound Employee API in the backend. If the XPath contains logical or other operators it is only checked if the XPath paths are syntactically correct.
- Are the *New / Old* column pairs considered in the formula of the column named *EC\_RelevantChanges*?  
This is essential because otherwise changes in these columns are not considered during delta calculation.
- Are segment XPath paths maintained for columns that have correct lookup formulas?

## 4.5.2.2 Tools

### Scramble Data

The Scramble Data tool makes the personal data on the result and internal sheets anonymous. Use this before sending a spreadsheet filled with sensitive data to unauthorized persons.

The rules for this scrambling are on the hidden *Scrambling* sheet. The sample workbooks deliver a set of XPath paths appropriate for the standard Employee Central data models. If a customer's data model introduces additional personal data in custom fields, it's up to the customer to add rules for these fields as well.

### Note

Data on the hidden sheets *Delta Result*, *Snapshot Result*, and *Full Result* may contain data that is not shown on the visible result sheets. Rules will cover this data as well.

### Show/Hide Internal Sheets

Hides or unhides all sheets used for internal purposes. Show these for trouble shooting in case unexpected results are shown on the result sheets.

## Show and Hide Configuration Rows/Columns

Shows and hides all rows and columns with configuration information, these are the rows one, two, and three of the result sheets

Use this function rather than manually hiding the configuration rows and columns to ensure that all such rows and columns are affected.

## Upgrade Workbook

Use [Upgrade Workbook](#) to upgrade your customer's workbook with new features delivered by SAP. The result is a workbook with customer-specific enhancements as well as new features, such as new result sheets, new selection parameters on the [Selection](#) and [Summary](#) sheets, or new configuration parameters on the [Config](#) sheet. The result sheets containing business data are typically not affected by upgrades.

To upgrade customer-specific workbooks, do the following:

1. Download the most recent version of the workbook provided by SAP from the [Employee Delta Export UI](#).
2. Open the customer-specific workbook. This workbook can be of version 1502 or higher.
3. Choose the [Upgrade Workbook](#) tool from under [Tools](#) on the [Developer](#) tab.
4. When prompted, open the standard workbook you just downloaded in step 1.
5. Follow the instructions.
6. Check the [Upgrade Log](#) sheet for messages and required follow-up actions.

## Upgrading Result Sheets

Before adding a new result sheet, the upgrade process checks if a sheet with the same name already exists. If, for example, the [Payment Info](#) sheet already exists, when it is about to be copied to the target workbook, the upgrade step is skipped entirely and a message is written to the [Upgrade Log](#). The upgrade log is a tab that is created for each upgrade run and is named [Upgrade Log <starting version> to <target version>](#) for example [Upgrade Log 1505 to 1511](#). It contains all information about which upgrade steps have been performed and all follow-up actions needed after the upgrade. Action items are highlighted in yellow.

## Upgrading Selection and Summary sheets

If new parameters need to be added to the [Selection](#) or [Summary](#) sheets, the upgrade process checks if the sheets still contain the default content as delivered in the standard workbooks. If the content is unchanged, the parameters are added in at the same place as they are located in the standard workbook of a higher version. Customer-specific parameters, comments, and any changed cell content is recognized as being different from the default content. In this case the new parameters are added to a new location at the bottom of the sheet. Feel free to move the parameters from their new location to a spot that suits you best.

### **i** Note

When moving rows to another position of the sheet, make sure to copy the named ranges to the new location as well. Otherwise, the workbook might not work correctly.

## Upgrading the Config Sheet

If new configuration parameters need to be added, the whole existing [Config](#) sheet is replaced and renamed to [Config Upgrade <Date>](#). The values are still available in the renamed sheet and can be copied over to the new sheet. You can delete the renamed sheet after the upgrade process.

## 4.5.2.3 Design

### Add Columns

The [Add Columns](#) function starts a wizard that enables you to add one or more columns based on fields that are available for a specific segment in the Compound Employee API signature.

#### ➔ Remember

Make sure that a backend system is defined on the [Selection](#) sheet.

The selection table has 4 columns.

- The column [Field Name](#) displays the technical name of the field in the Compound Employee API and a check box that allows the selection of multiple fields in one segment.

#### i Note

The selection of multiple fields from various segments is not supported

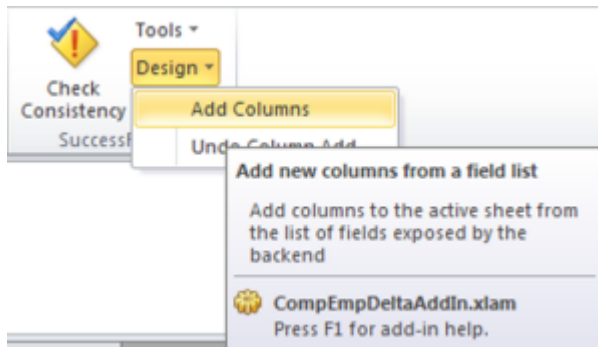
- The [Column Heading](#) column contains the field label found in the data model. If no field label has been maintained in the data model, the value is derived from the technical name.
- The columns [New/Old Columns](#) and [Always Show Value](#) display how the generated column will be configured.

