**SAP HANA Flat File Upload Tutorial: CSV, XLS & XLSX**

<https://www.guru99.com/flat-file-upload-to-sap-hana.html>



For an Excel list of ALL SAP extractors that are currently released for ODP - see the attachment to this SAP Note.

SAP ODP extractor for Excel

How to Export Data from Excel to SAP

Sap BW Source Systems Excel

Sap BW Data Source Excel

# Excel file (\*.xlsx) Export/Import

<https://blogs.sap.com/2019/06/06/excel-file-.xlsx-exportimport/>

# How to upload Excel to SAP(using ABAP)

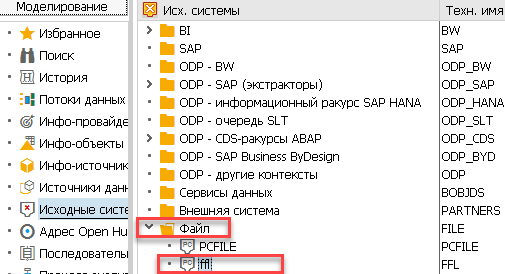
<https://www.samplecodeabap.com/how-to-upload-excel-to-sapusing-abap/>

**How to Load Transaction Data from Flat File in SAP BI/BW**

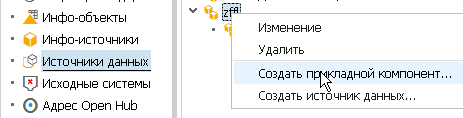
<https://www.guru99.com/how-to-load-transaction-data-from-flat-file.html>

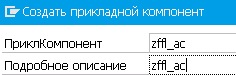
1. **Create Source System for flat file**

Source System = ZFLL



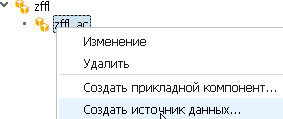
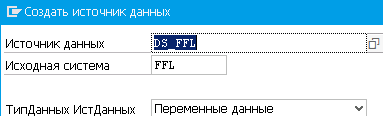
1. **Create Application component**

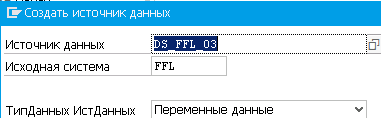




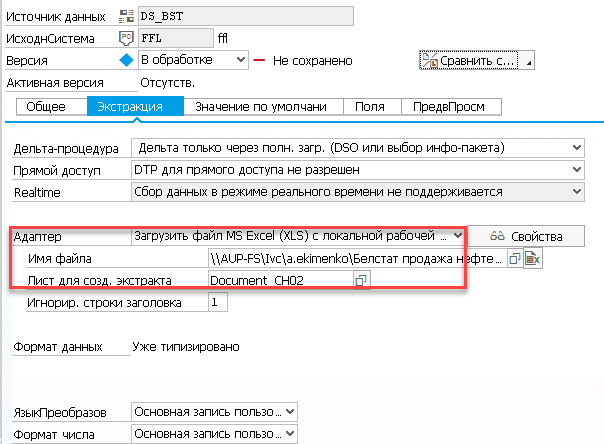


1. **Create Data Source**



* 1. ***Настройка Data Source***



Добавление/удаление полей –

Вкладка 



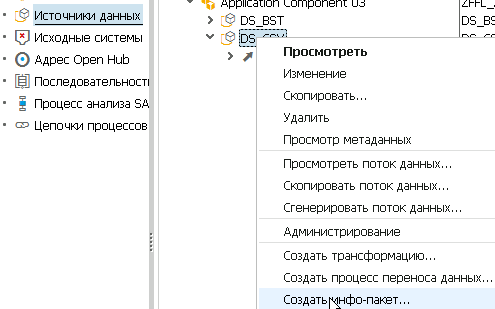
Подпрограмма RSDAT

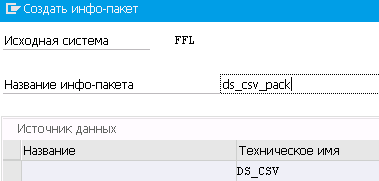


− Only transaction data is to be loaded to your InfoProviders. You have to connect your DataSource for transaction data to your cube creating a transformation / DTP. Then *load your transaction data to PSA via InfoPackage and update to cube*. It might be useful / necessary to think about an integration layer with DSO.

Master data is loaded to master data tables of InfoObjects - define your characteristics as *with master data* and connect your master data DataSource via transformation.

