**Работа с файлами**

[Содержание](#Содержание)

[Операторы файлового интерфейса AS](#Операторы_файлового_интерфейса_AS)

[Upload File from AS using ABAP](#UPLOAD_FILE_FROM_AS_USING_ABAP)

[How to read/write text file from your local machine](#How_read_write_file_on_local_machine)

[Выгрузка файла на локальный сервер в фоновой программе](#Выгрузка_на_лок_сервер_в_фоне)

[Windows ftp cmd](#Windows_ftp_cmd)

**У нас – юникоидная система!**

Решение

**ftp → internal table → ADSO**

[Содержание](#Содержание)

## **При использовании операторов файлового интерфейса** – OPEN DATASET **|** TRANSFER | READ DATASET | GET DATASET | SET DATASET | TRUNCATE DATASET | CLOSE DATASET | DELETE DATASET – файлы создаются на сервере приложений в директории *DIR\_HOME!*

SAP рекомендует

* всегда использовать UTF-8 при работе с текстовыми файлами.
* указывать дополнение linefeed **WITH { NATIVE | SMART| UNIX| WINDOWS }.**
* При открытии UTF-8 файлов на чтение - пропускать символ порядка байт - *SKIPING BYTE-ORDER MARK.*
* При записи в UTF-8 файл - записывать метку с дополнением *WITH BYTE-ORDER MARK*.

From

<https://abap-blog.ru/files/fajlovyj-interfejs-operacii-s-fajlami/>

**Для поддержки работы с юникодом** есть класс **CL\_ABAP\_FILE\_UTILITIES**. Некоторые его методы

* CHECK\_FOR\_BOM – проверить наличие в файле метки [BOM](http://ru.wikipedia.org/wiki/Byte_order_mark).
* CHECK\_UTF8 – *хранится ли файл в UTF-8*. MAX\_KB – определяет размер KB при сканировании.
* CREATE\_UTF8\_FILE\_WITH\_BOM – создание файла в UTF-8 с меткой.
* CHECK\_XSTRING\_UTF8 – проверить последовательность на наличие UTF-8 кодировки.
* CHECK\_STRING\_7BIT\_ASCII – *содержит ли строка только ASCII символы*.

## **Операторы файлового интерфейса** – файлы создаются на сервере приложений в директории *DIR\_HOME*

* OPEN DATASET **dset FOR *access* IN *mode* [position] [os\_additions] [error\_handling].**
* TRANSFER
* READ DATASET
* GET DATASET
* SET DATASET
* TRUNCATE DATASET
* CLOSE DATASET
* DELETE DATASET

Если имя файла приходит откуда-то извне - необходимо его **проверить** чтобы не подвергать систему **риску связанному с обходом** [**доступа к директориям**](http://help.sap.com/abapdocu_740/en/abendyn_file_scrty.htm)

CALL FUNCTION ***'FILE\_VALIDATE\_NAME'***

  EXPORTING

    logical\_filename           = 'EXAMPLE\_FILE'

  CHANGING

    physical\_filename          = file

  EXCEPTIONS

    logical\_filename\_not\_found = 1

    validation\_failed          = 2

    OTHERS                     = 3.

IF sy-subrc = 0.

  OPEN DATASET file FOR OUTPUT IN BINARY MODE.

ELSE.

  cl\_demo\_output=>display( |Error return code { sy-subrc }| ).

ENDIF.

***Атрибут*** **LEGACY**

**LEGACY BINARY MODE [*endian] [CODE PAGE cp]***

**LEGACY TEXT MODE *[endian] [CODE PAGE cp] [linefeed]***

Используется при необходимости обеспечить совместимость с не юникодными системами

* можно задать кодовую страницу - атрибут CODE PAGE
* порядок байт - атрибут endian.

В режиме LEGACY TEXT MODE пробелы в символьных (clike) переменных или плоских структурах обрезаются.

