

Important ABAP Functions:

Function modules are procedures that are defined in function groups (special ABAP programs with type F) and can be called from any ABAP program. Function groups act as containers for function modules that logically belong together. You create function groups and function modules in the ABAP Workbench using the Function Builder.

Function modules allow you to encapsulate and reuse global functions in the R/3 System. They are stored in a central library. The R/3 System contains a wide range of predefined function modules that you can call from any ABAP program. Function modules also play an important role in database updates and in remote communications between R/3 Systems or between an R/3 System and a non-SAP system. Unlike subroutines, you do not define function modules in the source code of your program.

Find here some usefull ABAP Function Modules!

I. Date-Time Functions:

- | | |
|------------------------------------|--|
| 1. DATE_GET_WEEK | - Returns week for a date |
| 2. WEEK_GET_FIRST_DAY | - Returns first day for a week |
| 3. RP_LAST_DAY_OF_MONTHS | - Returns last day of month |
| 4. FIRST_DAY_IN_PERIOD_GET | - Get first day of a period |
| 5. LAST_DAY_IN_PERIOD_GET | - Get last day of a period |
| 6. RP_LAST_DAY_OF_MONTHS | - Determine last day of month |
| 7. DATE_COMPUTE_DAY | - Returns a number indicating what day of the week the date falls on.
Monday is returned as a 1, Tuesday as 2, etc. |
| 8. DATE_IN_FUTURE | - Calculate a date N days in the future. |
| 9. RP_CALC_DATE_IN_INTERVAL | - Add days/months to a date |
| 10. RP_CALC_DATE_IN_INTERVAL | - Add/subtract years/months/days from a date |
| 11. SD_DATETIME_DIFFERENCE | - Give the difference in Days and Time for 2 dates |
| 12. SCOV_TIME_DIFF_46B | - Gets the time difference. |
| 13. DATE_CREATE | - Calculates a date |
| 14. MONTH_PLUS_DETERMINE | - Add or subtract months from a date. To subtract a month, enter a negative value for the 'months' parameter. |
| 15. HOLIDAY_GET | - Provides a table of all the holidays based upon a Factory Calendar &/ Holiday Calendar. |
| 16. HOLIDAY_CHECK_AND_GET_INFO | - Useful for determining whether or not a date is a holiday. Give the function a date, and a holiday calendar, and you can determine if the date is a holiday by checking the parameter HOLIDAY_FOUND. |
| 17. DATE_CHECK_PLAUSIBILITY | - Check to see if a date is in a valid format for SAP. Works well when validating dates being passed in from other systems. |
| 18. DATE_CONV_EXT_TO_INT | - Conversion of dates to SAP internal format e.g. '05.06.2000' -> 20000605.
Can also be used to check if a date is valid (sy-subrc <> 0) |
| 19. MONTH_NAMES_GET | - It returns all the month and names in respective language. |
| 20. GET_WEEK_INFO_BASED_ON_DATE | - Name says it all. |
| 21. CCM_GO_BACK_MONTHS | - FM to Go back specified number of months.
Eg. 09/15/2005 - 1 month = 08/15/2005. |
| 22. RE_ADD_MONTH_TO_DATE | - Can add/subtract months to/from date. |
| 23. FIMA_DAYS_AND_MONTHS_AND_YEARS | - Get number of days. Year, Months between two dates. |
| 24. HR_HK_DIFF_BT_2_DATES | - Calculate the days, months and years between 2 dates. |

File functions – WorkStation

- | | |
|-------------------|---|
| 1. WS_UPLOAD | - File transfer from presentation server file to internal table |
| 2. WS_DOWNLOAD | - File Transfer from Internal Table to Presentation Server File |
| 3. WS_DOWNLOAD | - Save Internal Table as File on the Presentation Server |
| 4. WS_FILE_DELETE | - Delete File at the Front-end |