When a row in the field list is selected, the [Column Properties](#) are shown below the list.

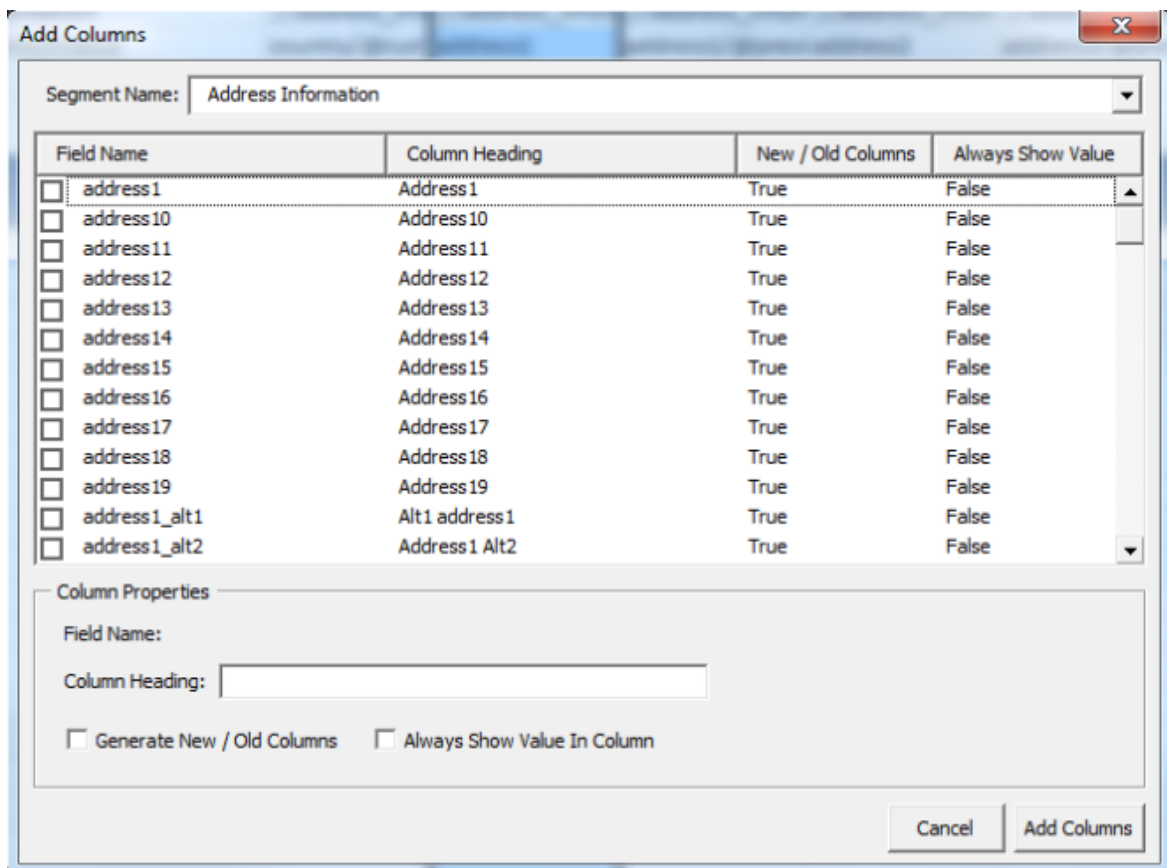
In the [Column Properties](#) you can change the Column Heading and the generation properties. Selecting the [Generate New/Old Columns](#) option creates a new/old column pair for the selected fields that displays the current value and the value before the change. Furthermore, these columns are considered in the formula in the [EC\\_Relevant Changes](#) column. If this option is not selected, only a single column is created that shows the current value (if that particular field has changed since the last extraction). By selecting the [Always Show Value in Column](#) option you generate a column that always shows the current value of a field, even if the value hasn't changed since last extraction. If neither [Always Show Value in Column](#) nor [Generate New/Old Columns](#) are selected, the new column will only display the values that have changed since the last extraction. For fields that are picklist based, lookup columns are generated that show the picklist labels. The columns that contain picklist values are generated and then hidden.

## Using the Wizard to add Columns

1. Place your cursor in the column to the left of where you would like to add the new column and click [Add Columns](#).



2. Verify that the pre-selected segment name is the right one, if not just select the desired segment from the list. This is the segment in the API signature that will be bound to the column.



3. Pressing the [Add Columns](#) button will add the selected columns to the currently selected sheet.

### **i** Note

If you use the [Add Columns](#) function on the [Hire](#) sheet of a single-sheet type workbook, the options [Generate New/Old Columns](#) and [Always Show Value in Column](#) are not available.

