

Documentation *asci4rfc*



C. GATHMANN

C. Gathmann
Computer-System-Beratung GmbH
Orleansstr. 40
D-28211 Bremen
WWW.CSBG.DE

Version: asci4rfc V 0.2.2 (9.3.2004)

„SAP“ und „R/3“ sind eingetragene Warenzeichen der SAP AG Walldorf
„Microsoft“, „Windows“, „Windows NT“, „Windows 2000“ und „Windows XP“ sind eingetragene Warenzeichen oder
Marken der Microsoft Corporation in den USA und/oder anderen Ländern

Usage

Asci4rfc can call RFC enabled function modules in an SAP R/3 system (or APO, BW).
This includes BAPIs.

Basically it works like *startRFC* program, which you can find on your SAP GUI CD.
The difference is the handling of tables regarding input and output and the possibility to run a script with one login handle.

Syntax

Asci4rfc <script file name> [Arg2 [Arg3] ...]

Syntax of Script:

The script syntax is very simple. All keywords are in capital letters.
A command always starts with “!”, the end of a command is the next “!”.
A variable starts with “\$”.
If you need a “!” or “\$” not as part of a keyword you have to use “\!” or “\\$”.
The command line starts with “!” and ends with the last character before end of file or next “!”.

Keywords:

!REM <string comment>
Can be used for remarks.

!DEBUG=<int Level>
Set the Debug Level, which controls the output of warning, error and other messages.
5 for all messages, 1 for Errors only.
Default is 3.

!PROTFILE=<string filename>
You can specify a filename for output of all messages.
Default is no name, so no protocol file is written.

!FILE=<string file name>[,modus]
For output of tables and export values you can set a output file (optional with path)
Modus can be
“w” to destroy existing file or
“a” to append.

!LOGIN:<string login string>
login string specifies the connection and login to SAP system, see SAP documentation.
A valid string structure is fx:
TYPE=3 CLIENT=<int Client Number> USER=<string SAP user name>
PASSWD=<string password> LANG=< string language, 1 or 2 chars>
ASHOST=<string histname> SYSNR=<int system number, 2 digits>

!REFRESH
Reset all values except connection data given by !LOGIN.

!FM:<string name of Function Module>
Specifies the RFC enabled Function Module to be called, which is called by !RUN command.

Reset by !REFRESH

`!EXPORT:<string export parameter>=<string value>, [<string export parameter>=<string value>, [...]]`

Specifies the export parameter and assigns value for function module defined by !FM command.
Reset by !REFRESH

`!IMPORT:<string import parameter>=<integer length>,[<string import parameter>=<integer length>,...]`

Specifies the import parameter for function module defined by !FM command. Length is the length as defined in SAP for the field.

`!TABLE,<int length>:<string name of table> [<string value of 1st record>,[<string value of 2nd record>,[<string value of 3rd record>,...]]]`

Defines a table for function module call. The length of a record (sum of all fields of this structure) is mandatory. You can specify value for the records. Please mind the structure of a record. Certain types need special initial values like "00000" and so on.

The record is by default filled with spaces, the string "value" is overlaying.

`!RUN`

Starts the RFC call with the IMPORT, EXPORT, TABLE

For many BAPI FMs you need to run a "BAPI_TRANSACTION_COMMIT" afterwards.

`!OUTPUT:TABLE=<string table name>,`

After calling the function module you can write the content of a table to a file defined in the last !FILE command.

`!OUTPUT:IMPORT=<string import parameter name>,[<string import parameter name>,...]`

After calling the function module you can write the value of import parameters to a file defined in the last !FILE command. (Without carriage return)

`!CR`

Write Carriage Return to file given by !FILE

`!WRITE=<string>`

Write string to file given by !FILE (without CR)

`!PAUSE`

Stops until key is pressed.

Variables

\$DATE () = current date (format yyyyymmdd)

\$DATE((<plus minus days>, <plus minus month>, <plus minus year>)
= specified date (format yyyyymmdd)

Values for:

<plus minus days>,
<plus minus month>,
<plus minus year>

Distinct value : raw integer like 1,2,3,4,...30,31

Difference value: sign and integer like -10 or +6

Examples:

\$DATE(01,-1,) --> 1st day of previous month

\$DATE(-5,) today - 5 days

\$DATE(-5,+0,-1) today - 5 days last year

\$NDATE () = current date (format dd.mm.yyyy)

\$NDATE ((<plus minus days>, <plus minus month>, <plus minus year>)
= specified date (format dd.mm.yyyy) as above.

\$ARG (<n>) = argument passed (Arg1,2,...):

Asci4rfc <script file name> [Arg2 [Arg3] ...].

Arg2 is **\$ARG (2)** , Arg3 is **\$ARG (3)** , ...

Example

```
!REM ZSCANHEUTE
!REM GATHMANN@CSBG.DE
!REM 19.11.2003
!REM Debug 1 to 5
!REM Mind syntax: Keyword in capital letters only,
!REM trailing comma

!DEBUG=5

!REM set debug level by command line argument No ,
!REM Example call: asci4rfc thisscript.dat 1 → Debug level = 1
!DEBUG=$ARG(2)

!FILE=zscan1.TXT,w

!LOGIN:TYPE=3 CLIENT=010 USER=USER1 PASSWD=123456 LANG=D ASHOST=SAPHOST
SYSNR=00

!REM Test Run
!REFRESH
!FM:ZPP_FU_SCANNER_001
!EXPORT:
ANWENDUNG=01,
SCANIN=12345,
!IMPORT:
ZEILE1=80,
ZEILE2=80,
ZEILE3=80,
ZEILE4=80,
SCANCMD=20,
!RUN
!OUTPUT:IMPORT=ZEILE1,ZEILE2,ZEILE3,ZEILE4,SCANCMD,

!FILE=zscan2.TXT,w
!REM Next Scanner Data
!REFRESH
!FM:RFC_READ_TABLE
!EXPORT:
QUERY_TABLE=ZSCAN,
DELIMITER=;,
!TABLE,103:FIELDS LDATE,LZEIT,TEXT,
!REM TABLE,72:OPTIONS LDATE = '20031119',
!TABLE,72:OPTIONS LDATE > '$DATE(-5,+0,-1)',
!TABLE,512:DATA,
!RUN
!OUTPUT:TABLE=DATA,
```