**Работа с файлами в фоне**

sap bw копирование локальных файлов на сервер приложений в фоновом режиме

sap bw copying local files to the application server in the background

**Accessing Local File in Background Mode**

<https://blogs.sap.com/2018/10/07/accessing-local-file-in-background-mode/>

## **FTP file transfer in Background**

<http://saptechnical.com/Tutorials/ABAP/FTP/Index.htm>

# 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>

### **Introduction**

In this article, we will explore how the process of loading flat files into SAP BW can be fully automated by using an FTP server on which the files are being stored. The reason for an FTP server, rather than *granting the right to put files on the SAP BW server*, is that we can keep control of what files end up on the SAP BW machine and we *minimize the risk of running out of disk space or security risks*.

We will explain how SAP BW can access the FTP server and copy the files unto its own application server.

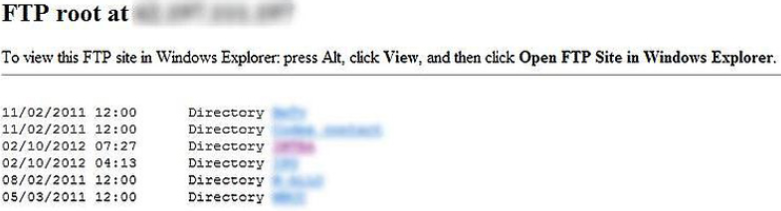
### **Flat files on an FTP server**

The **F**ile **T**ransfer **P**rotocol basically facilitates the downloading and uploading of files on the internet. Some of the benefits of using FTP are:

* Transferring large files can be done more efficiently.
* Automatic resumption of file downloads or uploads without a separate download manager.
* User authentication and secure transfers with encryption protocols.

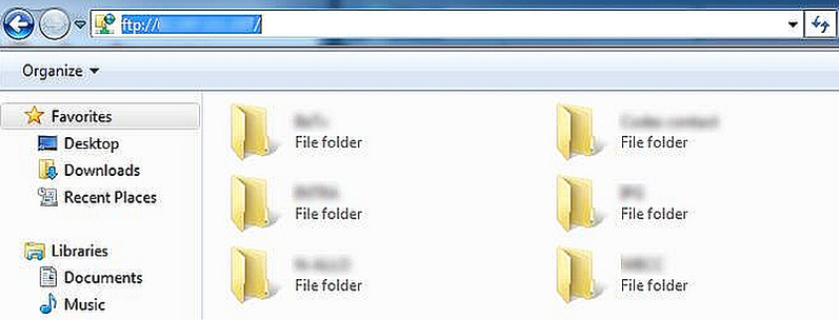
To access an FTP server via your Internet Browser you will need the host name of the server. Its address will be similar to a website address, except instead of "http://”, it will begin with "ftp://”.

*Image 1 - FTP access via Internet Explorer*



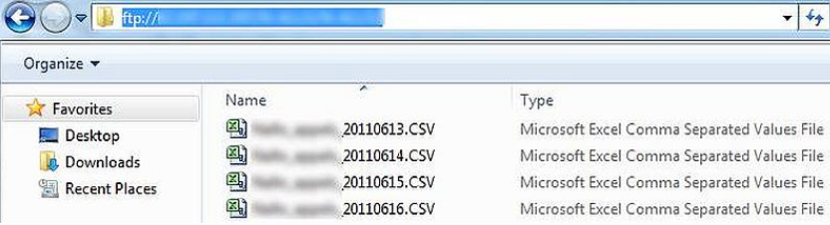
Alternatively, you can enter the same address in your Windows Explorer and use the Windows functionalities to drag and drop, organize, open, rename… the files.

*Image 2 - FTP access via Windows Explorer*



In this article we will discuss ABAP code that takes into account the date at which the flat file was generated. The naming convention used for the flat file is - *flatfilenameYYYYMMDD.CSV*. In our example we will be loading files the day after they were generated and posted on the FTP server.

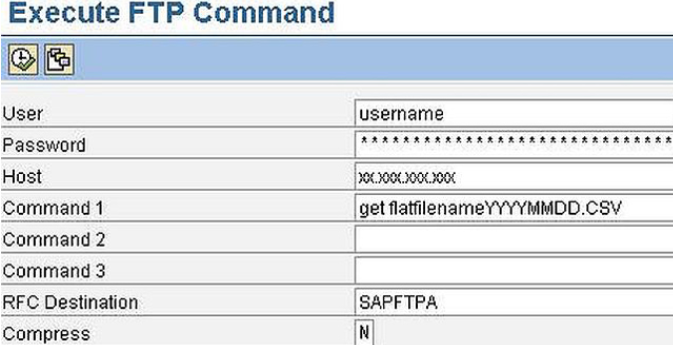
*Image 3 - Our naming convention includes the date the file was generated*



### **Accessing an FTP server from within SAP BW**

Fortunately, SAP has foreseen a standard ABAP program to be able *to communicate with an FTP server*. This program is called ***RSFTP002*** and can be executed using transaction code SE38.