## Payment Information Segments in Add Columns

If you have enabled Payment Information and you have configured the workbook to use Payment Information, the [Add Columns](#) wizard will display all available fields for the Payment Information segment and its sub-segments. The Payment Information segment has an hierarchical structure:

- `PaymentInformationV3`
  - `PaymentInformationDetailV3`
    - `PaymentInformationDetailV3USA`
    - ....
    - `PaymentInformationDetailV3GER`

Most of the relevant Payment Information fields are on the `PaymentInformationV3Detail` segment node. Therefore, the wizard uses this node as the base node. The parent node is the `PaymentInformationV3`, fields on the root node begin with `../` for example `../createdBy`. Fields that start with segment names are the country-specific child nodes of the Detail node. For example `PaymentInformationDetailV3JPN/createdBy`. Even if a field on such a country-specific segment has the same name as a field on another country segment, they are generated as separate columns on the [Result](#) sheet.

### Note

It is possible to create custom associations on the Payment Information segment and its sub nodes, however, the associations will not be taken into account by the [Add Columns](#) wizard and only fields on standard segments of Payment Information are included on the field list.

## Hierarchical MDF segments in Add Columns

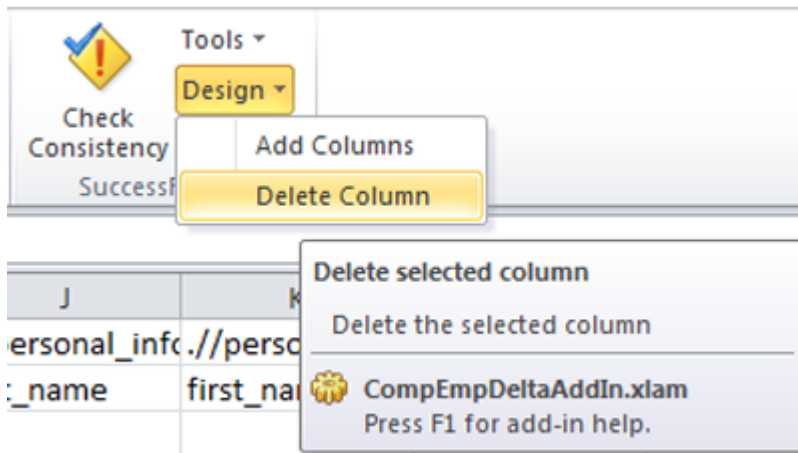
In the [Add Columns](#) wizard, hierarchical MDF-based segments that have a header segment with items underneath, like for example [Alternative Cost Distribution](#) or [Recurring Deductions](#), are handled the same way as [Payment Information](#). The item level is then used as segment XPath and fields of the header segment are addressed with `../` in the same way as is done for [Payment Information](#). Custom associations are not reflected in the [Add Columns](#) wizard.

## Delete Column

The [Delete Column](#) function deletes a selected column from a result sheet. The difference to the standard Excel delete function is that this function preserves the consistency of the workbook from an SAP perspective.

If the column to be deleted is used in the change formula of the active sheet, the formula is adjusted automatically. You can only delete single columns at a time.





### **i Note**

If the column you have selected for deletion is the last column referenced in the [EC\\_RelevantChanges](#) formula, the deletion is not executed and an error message is displayed. This ensures the functional correctness of the [EC\\_RelevantChanges](#) formula.

## **Refresh Field List Cache**

The [Add Columns](#) wizard uses an internal cache to display all fields that are available for the supported segments. With the help of the cache, only one login is needed to call a describeEx and retrieve the field list which is then displayed. In cases where changes have been made to the company's data model, that are not yet reflected in the [Add Columns](#) field list, you can trigger a refresh using this option. However, this will require another login to call the describeEx.

---

## 5 Extending Employee Delta Export

Employee Delta Export comes with two different variants of reporting workbooks. You can use these as a starting point but you certainly need to customize them to fulfill the requirements of a productive use case. Customizing can just be changing the configuration data that is included in the reporting templates. No programming is required for this. Some XML knowledge is beneficial, however, since the configuration uses XPath expressions to identify the elements in the XML response of the Compound Employee API calls. Customizations that require a change in the add-in should be sent to SAP as a requirement. Changes in the add-in's VBA code are not recommended since these will be overwritten by the next update.

There are three categories of changes you can apply to the reporting workbooks:

1. Simple layout changes
  - Change column headers
  - Delete columns
  - Delete an entity sheet: delete the entire worksheet and remove it from the configuration on the [Config](#) sheet
  - Add new calculated columns
  - Remove filter parameters: delete the row on the [Selection](#) sheet and ensure it is not used on the [Config](#) sheet
  - Change the file name pattern: see section *Config Sheet*.
  - Add new rows between row 3 and row 5 of a result sheet.
2. More sophisticated changes (requires some knowledge of XML and the Compound Employee API):
  - Add new columns from the database
  - Add user-friendly descriptions for picklist-based columns
  - Add user-friendly descriptions for FO-based columns
  - Add personal data for user ID fields
  - Add information about workflow approvals
  - Add filter parameters
  - Add validation rules for filter parameters
3. Destructive changes (will prevent the extraction from running properly):
  - Deletion of one of the non-entity related sheets such as [Summary](#), [Selection](#), [Config](#), and so on
  - Delete one of the first three (hidden) rows of the entity sheets
  - Change a name assigned to individual cells, rows, or columns
  - Add text adjacent to tables in the [Config](#) sheet (for example additional explanations)
  - Add rows after the header row (originally row 5) of the entity sheet

### 5.1 Column Configuration

Before you start adding new columns to a result sheet you need to understand how column configuration works. This configuration is in the hidden rows one, two and three of each result worksheet. You can make

configuration visible using the function [Show Configuration Rows/Columns](#) in the [SuccessFactors](#) group on the [Developer](#) tab.