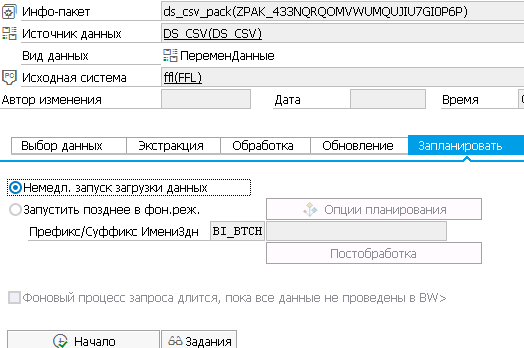
1. **Create Infopackage and Schedule dataload to the DataSource /PSA/**





***4.1 Click on Schedule tab***

***4.2 Click Start button to start the load from the flat file to the Data Source***



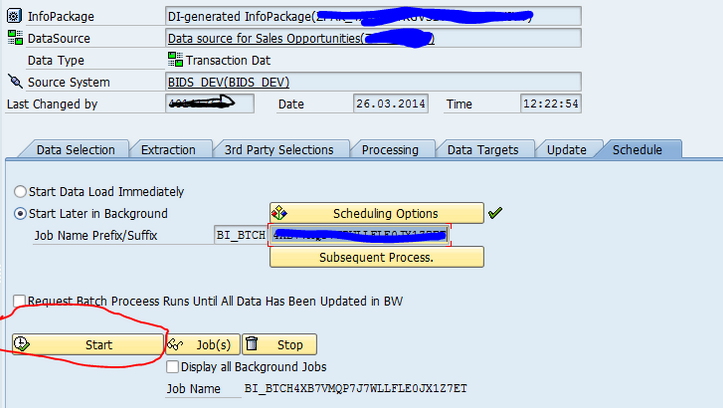
1. **Load data to the DSO**

***4.1 Click Execute tab in the DTP.***

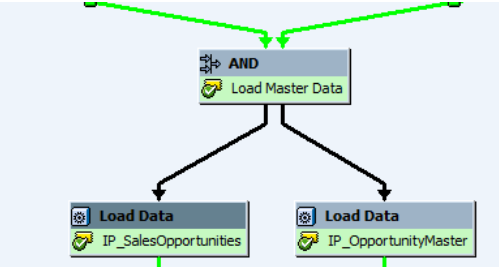
***4.2 Click the Execute Button to start data load from the DataSource(PSA) to the DSO.***

# How to Execute and scheduled BODS jobs from BW.

<https://blogs.sap.com/2014/04/11/how-to-execute-and-scheduled-bods-jobs-from-bw/>



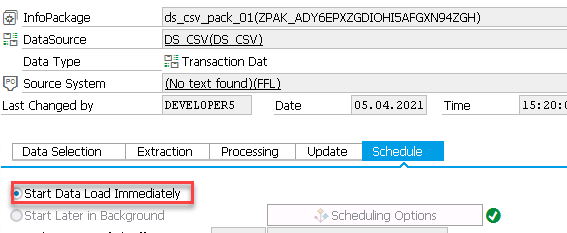
Add this infopackage into process chain and schedule it



*Строка поиска*

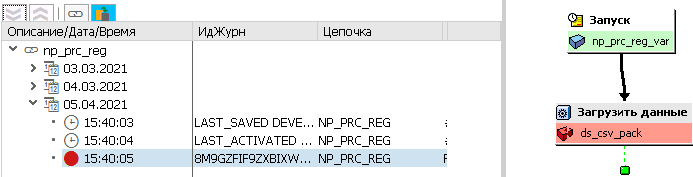
Sap blog infopackage shedule start later in background

*Почему не изменяется Start Data Load Immediatly*



<https://answers.sap.com/questions/6345663/infopackage---scheduling-tab.html>

*Loading InfoPackage ZPAK\_5D0CE8N9T90K28B8WPNEUVKW1 from local PC; cannot run in batch*



<https://apps.support.sap.com/sap/support/knowledge/en/2639702>

### **Cause**

You cannot load flat file data from a client workstation in the background. You therefore have to *store your data on an application server*.