- | | |
|-----------------------------------|---|
| 5. WS_FILENAME_GET | - Call File Selector |
| 6. WS_EXECUTE | - Execute an External Program on the Presentation Server. |
| 7. WS_MSG | - Execute an External Program on the Presentation Server. |
| 8. WS_EXCEL | - Download internal table into file and start Excel for it. |
| 9. WS_QUERY | - Execute query function on the Presentation Server |
| 10. FILENAME_GET | - Popup to get a filename from a user, returns blank filename if user selects cancel |
| 11. LIST_DOWNLOAD | - Download of ABAP list (Report) to local file |
| 12. LIST_TO_ASCII | - Converts the specified list (LIST_INDEX) or the provided list (LISTOBJECT) to ASCII. |
| 13. RZL_READ_FILE | - Read a file from the presentation server if no server name is given, or read file from remote server. Very useful to avoid authority checks that occur doing an OPEN DATASET. This function using a SAP C program to read the data. |
| 14. RZL_WRITE_FILE_LOCAL | - Saves table to the presentation server (not PC). Does not use OPEN DATASET, so it does not suffer from authority checks! |
| 15. F4_FILENAME | - F4 for filename / Filemanager support to locate file in a directory. Mail and Office |
| 16. RS_SEND_MAIL_FOR_SPOOLIST | - Send message from ABAP/4 program to SAPoffice. |
| 17. SO_NEW_DOCUMENT_SEND_API1 | - Sending express mail (SAP office) |
| 18. RS_SEND_MAIL_FOR_SPOOLIST | - Send message from ABAP/4 program to SAPoffice. |
| 19. SO_NEW_DOCUMENT_ATT_SEND_API1 | - SAPoffice: Send new document with attachments via RFC. |

Text functions

- | | |
|---------------------------------|---|
| 1. SAVE_TEXT | - To load long text into SAP |
| 2. READ_TEXT | - To load long text into SAP Important one! |
| 3. COMMIT_TEXT | - To load long text into SAP |
| 4. INIT_TEXT | - To load long text into SAP |
| 5. CONVERSION_EXIT_ALPHA_INPUT | - Place leading Zeros (Important one) |
| 6. CONVERSION_EXIT_ALPHA_OUTPUT | - Delete leading zeros (Important one) |
| 7. TEXT_SPLIT | - Long Text formatting function |
| 8. MATERIAL_BTCI_TEXT | - Long text for material |
| 9. SCP_REPLACE_STRANGE_CHARS | - It processes a text so that it only contains simple characters. Special characters and national characters are replaced in such a way that the text remains reasonably legible. |

Currency Functions

- | | |
|----------------------------------|---|
| 1. CONVERT_TO_FOREIGN_CURRENCY | - Convert local currency to foreign currency. |
| 2. CONVERT_TO_LOCAL_CURRENCY | - Convert from foreign currency to local currency |
| 3. CURRENCY_AMOUNT_SAP_TO_IDOC | - Convert currency to IDOC format |
| 4. HRCM_AMOUNT_TO_STRING_CONVERT | - Convert amount to string |
| 5. HRCM_STRING_TO_AMOUNT_CONVERT | - Convert amount from string |
| 6. CLOI_PUT_SIGN_IN_FRONT | - Move the negative sign from the left hand side of a number, to the right hand side of the number. Note that the result will be left justified (like all character fields), not right justified as numbers normally are. |
| 7. SPELL_AMOUNT | - Convert numbers and figures in words |
| 8. HR_IN_CHG_INR_WRDS | - Convert numbers and figures in words |

Address – Functions

- | | |
|---------------------------|--|
| 1. ADDR_GET | - Get address. |
| 2. ADDRESS_INTO_PRINTFORM | - Address format according to Post Office guidelines |
| 3. ADDRESS_SHOW_PRINTFORM | - Address format according to Post Office guidelines |

XML

1. TEXT_CONVERT_XML_TO_SAP

Units of measure (UOM)

1. UNIT_OF_MEASURE_ISO_TO_SAP - Converting ISO Unit to base unit of measure
2. MATERIAL_UNIT_CONVERSION - Material quantity conversion from Base Unit of Measure to Alternative Unit of Measure and vice versa.

User – Function:

1. TERMINAL_ID_GET - Returns the terminal id
2. TH_DELETE_USER - Logoff a user like we do in SM04.
3. TH_ENVIRONMENT - Get the UNIX environment
4. TH_POPUP - Display a popup system message on a specific users screen.
5. TH_REMOTE_TRANSACTION - Run a transaction on a remote server. Optionally provide BDC data to be used in the transaction
6. TH_USER_INFO - Give information about the current user (sessions, workstation logged in from, etc)
7. TH_USER_LIST - Show which users are logged into an app server
8. TMP_GUI_DIRECTORY_LIST_FILES - Retrieve all of the files and subdirectories on the Presentation Server (PC) for a given directory.