***Атрибут*** **TEXT MODE**

TEXT MODE ***encoding [linefeed]***

***Пробелы***

Если переменная является символьной - clike или плоской структурой - пробелы на концах обрезаются. Если используется строковый тип - пробелы не обрезаются.

***Символ конца строки***

При записи данных - по умолчанию вставляется *зависимый от платформы* символ конца строки.

***linefeed***

Обработка символов конца строки зависит от дополнения linefeed

* Для *юникодных* программ могут быть использованы только символьные типы + обязательное указание ***encoding***.
* В*не юникодных* - дополнение ***encoding*** не обязательно - при перемещении указателя происходит перемещение на указанное кол-во символов - не байт.

***Считывание/ запись байтовых | числовых переменных***

Только символьные переменные могут быть считаны или записаны в файл. ***В не юникодных системах*** допускается считывание и запись из байтовых или числовых переменных.

#### **Дополнение** **linefeed**

#### linefeed **WITH { NATIVE | SMART| UNIX| WINDOWS } - о**бработка символов конца строки.

* **NATIVE** - в зависимости от ОС на сервере приложений.

При использовании *linefeed* профильный параметр **abap/NTfmode** - тр. RZ11 игнорируется.

SAP рекомендует указывать дополнение linefeed - иначе логика определения указателя конца строки будет зависеть от множества факторов - ОС сервера приложений | профильного параметра | существующего символа в имеющемся файле.

#### **Дополнение CODEPAGE cp …**

В *юникодных* системах происходит преобразование символов из указанной кодовой страницы в юникодные символы в соответствии с текущим форматом на сервере приложений.

В качестве параметра **cp** необходимо использовать переменную со значением из столбца CPCODEPAGE в таблице TCP00 –



Пример – чтение с сервера файла записанного в кодировке cp1251 - кириллица Windows -

OPEN DATASET FILE FOR OUTPUT IN *LEGACY* text mode CODE PAGE '1504'.

SAP рекомендует всегда использовать UTF-8 при работе с текстовыми файлами – т.к.к кодовые страницы не всегда можно идентифицировать относительно содержимого файла.

* **DEFAULT** – в юникодных системах - UTF-8; в не юникодных - NON-UNICODE.
* **UTF-8** [SKIPPING|WITH BYTE-ORDER MARK] – файл обрабатывается в соответствии с кодировкой [UTF-8](http://ru.wikipedia.org/wiki/UTF-8). С помощью дополнения указывается - необходимо ли обрабатывать символ порядка байт - [**BOM**](http://ru.wikipedia.org/wiki/Byte_order_mark)
  + *SKIPING BYTE-ORDER MARK* – используется при режимах INPUT/ UPDATE. Если в файле присутствует символ BOM - указатель переводится на позицию после него.
  + *WITH BYTE-ORDER MARK* – используется при режиме OUTPUT. Символ BOM вставляется в начало файла.
* *NON-UNICODE* – для *не юникодных* систем данные переносятся без конвертации. В *юникодной* системе происходит преобразование в соответствии с настроенным присвоением - таблица TCP0C.

Рекомендации SAP

* При открытии UTF-8 файлов на чтение - пропускать символ порядка байт - *SKIPING BYTE-ORDER MARK.*
* При записи в UTF-8 файл - записывать метку с дополнением *WITH BYTE-ORDER MARK*.
* С помощью класса CL\_ABAP\_FILE\_UTILITIES и его метода  CREATE\_UTF8\_FILE\_WITH\_BOM можно создать UTF-8 файл с меткой BOM. С помощью метода CHECK\_UTF8 можно определить есть ли метка в указанном файле.

**Upload File from SAP Application Server using ABAP**

<https://www.saphub.com/files/upload-file-sap-application-server/>

[Содержание](#Содержание)

Use the following steps to upload data from a file in SAP application server to [ABAP internal table](https://www.saphub.com/abap-tutorial/what-is-an-abap-internal-table-and-how-to-create-it/).