**Row 1** contains an XPath expression identifying the segment in the XML result of the Compound Employee API that carries the field to be shown in the column in question. For example the XPath `./personal_information` identifies the **bold** element in the XML result below.

```
<person xmlns="urn:sfobject.sfapi.successfactors.com">
  <birth_name>Kendrick</birth_name>
  <personal_information>
    ...
    <first_name>Amy</first_name>
    ...
    <last_name>Kendrick</last_name>
    ...
    <start_date>2014-07-11</start_date>
  </personal_information>
  ...
</person>
```

**Row 2** contains an XPath fragment that identifies the element starting from the corresponding segment XPath in row 1. For example, if you want to show the field `first_name` in the column just enter `first_name` in row 2.

**Row 3** can contain a valid Excel formula that calculates the values in this column from other columns. The formula must use cells references in row 3 to refer to values in the same row. The formula will be copied to rows below automatically.

The screenshot below shows the configuration for first name, last name, and formatted name calculated from the first and last name. Note especially the formula in cell G3 of the graphic below.

	D	E	F	G	
	<code>./personal_information</code>	<code>./personal_information</code>	<code>./personal_information</code>		
	salutation	last_name	first_name		date
				<code>=F3 &amp; " " &amp; E3</code>	
	Personal Information				
	Title	Last Name	First Name	Formatted Name	

### **i** Note

All columns of the same segment XPath should be configured contiguously. A column with a different segment XPath B in between a sequence of columns with XPath A leads to unexpected results. However, calculated columns interrupting the sequence is not harmful.

## 5.2 Adding new columns from the database

### Context

Adding a new column to an existing sheet works as follows. For the following description let's assume you want to add the field `custom_string5` of personal information.

### Procedure

1. Figure out the field name of the field in the signature of the Compound Employee API.

For a reference of valid fields please go to the [SFAPI Data Dictionary](#) in [Admin Tools](#).

The dictionary says the field name is `/person/personal_information/custom_string5`

2. Make the first three rows of the worksheet visible. These rows contain configuration information.
3. Insert a new column where you want the additional data to appear.

In the first row of the new column add `./person_information` which is an XPath expression selecting the time slices of personal information within the result of the Compound API. The starting point of the XPath is assumed to be the `<person>` element. XPath expressions may also include restrictions like `./address_information[address_type='home']`. The XPath can also make use of content in neighboring columns of the same row. In order to do this, introduce a name for the column with the prefix `EC_Col_` and use the part following the prefix in the XPath.

#### Example

The XPath `./address_information[start_date>='%StartDate%']` refers to a column with the name `EC_Col_StartDate`.

4. In the second row of the new column add `custom_string5` which is the field name of the new field in the result of Compound Employee API.

The final XPath picking the field value out of the XML result is concatenated from the segment XPath above and the field name, which is in fact taken as an XPath fragment.

5. The purpose of the Employee Delta Export application is to show only records that shall be forwarded to a payroll system or a similar system.

Typically, these are the records which are new or have changed significantly since a certain point in time. For this purpose each sheet contains a column with a formula that determines if a row is relevant.

To add the newly added columns to this formula, go to the column with name [EC\\_RelevantChanges](#) on the same sheet and adapt the formula in row 3 of this column. The formula will return a value unequal blank if the row contains relevant changes. This ensures that the row appears on the result sheet. Note that this column may be hidden. You can figure out where it is by using the [Name Manager](#) function on the [Formulas](#) tab in Excel.

The next time you run the extraction, the new column will be populated if the field value has changed in Employee Central since the specified report start date.

If you want to see both the current field value and the old value on the sheet you need to introduce a second column for the same field. This works mostly as described above with the difference that the field

name entered in the second row must be [custom\\_string5/@previousValue](#). This denotes the value as it was at the point in time of the last extraction.

If you want to see the current value regardless of whether it was changed since the last extraction or not, use [custom\\_string5/@currentValue](#) as a field name in the second row of the column in question. This makes sense for key fields as well as for attributes like currency.

In the segment XPath you can use a notation like [full://personal\\_information](#). This denotes that the column is filled from the full extract rather than from the delta extract. This ensures that the field is filled even if `personal_information` (in this example) has not actually changed.

You may populate the columns of a particular sheet from different segments in the API result structure. Note, however, that data from different segments may go into different rows in the worksheet. This may lead to unexpected results.