### **Resolution**

1. Place your Flat File in Application server. Use tcode AL11-->DIR\_HOME
2. Upload can be performed in numerous ways
3. Network share
4. FTP
5. In the Infopackage Extraction tab, select Application server instead of local workstation

# [Loading Transaction data from flat file](https://easy-learn-bw.blogspot.com/2013/06/loading-transaction-data-from-flat-file.html)

<https://easy-learn-bw.blogspot.com/2013/06/loading-transaction-data-from-flat-file.html>

# How to automatically load flat files from an FTP server into SAP BW

<https://www.element61.be/en/resource/how-automatically-load-flat-files-ftp-server-sap-bw>

# SAP BW Flat File Loading to InfoObject

# <https://erproof.com/bi/sap-bw-training/sap-bw-flat-file-loading-to-infoobject/>

Make sure that you keep the following things in mind for the various file types

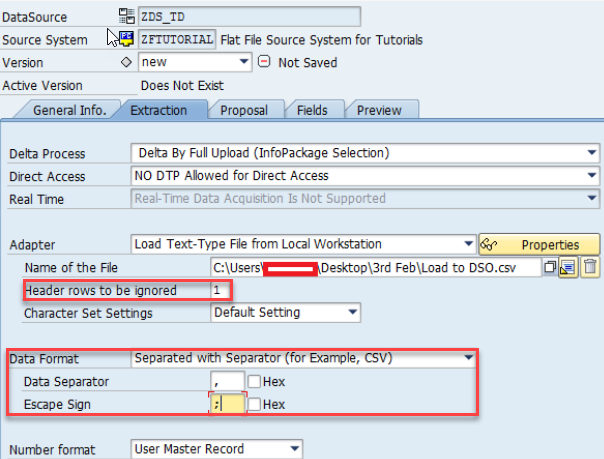
1. **CSV files** - When you load data from a CSV file, you should specify the data separator and the escape character.
2. **ASCII files** - When you load data from an ASCII file, you should specify the fixed data record length.
3. **Binary files** - When you load data from a binary file, you should mention the character record settings.
4. **Text-type files** - When you load data from text type files, you should mention the header rows (which will be ignored) and character record settings.

# SAP BW Flat File Loading to DSO - 17/03/2019

<https://erproof.com/bi/sap-bw-training/sap-bw-flat-file-loading-to-dso/>

SAP BW allows you to define custom DSO’s based on any of the 3 types – Standard | Write-Optimized | and Direct Update DSO. If you have a completely new requirement to load data from SAP BW flat file to DSO - you will have to create a customized data flow and load data accordingly.

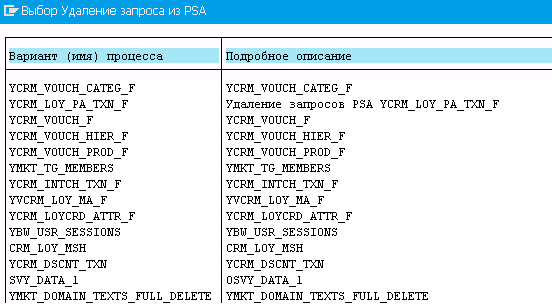
1. Create Flat File Source System.
2. Create Transaction Data DataSource.



1. Create Transformation (Source - Transaction Data DataSource and Target - DSO).
2. Create InfoPackage and Data Transfer Process (DTP) for loading data.

# Insert data into custom tables – Flat file loads to SAP HANA

<http://teachmehana.com/loading-data-flat-file-sap-hana/>



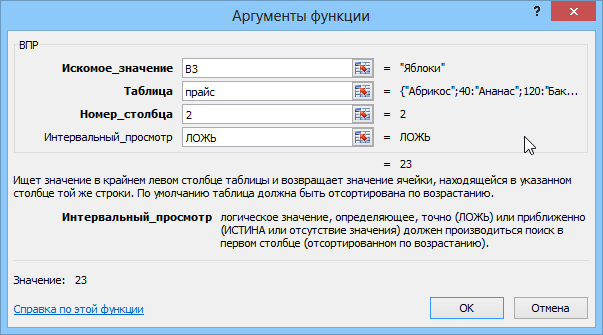
# Excel

# *Использование функции ВПР (VLOOKUP) для подстановки значений*

<https://www.planetaexcel.ru/techniques/2/106/>

Дайте диапазону ячеек прайс-листа собственное имя - выделите все ячейки прайс-листа кроме "шапки" (G3:H19), выберите в меню ***Вставка - Имя - Присвоить (Insert - Name - Define)*** или нажмите **CTRL+F3** и введите любое имя (без пробелов), например *Прайс*. Теперь в дальнейшем можно будет использовать это имя для ссылки на прайс-лист.