1. Declare a ABAP internal table.
2. Use OPEN DATASET - to open file on appliction server.
3. Use READ DATASET - to read each line in the file to workarea. Append work area data to internal table.
4. Use CLOSE DATASET - to close the application server file.
5. Process the data in the internal table.

*\* Data Decalaration*

DATA:

gt\_spfli TYPE TABLE OF spfli,

gwa\_spfli TYPE spfli.

gv\_file TYPE rlgrap-filename.

*\* START-OF-SELECTION*

PERFORM read\_file.

PERFORM display\_data.

*\* Form read\_file*

FORM ***read\_file***.

DATA: lv\_data TYPE string.

*\*Move complete path to filename*

gv\_file = 'spfli.txt'.

*\*Open the file in application server to read the data*

OPEN DATASET gv\_file FOR INPUT IN TEXT MODE ENCODING DEFAULT.

IF sy-subrc NE 0.

MESSAGE 'Unable to open file' TYPE 'I'.

ENDIF.

DO.

*\* Loop through the file - if a record is found move it to temporary structure else exit out of the loop.*

READ DATASET gv\_file INTO lv\_data.

IF sy-subrc = 0.

*\* Split the fields in temporary structure to corresponding fields in workarea.*

SPLIT lv\_data AT ',' INTO gwa\_spfli-carrid

gwa\_spfli-connid

gwa\_spfli-countryfr

gwa\_spfli-cityfrom

gwa\_spfli-airpfrom

gwa\_spfli-countryto

gwa\_spfli-cityto

gwa\_spfli-airpto

gwa\_spfli-arrtime.

APPEND gwa\_spfli TO gt\_spfli.

CLEAR gwa\_spfli.

ELSE.

EXIT.

ENDIF.

ENDDO.

*\* Close the file*

CLOSE DATASET gv\_file.

ENDFORM.  *" read\_file*

*\* Form DISPLAY\_DATA*

FORM ***display\_data*** .

LOOP AT gt\_spfli INTO gwa\_spfli.

WRITE:/ gwa\_spfli-carrid,

gwa\_spfli-connid,

gwa\_spfli-countryfr,

gwa\_spfli-cityfrom,

gwa\_spfli-airpfrom,

gwa\_spfli-countryto,

gwa\_spfli-cityto,

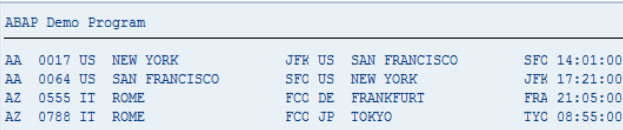
gwa\_spfli-airpto,

gwa\_spfli-arrtime.

CLEAR: gwa\_spfli.

ENDLOOP.

ENDFORM. *" DISPLAY\_DATA*



**FIELD-SYMBOLS** можно определять родовыми типами – any | data | csequence | simple | any table и т.д. ⇒ к ним можно присвоить все что угодно.

FIELD-SYMBOLS <fs\_any\_string> TYPE CSEQUENCE.

# H[ow to read/write text file from your local machine with abap?](https://stackoverflow.com/questions/20330678/how-to-read-write-text-file-from-your-local-machine-with-abap)

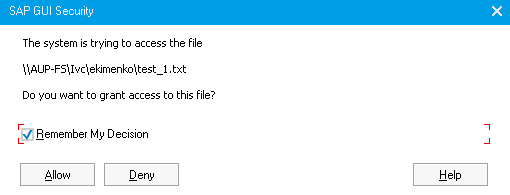
# <https://stackoverflow.com/questions/20330678/how-to-read-write-text-file-from-your-local-machine-with-abap>

[Содержание](#Содержание)

# *To access the front-end* - use the methods of the class CL\_GUI\_FRONTEND\_SERVICES. For an example - check the report GRAPHICS\_IGS\_ZIPPER\_DEMO.