## 5.3 Adding Calculated Columns

To add a column that contains a value calculated from other columns of the same row, enter a formula into row 3 of the column that refers to other cells in the same row. See section *Column Configuration* for an example.

## 5.4 Adding user-friendly descriptions for picklist-based columns

The Compound Employee API only provides external codes for fields based on picklists. If you want to display your own user-friendly descriptions rather than the external codes you can do so by adding a new column with a formula that does a lookup to the [Picklists](#) sheet, which is typically hidden. Take column [Address Type](#) in the sample reporting workbook [CompEmpDelta\\_multiSheet](#) as a reference. It uses the formula

```
=VLOOKUP (CONCATENATE ("addressType|";G3);Picklists!A:D;4;FALSE)
```

which looks for the row containing the address type's external code on sheet `Picklists` and returns the value in column 4 which is in fact the label.

The screenshot below shows the configuration for the lookup of the field `Address Type` related to the `Address Type` code.

G	I	
<a href="#">./address_information</a>		
<a href="#">address_type/@currentValue</a>		
	<code>=VLOOKUP(CONCATENATE("addressType ";G3);Picklists!A:D;4;FALSE)</code>	=\
<b>Address Type Code</b> [New]	<b>Address Type</b> [New]	<b>A</b> [C

## 5.5 Adding user-friendly descriptions for FO-based columns

The Compound Employee API only provides external codes for fields based on foundation objects. If you want to display your own user-friendly descriptions rather than the external codes you can do so by adding a new column containing a formula that does a lookup to the [Lookup](#) sheet, which is typically hidden. For example, use column [Pay Component](#) in the sample reporting workbook `CompEmpDelta_multiSheet` as a reference. It uses the formula

```
=VLOOKUP(E3;EC_Lookup_PayComponent;2;FALSE)
```

which looks for the row containing the pay component's external code in table [EC\\_Lookup\\_PayComponent](#) on sheet [Lookup](#) and returns the value in column 4 which is in fact the label. The table is filled automatically when unresolved references are found after processing the result of the Compound Employee API call.

The graphic below shows the configuration of the lookup of the pay component related to the pay component code.

E	F	
./paycompensation_recurring pay_component/@currentValue		./p star
	=VLOOKUP(E3;EC_Lookup_PayComponent;2;FALSE)	
Pay Component Code	Pay Component	

### Note

Time dependencies of foundation objects are not considered. Only the currently effective values are used.

## 5.6 Deleting columns

When you delete a column from the result sheet, you have to make sure that the references are still valid. To do so, check the formulas in row 3 for [REF!](#) errors and resolve them, especially the [EC\\_RelevantChanges](#) column. If the column you want to delete is referenced in a lookup as source column, you have to delete the lookup column as well. If the [REF!](#) error appears in the formula in the [EC\\_RelevantChanges](#) column, delete the expression containing the [REF!](#) entry from the formula.

### Tip

Use the [Delete Column](#) tool as described in section [Design \[page 46\]](#) to remove columns from result sheets. When deleting columns using this tool, all references stay valid.

### Example

If the formula in *EC\_RelevantChanges* looks like this:

```
=IF(OR(#REF!<>G3;H3<>I3);"X";"")
```

Change it to:

```
=IF(OR(H3<>I3);"X";"")
```

In this way, you make sure that the changes calculation still works as desired, because corrupt formulas will prevent the formula to work at all. Use the *Check Consistency* function after removing columns to verify that workbook configuration is still intact.

## Related Information

[Design \[page 46\]](#)

## 5.7 Adding personal data for user ID fields

To show personal information like first name or last name along with the user ID that is returned by the Compound Employee API, you need to introduce a new column with a lookup formula like:

```
=VLOOKUP(A3; EC_Lookup_PersonalInfo; 3; FALSE)
```

which refers to the table *EC\_Lookup\_PersonalInfo* on the *LookupPerson* sheet. Once the formula is detected by the Delta Reporting logic, the personal information is retrieved for the user IDs found in the column referred to by the formula (in this case column A).

The graphic below shows the configuration for the lookup of the last name related to the user id (here *kkkk*).

	A	B	
1			
2	person_id_external		
3		=VLOOKUP(A3; EC_Lookup_PersonalInfo; 3; FALSE)	=VLOOKUP(
6	User Id	Last Name	First Name
7	kkkk	=VLOOKUP(A7;EC_Lookup_PersonalInfo;3;FALSE)	=VLOOKUP(

### Note

Only personal data for active users can be retrieved.

### **i** Note

This logic will have an impact on performance.

## 5.8 Adding information about workflow approvals

### Context

To show information about workflow approvals related to a segment you need to

### Procedure

1. Introduce a column that is mapped to the field *workflow\_request\_id* of the segment in question.

This is supported for the segments of the Compound Employee API:

- personal\_information,
  - job\_information,
  - compensation\_information, and
  - pay\_compensation\_non\_recurring.
2. Introduce a column with a lookup formula like

`=VLOOKUP(J3; EC_Lookup_Workflow; 4; FALSE)`

that refers to the table *EC\_Lookup\_Workflow* on sheet *LookupOData*. Once the formula is detected by the Employee Delta Export logic the workflow information is retrieved for the workflow requests found in the column referred to by the formula (in this case column J).