Теперь используем функцию **ВПР**. Выделите ячейку, куда она будет введена (D3) и откройте вкладку **Формулы - Вставка функции (Formulas - Insert Function)**. В категории **Ссылки и массивы (Lookup and Reference)** найдите функцию **ВПР (VLOOKUP)** и нажмите **ОК**. Появится окно ввода аргументов для функции



Заполняем их по очереди

* **Искомое значение (Lookup Value)** - то наименование товара, которое функция должна найти в крайнем левом столбце прайс-листа. В нашем случае - слово "Яблоки" из ячейки B3.
* **Таблица (Table Array)** - таблица из которой берутся искомые значения, то есть наш прайс-лист. Для ссылки используем собственное имя "Прайс" данное ранее. Если вы не давали имя, то можно просто выделить таблицу, но не забудьте нажать потом клавишу **F4**, чтобы закрепить ссылку знаками доллара, т.к. в противном случае она будет соскальзывать при копировании нашей формулы вниз, на остальные ячейки столбца D3:D30.
* **Номер\_столбца (Column index number)**- порядковый номер (не буква!) столбца в прайс-листе из которого будем брать значения цены. Первый столбец прайс-листа с названиями имеет номер 1, следовательно нам нужна цена из столбца с номером 2.
* **Интервальный\_просмотр (Range Lookup)** - в это поле можно вводить только два значения: ЛОЖЬ или ИСТИНА
  + 1. Если введено значение **0** или **ЛОЖЬ (FALSE)**, то фактически это означает, что разрешен поиск только точного соответствия, т.е. если функция не найдет в прайс-листе укзанного в таблице заказов нестандартного товара (если будет введено, например, "Кокос"), то она выдаст ошибку #Н/Д (нет данных).
    2. Если введено значение **1** или **ИСТИНА (TRUE)**, то это значит, что Вы разрешаете поиск не точного, а приблизительного соответствия, т.е. в случае с "кокосом" функция попытается найти товар с наименованием, которое максимально похоже на "кокос" и выдаст цену для этого наименования. В большинстве случаев такая приблизительная подстановка может сыграть с пользователем злую шутку, подставив значение не того товара, который был на самом деле! Так что для большинства реальных бизнес-задач приблизительный поиск лучше не разрешать. Исключением является случай, когда мы ищем числа, а не текст - например, при расчете [Ступенчатых скидок](http://www.planetaexcel.ru/tip.php?aid=39).

Осталось нажать **ОК** и скопировать введенную функцию на весь столбец.

### Ошибки #Н/Д и их подавление

Функция **ВПР (VLOOKUP)** возвращает ошибку #Н/Д (#N/A) если

* Включен точный поиск (аргумент **Интервальный просмотр=0**) и искомого наименования нет в **Таблице**.
* Включен приблизительный поиск (**Интервальный просмотр=1**), но **Таблица**, в которой происходит поиск не отсортирована по возрастанию наименований.
* Формат ячейки, откуда берется искомое значение наименования (например B3 в нашем случае) и формат ячеек первого столбца (F3:F19) таблицы отличаются (например, числовой и текстовый). Этот случай особенно характерен при использовании вместо текстовых наименований числовых кодов (номера счетов, идентификаторы, даты и т.п.) В этом случае можно использовать функции **Ч** и **ТЕКСТ** для преобразования форматов данных. Выглядеть это будет примерно так

=ВПР(ТЕКСТ(B3);прайс;0).

* Функция не может найти нужного значения, потому что в коде присутствуют пробелы или невидимые непечатаемые знаки (перенос строки и т.п.). В этом случае можно использовать текстовые функции **СЖПРОБЕЛЫ (TRIM)** и **ПЕЧСИМВ (CLEAN)** для их удаления

=ВПР(СЖПРОБЕЛЫ(ПЕЧСИМВ(B3));прайс;0)   
=VLOOKUP(TRIM(CLEAN(B3));прайс;0)

Для подавления сообщения об ошибке **#Н/Д (#N/A)** в тех случаях, когда функция не может найти точно соответствия, можно воспользоваться функцией **ЕСЛИОШИБКА (IFERROR)**. Так, например, вот такая конструкция перехватывает любые ошибки создаваемые ВПР и заменяет их нулями:

=ЕСЛИОШИБКА(ВПР(B3;прайс;2;0);0)