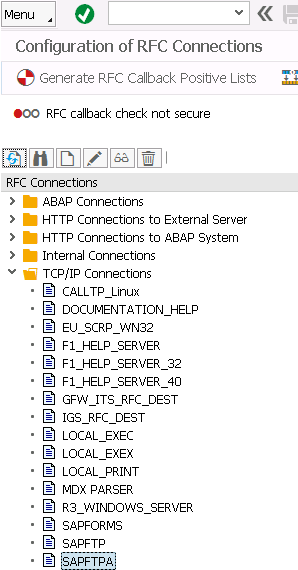
*Image 4 - Program RSFTP002*



The default setting for the RFC destination is **SAPFTP** - this destination is used *to send a file residing locally on the user’s client directly to a remote server without technically pulling the file through the SAP application server*. However, when you want to transfer files to an SAP application server from a remote system, you need to *use the* ***SAPFTPA*** *destination*.

Rem

Посмотреть настройки всех доступных RFC-соединений /и в частности SAPFTPA/ можно в тр. **sm59**



In *Appendix A* we give the ABAP code of a program that is capable of connecting to an FTP server and transferring a file from that FTP server to the SAP BW application server.

get test\_.txt

BELARUSNEFT\a.ekimenko

shm21987.

get Common\np\_test.txt

[\\aup-iis\Ftpserver\Common](file:///\\aup-iis\Ftpserver\Common)

192.168.11.40

ftp://BELARUSNEFT%5Ca.ekimenko@ftp.it.beloil.by/Common/

ftp> *open aup-iis*

Connected to aup-iis.belarusneft.beloil.by.

220

200 OPTS UTF8 command successful - UTF8 encoding now ON.

User (aup-iis.belarusneft.beloil.by:(none)): *a.ekimenko*

331 Password required for a.ekimenko.

Password:

230-Directory has 8,998,223,872 bytes of disk space available.

230 User logged in.

ftp>

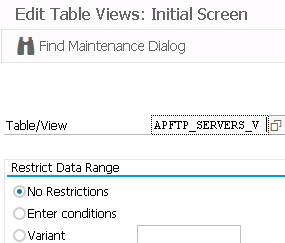
Ошибка

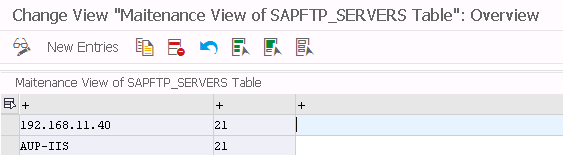
1. *Program RSFTP002* User a.ekimenko has no access authorization for computer aup-iis

Решение

<https://launchpad.support.sap.com/#/notes/2072995>

Using transaction **SM30**, the view SAPFTP\_SERVERS\_V can be used to maintained entries in table SAPFTP\_SERVERS





1. *RSFTP002* Attempt to set up connection to 192.168.11.40 failed

### **Where are my flat files located within BW after the transfer?**

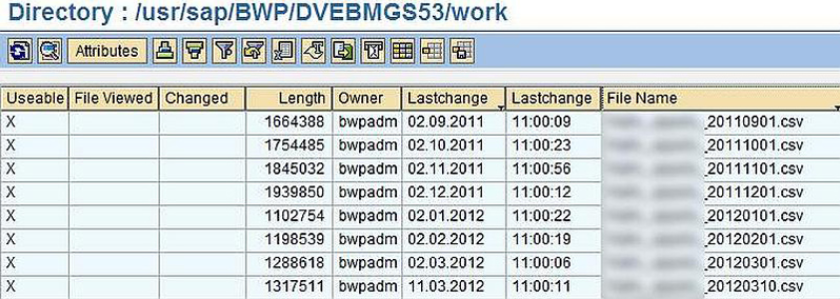
Once you have transferred the flat files from the FTP server onto the BW application server, you can find them in the SAP home directory /**DIR\_HOME**/. This directory can be accessed using transaction code AL11.