The graphic below shows the lookup for first name and last name of the approver of the last approval step.

J	K	L
./personal_information		./personal_information
workflow_request_id		
	=VLOOKUP(J3;LookupOData!A:E;4;FALSE)	=VLOOKUP(J3;LookupOData!A:E;5;FALSE)
WF Req Id	WF Last Name	WF First Name
	=VLOOKUP(J6;LookupOData!A:E;4;FALSE)	=VLOOKUP(J6;LookupOData!A:E;5;FALSE)
803	=VLOOKUP(I7;LookupOData!A:E;4;FALSE)	=VLOOKUP(I7;LookupOData!A:E;5;FALSE)

### **i** Note

This logic will have an impact on performance.



## 5.9 Adding new filter parameters

There are two types of filters

1. Filters supported by the Compound Employee API in the backend
2. Locally applied filters

To add a new filter that is supported by the Compound Employee API proceed as follows:

1. Go to the [Selection](#) sheet and add a new row similar to the one for cost center  
For the cell where the filter value is entered you need to introduce a name following the pattern [EC\\_Filter\\_<parameter name>](#) where [<parameter name>](#) is the name of the filter parameter defined in the Compound Employee API signature. For a reference of valid fields please go to the [SFAPI Data Dictionary](#) in the [Admin Tools](#). Uppercase/lowercase don't matter for the parameter name.

To add a new locally applied filter proceed as follows:

1. Go to the [Selection](#) sheet and add a new row similar to the one for country  
For the cell where the filter value is entered you need to introduce a name following the pattern [EC\\_FilterAtClient\\_<parameter name>](#) where [<parameter name>](#) is a name of your choice
2. Go to the [Config](#) sheet and add the new parameter to the XPaths of one of the [Additional Filters](#) sections that are already in place or add a new XPath to one of these sections.

You can also add filter parameters that are applied locally on the client after the result of the Compound Employee API call is available. To do so add a new filter condition on the [Config](#) sheet that makes use of the new filter parameter as a placeholder.

## 5.10 Sorting in the Result Sheet

By default, the result records on a single sheet are not sorted. However, it is possible to maintain sort filters on certain columns. The first columns of each sheet of a workbook use the [full://](#) notation in the field [XPath row](#). These columns can be sorted because all fields contain values, even though they might not be visible. Adding a new column using the same path notation scheme as the first three columns of a sheet will make that column sortable.

The columns [User ID](#) and [Person ID](#) (columns B and C) are always sortable even though they don't use the [full://](#) notation.

	A	B	C	D	E	F
1						./personal_information
2		person_id_external/@currentValue	person_id/@currentValue	full://personal_inform	full://personal_inform	start_date/@currentValue
3						
4	<b>Personal Information</b>					
5	Status	User Id	Person Id	First Name	Last Name	Start Date
6	Changed	Snapshot01	988	Snapshot	Germany	
7	Changed	Snapshot02	989	Snapshot	United States	
8	Changed					

## 5.11 Using Payment Information

If you have enabled Payment Information in Provisioning, there are a few steps you need to take to make the [Payment Info](#) sheet visible in the single sheet workbook as well as new columns on the [Hire](#) and [change](#) sheets of the single sheet workbook.

### Enabling Payment Information in the multi-sheet workbook

The [Payment Info](#) sheet is not used by the multi-sheet workbook by default, therefore, it is not included in the [Change Sheets](#) table on the [Config](#) sheet. To fill the [Payment Info](#) result sheet you have to, in the [Change Sheets](#) table:

1. Delete Direct Deposit
2. Add Payment Info

You can add either Direct Deposit or Payment Information. If you include both sheets in the table, you'll get a warning. The request that is sent to the Compound Employee API changes depending on which sheet is added to the table.

If you have already enabled Payment Information in Provisioning but the workbook is configured to use Direct Deposit, you will get a warning message and the call will not be executed. Same applies to the case where Direct Deposit is enabled in Provisioning and the workbook is configured to use Payment Information.

### Enabling Payment Information in the single-sheet workbook

To enable Payment Information for the single-sheet workbook you have to go to the [Config](#) sheet and set the [Select Option](#) for [Use Payment Information V3](#) to **Yes**.

Select Options	
Update New Report Start Date	Yes
Records per Request	100
QueryMore Iterations	unlimited
Use PaymentInformationV3	Yes

If you have already enabled Payment Information in Provisioning but the workbook is configured to use Direct Deposit, you will get a warning message and the call will not be executed. The same applies to the case where Direct Deposit is enabled in Provisioning and the workbook is configured to use Payment Information.

## 5.12 Using the Requested Segments Table

You can use the [Requested Segments](#) table on the [Config Sheet](#) to choose which segments are extracted from the backend. By default, the table contains all segments supported by the Compound Employee API by their

technical name. Remove the segments that are not needed for data processing in your case, to reduce the amount of data that is extracted from the backend.