# Считывание файлов с фронтенда

<https://abap-blog.ru/files/upload_from_frontend/>



− При записи с локальной станции - проверяются доступы локальной станции а ***при записи из фона*** - полномочия AS.

## **В**[**ыгрузка файла на локальный сервер в фоновой программе**](https://sapboard.ru/forum/viewtopic.php?f=13&t=20127&start=0)

[Содержание](#Содержание)

Когда работа идет в фоне - задача выполняется на AS → он должен иметь доступ к пути расположения файла начиная от имени диска.

1. Если ***ОС******AS - Linux*** - дисков с буквенными именами он не знает → удаленный каталог должен быть смонтирован в дерево каталогов AS - это с базисом надо решать.
2. Если ***ОС AS - Windows*** - расшарь на лок. машине каталог и выгружай на него с помощью **open dataset**. WS\_DOWNLOAD – gui- ф-ция ⇒ в фоне работать не может.

Фоновые задания выполняются фоновыми процессами BTC.

Фоновый процесс крутится на AS ⇒ все что Вы делаете с фронтэндом в диалоге работать не будет - ALV | выгрузки в Excel и пр.

− Моя проблема решилась так - попросила сделать ftp к расшаренной папке куда мне нужно было кидать файлики в фоне ⇒ теперь перекидываю файлики с APS по ftp в фоне.

− Решила задачу без промежуточного выкладывания файла на AS - в программе формирую внутреннюю табличку и на прямую кидаю ее на ftp через *FUNCTION 'FTP\_R3\_TO\_SERVER'*. Были проблемы с кодировкой русских символов - если переносить в char mode - решила конвертацией в бинарный вид. На FTP сохраняется xls файл.

− It is not possible to write to a client while in background processing.  
The nature of background processing is that no client machine has to be connected to the WAS.

If the creation of the file is time consuming you could trigger the creation of the file on the WAS and then offer a list of files to download somewhere in your BSP application.

***If you just specify the file name*** - the file is put into the directory specified by profile parameter DIR\_HOME. You can check/edit the profile parameters in RZ11 or view the directories in AL11.

For general information on working with files in ABAP you can refer to <http://wiki.sdn.sap.com/wiki/display/ABAP/Working+with+files>.

Generally ***downloading data is achieved by setting the right http header fields and pushing the binary data into the http response***.

You should take a look at cl\_bsp\_utility=>download.

This sets the right content headers in your response. You have to specify your data as an XSTRING and specify which Content-Type and Content-Disposition you want.

***Content-Type*** is the mime type of the file - so for an Excel file that could be application/vnd.ms-excel or you could use something general like application/octetstream.

***Content-Disposition*** can be used to tell the browser the default filename to use. An example value could be attachment; filename=filexyz.xls.

Be aware that browsers could have slightly different behavior based on the combination of Content-Type and Content-Disposition - i.e. offering a 'Save as...' dialog instead of showing it inline.

## В[ыгрузка файла на лок. сервер в фоновой программе](https://sapboard.ru/forum/viewtopic.php?f=13&t=20127&start=0) - решения

* + 1. ***Если ОС AS и клиентской машины - Windows*** - можно сопоставить путь Windows в AL11 и сохранить файл там.

**Rem** - чтобы разрешить такой трафик - нужно *открыть целевой брандмауэр Windows*.

* + 1. Запустить FTP-сервер на целевом компьютере - SAP имеет ФМ для решения этой проблемы. Посмотрите на *группу ф-ций SFTP*.
    2. ***Если ОС обих машин - Linux*** – мой вариант – на целевой машине можно использовать [**SCP**](https://en.wikipedia.org/wiki/Secure_copy) т.к. большинство Unix-систем будут включать ssh и следовательно - scp.

Предварительно

1. Создать внешнюю команду в SM49.
2. Настроить аутентификацию с открытым ключом с AS на целевой компьютер. Сделать это сложнее если у вас нет прав администратора на AS - но есть способы обойти это.