Batch Data Communication (BDC)

1. BDC_OPEN_GROUP
2. BDC_CLOSE_GROUP
3. BDC_INSERT

Pop-up screens

1. WS_MSG - Create a dialog box in which you display an one-line message.
2. POPUP_TO_CONFIRM_STEP - Create a dialog box in which you make a question whether the user wishes to perform the step.
3. POPUP_TO_CONFIRM_WITH_MESSAGE - Create a dialog box in which you inform the user about a

- specific decision point during an action.
3. POPUP_TO_CONFIRM_WITH_VALUE - Create a dialog box in which you make a question whether the user wishes to perform a processing step with a particular object.
 4. POPUP_TO_DECIDE - Provide user with several choices as radio buttons
 5. POPUP_TO_DECIDE_WITH_MESSAGE - Create a dialog box in which you inform the user about a specific decision point via a diagnosis text.
 6. POPUP_TO_DISPLAY_TEXT - Create a dialog box in which you display a two-line message.
 7. POPUP_TO_SELECT_MONTH - Popup to choose a month
 8. POPUP_WITH_TABLE_DISPLAY - Provide a display of a table for user to select one, with the value of the table line returned when selected.
 9. POPUP_TO_CONFIRM - Pop-up dialog confirm an action before it is carried out.
 10. POPUP_TO_DISPLAY_TEXT - Displays a text in a modal screen
 11. POPUP_TO_INFORM - Displays several lines of text. No OK or Cancel buttons.
 12. POPUP_TO_CONFIRM/POPUP_TO_DISPLAY_TEXT
 13. TH_POPUP - Display a popup system message on a specific users screen (Good one).
 14. POPUP_TO_CONFIRM_LOSS_OF_DATA - Create a dialog box in which you make a question whether the user wishes to perform a processing step with loss of data.
 15. POPUP_TO_CONFIRM_STEP - Create a dialog box in which you make a question whether the user wishes to perform the step.
 16. POPUP_TO_CONFIRM_WITH_MESSAGE - Create a dialog box in which you inform the user about a specific decision point during an action.
 17. POPUP_TO_CONFIRM_WITH_VALUE - Create a dialog box in which you make a question whether the user wishes to perform a processing step with a particular object.
 18. POPUP_TO_DECIDE - Provide user with several choices as radio buttons
 19. POPUP_TO_DECIDE_WITH_MESSAGE - Create a dialog box in which you inform the user about a specific decision point via a diagnosis text.
 20. POPUP_TO_DISPLAY_TEXT - Create a dialog box in which you display a two-line message.
 21. POPUP_WITH_TABLE_DISPLAY - Provide a display of a table for user to select one, with the value of the table line returned when selected.

GUI – Functions:

1. GUI_DOWNLOAD - Down load functionality data communication of an internal table of the server into a file on the PC. The component Gui_Download replaces the obsoleten components Ws_Download and Download. The file dialogue of the Download of component is available in the class Cl_Gui_Frontend_Services.
GUI_DOWNLOAD - Class of CL_GUI_FRONTEND_SERVICES, this function is to be used platform spefic and therefore only with appropriate care.
2. GUI_HAS_ACTIVEX - Test Whether ActiveX Controls are Supported
3. GUI_HAS_JAVABEANS - Shortcut to GUI_HAS_OBJECTS(SFES_OBJ_JAVABEANS)
4. GUI_UPLOAD - The component loads a file from the PC on the server. The data can be transmitted binary or as text. Numbers and date fields can be interpreted in accordance with the user attitudes.
5. GUI_VSS_DOWNLOAD - Download on the local PC in the VSS format down load of internal tables on the local PC with variable line length. As line end in each case the control characters 0D, 0A become write-writing those delivery table DATA_TAB must from two fields bestehen:Feld 1 (type I): the desired length (number of indication) contains that expenditure line field 2 (type C): datawith indication code PAGE (parameter CODE PAGE) made of table contains of a TCP00 the output file in this code PAGE is provided. Otherwise the code PAGE of the Frontends is used.
6. GUI_VSS_UPLOAD - Upload of the local PC in the VSS FormatUpload of files of local PC with variable line length into an internal table DATA_TAB. The lines of the file must by the control characters 0D, 0a separately be-being those delivery table DATA_TAB must from two fields bestehen:Feld 1 (type I): the length (number of indications) contains the read in line field 2 (type C): the read in line contains file with indication code PAGE (parameter CODE PAGE) from table of the TCP00 the read in file in this code PAGE is interpreted. It takes place internally a conversion in the system code PAGE.
7. GUI_ADD_TCODE_TO_FAVORITES