Additionally, you can add the following segments that aren't available in the standard query:

- Deduction Recurring
- One Time Deduction
- Alternative Cost Distribution
- Personal Documents Information

By default, these result sheets are included in the [Change Sheet](#) table on the [Config Sheet](#), but are not filled with results. Data extraction for these sheets will only happen if defined in the [Requested Segments Table](#).

#### Restriction

Recurring Deduction, One Time Deduction, and Alternative Cost Distribution must have start dates later than **October 10, 2015**. Delta calculation can't be performed for time periods before, due to missing MDF audit history data.

#### Note

You can't use the [Period-based Delta](#) functionality if you request One Time Deductions (deductions\_non\_recurring).

After adjusting the segment list in the [Requested Segments](#) table remember to set the configuration parameter [Build Dynamic Request](#) on the [Config Sheet](#) to [Yes](#) so that the request XML is built using the segments you maintained in the table.

## Related Information

[Config Sheet \[page 36\]](#)

[Period-based Delta \[page 35\]](#)

## 5.13 Dos and Don'ts

### Field Sequence

Fields related to a particular segment in the result of the Compound Employee API should be in sequence without being interrupted by fields from other segments. The reason is that in sheets showing changes, a new row is started as soon as the segment changes. If field sequence is interrupted the fields of the same segment spread across different rows.

---

## Hidden sheets, rows

Don't delete hidden worksheets.

Don't delete the first 3 hidden rows of the worksheets that show the changes of an entity.

## Named Columns

Don't delete named columns on the change sheets. You can figure whether a column is a named column by selecting the column and checking the name field in the area above the A1 cell. Other columns can be deleted without restrictions, for example if the users no longer want to see it.

## Copying Sheets

You may copy sheets from one workbook to another to save time. A typical use case is to copy the [Config](#) or the [Systems](#) sheet to duplicate settings. To do so, right-click on the source sheet, choose [Copy](#) and choose the target. This may, however, create undesired copies of names that carry references to the source file. Use the [Name Manager](#) in Microsoft Excel to check this and to remove the undesired names.

## Report Start Date

Make sure to set a [Report Start Date](#) and other available selection criteria, such as the [Requested Segments](#) table to narrow down the target population that is extracted to the Delta Export workbook, thereby improving performance and memory usage.

## Requested Segments Table

By default, the [Requested Segments](#) table is delivered containing all segments that are supported by the Compound Employee API. Remove all the segments of the table you don't need for data processing. In that way, data extraction will take less time and post-processing has less data that needs to be managed.

### Note

This is only relevant if you have set the [Build Dynamic Request](#) parameter.

---

## 5.14 Tips and Tricks

### Using Formulas

If the basic fields retrieved from the backend do not fully meet your requirements you can introduce new columns that use formulas to calculate the content from other basic fields. For example, you can use a formula to compute a full address from the columns postal code, city, country, and address1. The formula must be entered in row 3 of a change sheet. Only this ensures that it is copied to the data when the sheet is populated.

Columns containing formulas should typically not be considered in the formula contained in the column named *EC\_RelevantChanges* if the original field is already in there. The reason is that the calculation may fail in a row for some reason and the row may appear unexpectedly.

### Population Changes

Replication of employee changes is easy when you have one payroll (or other type of) subsequent system to where you transfer the data. Things become more complex once you have various subsequent systems with different processors using different variants for the reporting workbooks. This is what we call populations. One population is a set of filter criteria which specifies the employees going into one subsequent system. Things become even more complex once employees move around between different populations. In this case you need to ensure that processors recognize when an employee enters his own population and you need to ensure that the processor doesn't receive changes for employees who don't belong to his own population.

---

## 6 Troubleshooting

### Context

When an error occurs during data extraction that needs to be analyzed by your administrator or by SAP, proceed as follows:

### Procedure

1. Switch *Keep Intermediate Results* to *Yes*. For a detailed description of this parameter see section *Config Sheet*.
2. Perform the extraction
3. Scramble the data to make personal data anonymous. For more information see section *Tools*.
4. Save the workbook and send it to the person who analyzes the issue. Remember that the workbook contains all business-related data of the extracted employees.

### Related Information

[Config Sheet \[page 36\]](#)

[Tools \[page 44\]](#)

## 7 Appendix

### 7.1 Screenshots

Selection sheet with selection parameters:

	A	B	C
1		<b>Employee Delta Export</b>	
2		Workbook Version: 1.18	Workbook Release: 1411
3		Add-in Version: 1411.0.0	Add-in Release: 1411
4		This workbook needs to be customized before using it productively	
5			
6		Choose System:	localhost
7			
8		New Report Start Date [UTC]	13.01.2015
9		New Report Start Time [UTC]	14:01:08
10			
11		Country	USA
12		Company (Legal Entity)	
13		Division	
14		Business Unit	
15		Department	
16		Cost Center	ABC
17		Work Location	
18		Pay Group	
19		Employee Class	
20		User Id	
21		Exclude User Ids	
22			
23		Last Report Start Date [UTC]	01.08.2014
24		Last Report Start Time [UTC]	
25		Last Report End Date [UTC]	13.01.2015
26		Last Report End Time [UTC]	14:01:08