*Image 5 - SAP directories using AL11*



Here you have a clear overview of the files that have been copied from the FTP server. You can also find useful information like the time and date the flat file was put onto the BW application server as well as the size of the files. *You can even have a look at the actual data in the file by double clicking it.*

*Image 6 - Examples of files transferred to the home directory*



### **Loading flat files from the BW application server into the PSA**

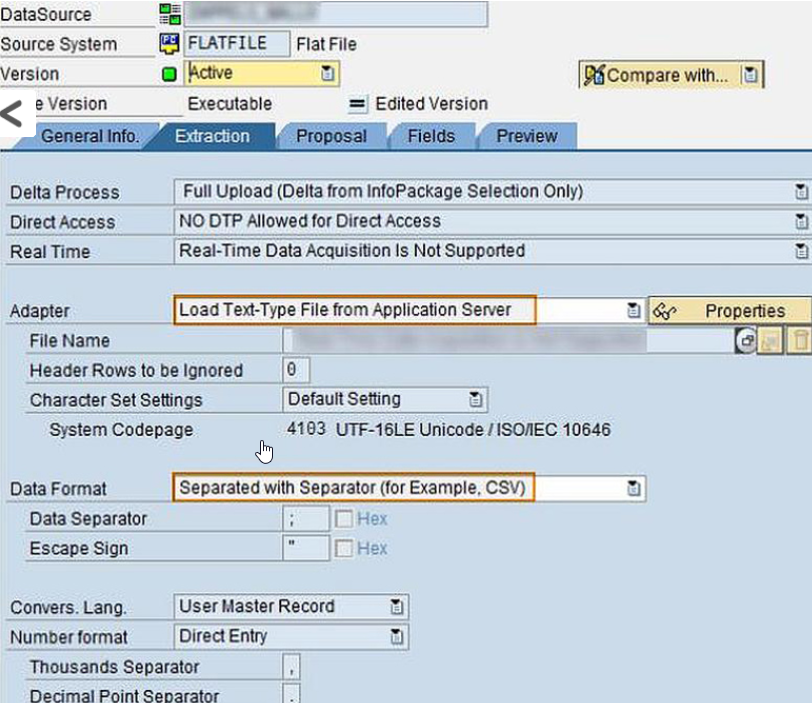
Now that your flat files are available on the BW application server, you would like to load them into the Persisting Staging Area /PSA/ of BW. Once the data is located in the PSA you can load it further on into DSO’s, InfoCubes, etc.

First you have to define a DataSource suitable for loading the type of data in the flat file. Make sure you create this DataSource in the Source System that was created for flat file loading. Then make the correct selections in the fields below

* Adapter = Load Text-Type File from Application Server
* Data Format = Separated with Separator /for Example, CSV/

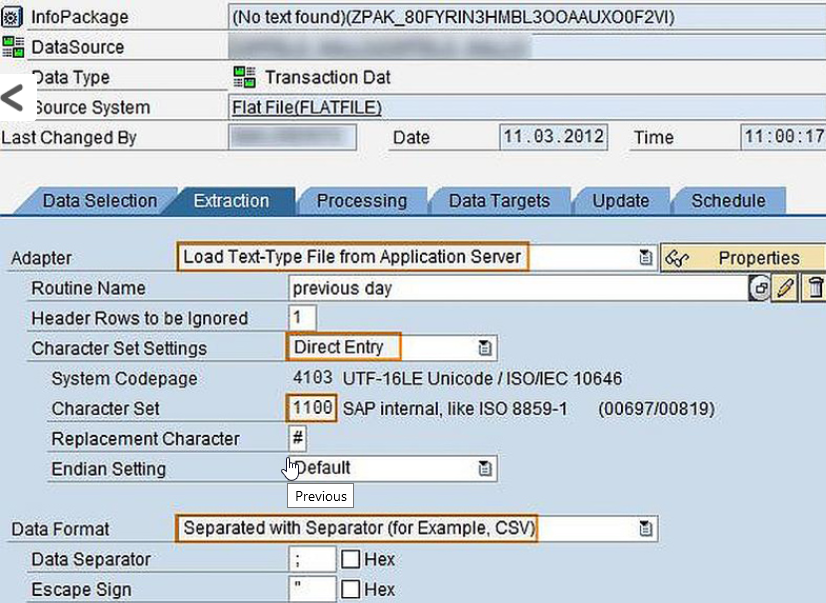
You should also carefully define the properties of each individual field. This can be done in the ***Fields*** tab. Finally, you should test your DataSource by using the ***Read Preview Data*** button which can be found in the ***Preview*** tab.