Others:

1. OPEN_FORM Ever used one!
2. CLOSE_FORM Ever used one!
3. WRITE_FORM Ever used one!
4. CONTROL_FORM Ever used one!
5. CONVERT_OTF_2_PDF - Convert OTF format to various formats (TLINE table)
6. CONVERT_OTF_2_PDF - Convert OTF to PDF (TLINE table). OTF can be filled used archive link.
7. SAPGUI_PROGRESS_INDICATOR - Set 'Progress Indicator' in Current Window (Good one).
8. CONVERSION_EXIT_ALPHA_INPUT - Conversion exit (Famous one).
9. CONVERSION_EXIT_ALPHA_OUTPUT - Conversion exit (Famous one).
10. QF05_RANDOM - Remember Visual Basic guys! Random Function is here.
11. QF05_RANDOM_INTEGER -

ALE EDI IDocs

1. EDI_DOCUMENT_OPEN_FOR_CREATE
2. EDI_DOCUMENT_OPEN_FOR_FORCE
3. EDI_DOCUMENT_OPEN_FOR_TRACE
4. EDI_DOCUMENT_OPEN_FOR_EDIT
5. EDI_DOCUMENT_OPEN_FOR_PROCESS
6. EDI_DOCUMENT_OPEN_FOR_READ
7. EDI_SEGMENT_GET_NEXT
8. EDI_SEGMENT_GET
9. EDI_SEGMENTS_GET_ALL
10. EDI_SEGMENTS_GET_RANGE EDI
11. EDI_DOCUMENT_READ_ALL_STATUS
12. EDI_DOCUMENT_READ_LAST_STATUS
13. EDI_DOCUMENT_STATUS_SET
14. EDI_STATUS_ADD_BLOCK
15. EDI_SEGMENT_ADD_NEXT
16. EDI_SEGMENTS_ADD_BLOCK
17. EDI_DOCUMENT_CLOSE_CREATE
18. EDI_DOCUMENT_CLOSE_CREATE_TAB
19. EDI_DOCUMENT_CLOSE_FORCE
20. EDI_DOCUMENT_CLOSE_TRACE
21. EDI_DOCUMENT_CLOSE_EDIT
22. EDI_DOCUMENT_CLOSE_PROCESS EDI
23. EDI_DOCUMENT_CLOSE_PROCESS_UPD
24. EDI_DOCUMENT_CLOSE_READ
25. EDI_DOCUMENT_STATUS_DISPLAY

Functional Part

Sales & Distribution (SD)

1. SD_SALES_DOCUMENT_READ - Reads sales document header and business data: tables VBAK, VBKD and VBPA (Sold-to (AG), Payer (RG) and Ship-to (WE) parties).
2. RV_ORDER_FLOW_INFORMATION - Reads sales document flow of sales document after delivery and billing.
3. SD_SALES_DOCUMENT_READ_POS
4. SD_DOCUMENT_PARTNER_READ
5. SD_DETERMINE_CONTRACT_TYPE
6. SD_SALES_DOCUMENT_COPY
7. SD_SALES_DOCUMENT_SAVE
8. SD_PACKING_PRINT_VIEW
9. SD_DELIVERY_VIEW
10. RV_BILLING_PRINT_VIEW

Material Management (MM)

1. CONVERSION_EXIT_MATN1_OUTPUT
2. CONVERSION_EXIT_MATN1_INPUT
3. ME_READ_HISTORY
4. ME_DISPLAY_PURCHASE_DOCUMENT

- Jimstex for [SAP Database](#)