Systems sheet with connecting backend system:

	A	B	C	D	E
2	Name	Url	Company	User	Password
3	Production	<a href="https://api099.successfactors.eu">https://api099.successfactors.eu</a>	myprod	sfadmin	
4	Test	<a href="https://api099.successfactors.eu">https://api099.successfactors.eu</a>	mytest	sfadmin	
5					

## Personal Info sheet with changes and new data of multiple employees:

	A	B	D	E	F	G	J	K	L	M	N
4	Personal Information										
5	Status	User Id	First Name	Last Name	Effective As Of (Start Date)	End Date	First Name (New)	First Name (Old)	Last Name (New)	Last Name (Old)	Middle Name (New)
6	New	20010	Amy	Kendrick	11.07.2014	11.12.9999	Amy		Kendrick		
7	New	20030	John	Scott	11.07.2014	24.08.2014	John		Scott		
8	New	*			25.08.2014	11.12.9999	John		Scott		
9	New	20040	Kevin	Anderson	11.07.2014	11.12.9999	Kevin		Anderson		
10	New	20050	Patricia	Walker	11.07.2014	11.12.9999	Patricia		Walker		
11	New	20060	Kimberly	Smith	11.07.2014	11.12.9999	Kimberly		Smith		
12	New	20070	Donna	Evans	11.07.2014	11.12.9999	Donna		Evans		
13	New	20080	Mark	Turner	11.07.2014	11.12.9999	Mark		Turner		
14	New	20090	Ethan	White	11.07.2014	11.12.9999	Ethan		White		
15	New	20100	James	Green	11.07.2014	11.12.9999	James		Green		
16	New	20110	Barbara	Williams	11.07.2014	11.12.9999	Barbara		Williams		
17	New	20120	Joseph	Wright	11.07.2014	10.09.2014	Joseph		Wright		
18	New	*			01.10.2014	11.12.9999	Joseph		Wright		Alexander
19	New	20200	Laura	Edwards	11.07.2014	11.12.9999	Laura		Edwards		
20	New	20210	Sandra	Nelson	11.07.2014	11.12.9999	Sandra		Nelson		
21	New	20220	Thomas	Baker	11.07.2014	11.12.9999	Thomas		Baker		
22	New	20240	Justin	Lewis	11.07.2014	11.12.9999	Justin		Lewis		
23	New	20250	Todd	Allen	11.07.2014	11.12.9999	Todd		Allen		
24	New	20260	Jesse	Young	11.07.2014	11.12.9999	Jesse		Young		
25	New	20270	Sean	Hernandez	11.07.2014	11.12.9999	Sean		Hernandez		
26	New	20280	Earl	Hall	11.07.2014	11.12.9999	Earl		Hall		
27	New	20120	Robert	Lopez	11.07.2014	11.12.9999	Robert		Lopez		
28	New	20130	Michael	Lee	11.07.2014	11.12.9999	Michael		Lee		
29	New	20140	Elizabeth	Mitchell	11.07.2014	11.12.9999	Elizabeth		Mitchell		



---

# Important Disclaimers and Legal Information

## Coding Samples

Any software coding and/or code lines / strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended to better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, unless damages were caused by SAP intentionally or by SAP's gross negligence.

## Accessibility

The information contained in the SAP documentation represents SAP's current view of accessibility criteria as of the date of publication; it is in no way intended to be a binding guideline on how to ensure accessibility of software products. SAP in particular disclaims any liability in relation to this document. This disclaimer, however, does not apply in cases of wilful misconduct or gross negligence of SAP. Furthermore, this document does not result in any direct or indirect contractual obligations of SAP.

## Gender-Neutral Language

As far as possible, SAP documentation is gender neutral. Depending on the context, the reader is addressed directly with "you", or a gender-neutral noun (such as "sales person" or "working days") is used. If when referring to members of both sexes, however, the third-person singular cannot be avoided or a gender-neutral noun does not exist, SAP reserves the right to use the masculine form of the noun and pronoun. This is to ensure that the documentation remains comprehensible.

## Internet Hyperlinks

The SAP documentation may contain hyperlinks to the Internet. These hyperlinks are intended to serve as a hint about where to find related information. SAP does not warrant the availability and correctness of this related information or the ability of this information to serve a particular purpose. SAP shall not be liable for any damages caused by the use of related information unless damages have been caused by SAP's gross negligence or willful misconduct. All links are categorized for transparency (see: <http://help.sap.com/disclaimer>).

**go.sap.com/registration/  
contact.html**

© 2016 SAP SE or an SAP affiliate company. All rights reserved.  
No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company. The information contained herein may be changed without prior notice.  
Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.  
These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.  
SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.  
Please see <http://www.sap.com/corporate-en/legal/copyright/index.epx> for additional trademark information and notices.