*Image 7 - DataSource for flat file loading from application server*



Once the DataSource has been created you can move on to the next step - ***creating an InfoPackage*** for carrying out the actual data load from the flat file into the PSA. As you could have guessed, the settings with respect to the *Adapter* and *Data Format* fields required for the DataSource are also required here. Furthermore, it is also important to set the *Character Set Settings* to *Direct Entry* and to choose *1100* as the *Character Set*. *1100 is the code page used by the default installation.*

*Image 8 - InfoPackage for flat file loading from application server*



Note that in the InfoPackage in Image 8, the path towards the flat file to be loaded is determined in a routine. This is done because the flat file name includes a date which changes every day. Assume that every night, a flat file generated the day before is loaded. The code for the path will look like

data *previous\_day* type sy-datum.

previous\_day = sy-datum - 1.

concatenate '/usr/sap/BWP/DVEBMGS53/work/*flatfilename*' *previous\_day* '.csv'

into p\_filename.

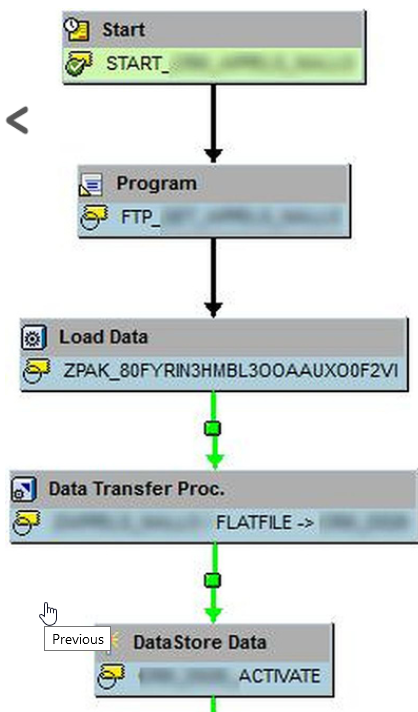
p\_subrc = 0.

The path for the home directory in which the flat file is located can be found using transaction code AL11. /See Image 5./

### **Bringing the pieces together in a process chain**

Obviously, you would like to limit the manual actions required for loading the flat files into SAP BW to an absolute minimum. For that purpose, you can integrate the program presented in ‘Appendix A’ and the InfoPackage for flat file loading described in the previous section into a process chain that also contains all the steps necessary for loading the flat file data from the PSA into DataStore Objets, InfoCubes, InfoObjetcs, etc.

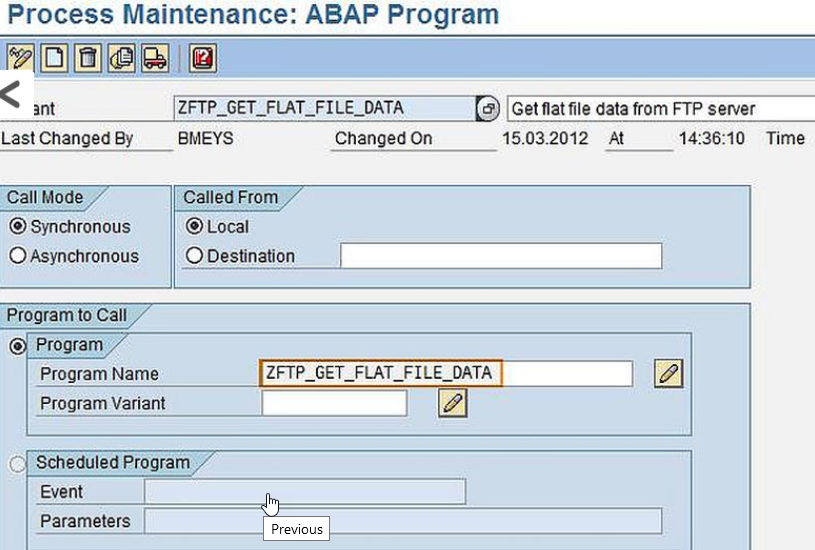
*Image 9 - Process chain including program for retrieving flat file from FTP*



As shown in *Image 9*, the second step of the process chain includes the ABAP program for retrieving the flat file from the FTP server and copying it to SAP BW whereas step 2 represents the InfoPackage to actually load the flat file data from the SAP BW application server into the PSA. The ensuing steps can be standard processes to load data into a DSO, activating it, loading it into an InfoCube and so on.