## The *Secure Copy Protocol* - **SCP** is a [network protocol](https://en.wikipedia.org/wiki/Network_protocol) based on the [BSD](https://en.wikipedia.org/wiki/BSD) [RCP](https://en.wikipedia.org/wiki/Rcp_(Unix)) protocol which supports [file transfers](https://en.wikipedia.org/wiki/File_transfer) between hosts on a network. SCP uses [*Secure Shell*](https://en.wikipedia.org/wiki/Secure_Shell)- **SSH** for data transfer and uses the same mechanisms for authentication thereby ensuring the [authenticity](https://en.wikipedia.org/wiki/Information_security#Authenticity) and [confidentiality](https://en.wikipedia.org/wiki/Confidentiality) of the [data in transit](https://en.wikipedia.org/wiki/Data_in_transit). A client can send/ upload files to a server optionally including their basic attributes – permissions | timestamps. Clients can also request/ download files or directories from a server. SCP runs over [TCP port](https://en.wikipedia.org/wiki/TCP_port) 22 by default.

parameters: FILE\_OUT like **RLGRAP-FILENAME** default '\\server.local\ForAll\list.txt'.

## **SAP RLGRAP Structure** fields within data Dictionary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Component Type** | **Data Type** | **length - Decimals** | **Description** |
| FILENAME | [LOCALFILE](https://www.se80.co.uk/dataelements/l/loca/localfile.htm) | CHAR | 128 | Local file for upload/download |
| FILETYPE | [FILEFORMAT](https://www.se80.co.uk/dataelements/f/file/fileformat.htm) | CHAR | 3 | Transfer file format (upload/download) |
| FILEMODE | [FILEMODE](https://www.se80.co.uk/dataelements/f/file/filemode.htm) | CHAR | 1 | Type of data transfer to local file |
| PRINTER | [G\_PRINTER](https://www.se80.co.uk/dataelements/g/g_pr/g_printer.htm) | CHAR | 4 | Printer for SAP Graphics |
| FILEFM\_UL | [FILEFM\_UL](https://www.se80.co.uk/dataelements/f/file/filefm_ul.htm) | CHAR | 3 | Format of the transfer data (uploads only) |

**Windows ftp cmd**

[Содержание](#Содержание)

>ftp **open** 10.93.236.36 *21*

user branches\e000

password bYw2h9

**cd** /transit

**cd** 650

**cd** ..

**cd** temp

Команды

* ***CLOSE*** или DISCONNECT - закрыть соединение с текущим FTP сервером.
* ***BYE*** или QUIT - закрыть соединение и выйти из утилиты FTP.
* ***DIR***
* ***USER*** - залогиниться на данном сервере под указанным пользователем -Вы должны быть уже подключены.

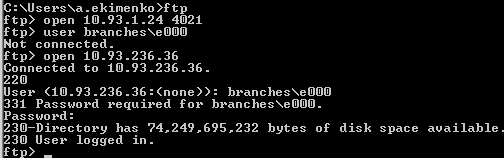
ftp> user myusername

* ***CD*** - перейти в указанную папку на сервере.

ftp> cd ../another/folder

## **FTP file transfer in Background**

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



# How to connect ABAP to FTP

<https://www.saptutorial.org/connect-abap-ftp/>

**1.** Before you use FTP\_CONNECT function module to connect SAP to FTP - you need to Scramble ('skræmbl - шифровать) FTP Password with HTTP\_SCRAMBLE Function module.

gv\_slen = strlen( lv\_password ).

CALL FUNCTION ***'HTTP\_SCRAMBLE'***

EXPORTING

source = lv\_password

sourcelen = gv\_slen

key = gv\_key

IMPORTING

destination = lv\_password.

Key value is the key for scramble your password - you can entered anything value.