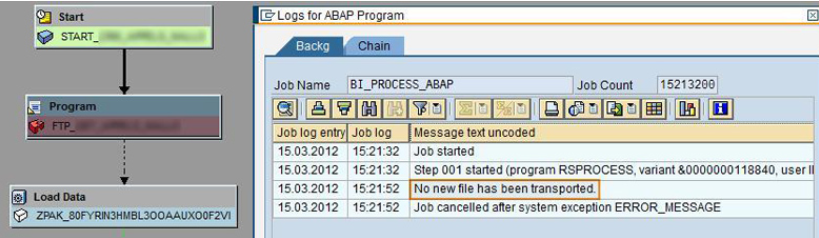
The Image below shows how the execution of an ABAP program can be included as a step in a process chain. Using transaction code RSPC, you can include such a step by selecting the ***ABAP program*** process type under the *General Services* folder. All you need to do then is *give the variant a proper name and select the program of Appendix A as the program to call*. In ideal circumstances, this process chain will be scheduled to run during the night, so that by morning all flat file data is available in the InfoCubes and ready to be used in reports.

*Image 10 - Including an ABAP program in a process chain*



If you want your custom ABAP program to generate meaningful error messages, you will need to write some extra code yourself.

*Image 11 - Error message when no new flat file is found*



### **APPENDIX A**

*In part 1* of the code a connection is made with the FTP server. The correct directory is selected and a ***get*** command is executed to fetch the flat file from the server. The name of the flat file has been constructed using the system date. Make sure that whoever is responsible for the FTP server gives you the host address, a username for BW to access the server as well as a password.

*Part 2* provides some code to generate a meaningful error message in case something goes wrong. Since this is not the main focus of this paper, we will not go into detail. Obviously you can use this code as an inspiration for generating your own error messages.

*Part 3* makes sure that the flat file that was loaded during the previous run of our daily process chain is removed from the FTP directory. If you want to keep a full history of your data on the FTP server, this clean up may not be desired. However, if no historic files are to be kept on the FTP server, you might as well delete them for the sake of recovering disk space.

Something similar can be done for the flat files that have been copied to the SAP BW application server and are now stored in the SAP home directory /DIR\_HOME/. In our case, *we want to keep every flat file that was generated on the first day of the month. All the other files may be deleted*, which helps keeping the home directory clean.

report **zftp\_get\_flat\_file\_data**.

data:

list\_tab type table ofabaplist,

listtxt type list\_string\_table,

file\_name(25) typec,

command\_rename(69) type c,

command\_get(29) typec,

command\_dele(30) type c,

transfer\_ok(30) typec,

ftp\_code(3) typec,

date1 type sy-datum,

date2 type sy-datum,

day type clength 2.

date1 = sy-datum - 1.

date2 = date1 - 1.

concatenate '*flatfilename*\_' date1 '.csv' into *file\_name*.

concatenate 'get' file\_name into command\_get *separated by space*.

***\*\*\*\*\* part 1 \*\*\*\*\****

submit rsftp002

with cmd1 = 'cd *directory\_name*'

with cmd2 = command\_get

with compress = 'N'

with dest = 'SAPFTPA'

with host = *'xx.xxx.xxx.xxx*'

with pwd = '*password*'

with user = '*username*'

and return

exporting list to memory.

***\*\*\*\*\* part 2 \*\*\*\*\****

call function 'LIST\_FROM\_MEMORY'

tables

listobject = list\_tab

exceptions

not\_found = 1

others = 2.

if sy-subrc = 0.

call function 'LIST\_TO\_ASCI'

importing

list\_string\_ascii = listtxt

tables

listobject = list\_tab.

if sy-subrc 0.

message idsy-msgid type sy-msgty number sy-msgno

with sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.

endif.

read tablelisttxt into transfer\_ok index 8.

ftp\_code = transfer\_ok+0(3).

if ftp\_code '226'.

message e051(rsar) with'No new file has been transported.'.

endif.

endif.

***\*\*\*\*\* part 3 \*\*\*\*\****

clear file\_name.

concatenate '*flatfilename*\_'date2 '.csv' into file\_name.

concatenate 'dele' file\_name into command\_dele separated by space.

submit rsftp002

with cmd1 = 'cd *directory\_name*'

with cmd2 = command\_dele

with compress = 'N'

with dest = 'SAPFTPA'

with host = *'xx.xxx.xxx.xxx'*

with pwd = *'password'*

with user = *'username'*

and return

exporting list to memory.

***\*\*\*\*\* part 4 \*\*\*\*\****

day = date2+6(2).

if day ne '01'.  
delete datasetfile\_name.  
endif.

## **Handling Files on Application Server in SAP - DATASET Statement and other Tips**

<https://sap4tech.net/handling-files-application-server-sap/>