**2.** Next, you need to call FTP\_CONNECT function module to create connection from SAP System to FTP.

CALL FUNCTION ***'FTP\_CONNECT'***

EXPORTING

user = fu\_cuser

password = lv\_password

host = fu\_chost

rfc\_destination = fu\_ftp

IMPORTING

handle = gv\_hdl

EXCEPTIONS

not\_connected = 1

OTHERS = 2.

**3.** In this step you can use FTP\_COMMAND to execute any of FTP command - for this sample we will run FTP\_COMMAND to transfer file from SAP to FTP.

CONCATENATE 'put' fu\_src lv\_destination INTO gv\_cmd SEPARATED BY space.

CALL FUNCTION ***'FTP\_COMMAND'***

EXPORTING

handle = gv\_hdl

command = gv\_cmd

TABLES

data = itab

EXCEPTIONS

tcpip\_error = 1

command\_error = 2

data\_error = 3

OTHERS = 4.

# In this FM you need to pass parameter HANDLE. This value you just entered 1 as integer data type –

# COMMAND - this parameter contain your FTP command

# DATA - internal table char.

# 4. The last step you need - to close connection | disconnect FTP connection using FTP\_DISCONNECT.

# CALL FUNCTION *'FTP\_DISCONNECT'*

# EXPORTING

# handle = gv\_hdl

# EXCEPTIONS

# OTHERS = 1.

# CALL FUNCTION *'RFC\_CONNECTION\_CLOSE'*

# EXPORTING

# destination = fu\_ftp

# EXCEPTIONS

# OTHERS = 1.

*\*Transfer file into server*

CONCATENATE fu\_src lv\_destination INTO gv\_cmd SEPARATED BY space.

CALL FUNCTION 'FTP COMMAND'

EXPORTING

handle = gv\_hdl

command = gv\_cmd

TA3LES

data = itab

EXCEPTIONS

tcpip\_error 1

command\_error = 2

data\_error = 3

OTHERS = 4.

IF sy-subrc <> 0.

MESSAGE 'Fail to Upload’ TYPE 'E' DISPLAY LIKE 'I'.

ELSE.

lv\_string = lines( ft\_ztbsd ) .

CONDENSE lv\_string.

CONCATENATE lv\_string ‘rows Successfully Uploaded' INTO lv\_message SEPARATED BY space.

CONDENSE lv\_message.

MESSAGE lv\_message TYPE 'I\*.

ENDIF.

# PERFORM fm\_ftp\_disconnect USING fu\_ftp.

# [The SAPFTP Library - FTP Programming in ABAP](https://wiki.scn.sap.com/wiki/display/ABAP/The+SAPFTP+Library+-+FTP+Programming+in+ABAP)

# <https://wiki.scn.sap.com/wiki/display/ABAP/The+SAPFTP+Library+-+FTP+Programming+in+ABAP>

# **Команда **FTP\_CONNECT****

# Параметры

# PASSWORD = lv\_password - **could be scrambled via the HTTP\_SCRAMBLE function module**

# RFC\_DESTINATION = 1) **SAPFTPA - for background access to FTP Host server; 2) SAPFTP = for front end access to FTP.**

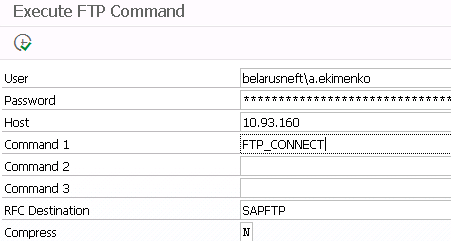
## ***Creating RFC Destinations for FTP communications***

# **Программа **RSFTP005** с атрибутом** *SAPFTP check* - creates following RFC destinations – 1) **SAPFTP; 2) SAPFTPA.**

## ***Testing the RFC destinations***

**Программа RSFTP002.**







## **Checking the Trace of the FTP communication**

**Программа RSFTP001.**



