

ABAP Cheatsheet

This cheatsheet covers statements, system fields, program types and selectors, I am not going to cover the whole syntax for each but give you enough so that you can lookup the full syntax either using the SAP portal or the web.

- Statements
- System Fields
- ABAP Program Types
- Selectors

Statements

Introducing a Program	
CLASS-POOL	Introduction of a class pool
FUNCTION-POOL	Introduction of a function group
INTERFACE-POOL	Introduction of a interface pool
PROGRAM	Introduction of a module pool or a subroutine
REPORT	Introduction of a executable program
TYPE-POOL	Introduction of a type pool
Modularization (procedures)	
FORM ENDFORM	Definition of a subroutine
FUNCTION ENDFUNCTION	Definition of a function module
METHOD ENDMETHOD	Definition of a method
Modularization (dialog modules)	
MODULE ENDMODULE	Definition of a dialog module
Modularization (event blocks)	
AT LINE-SELECTION	List event
AT PF##	List event
AT SELECTION-SCREEN	Selection screen event
AT USER-COMMAND	List event
END-OF-PAGE	List event
END-OF-SELECTION	Reporting event
GET	Reporting event
INITIALIZATION	Reporting event
LOAD-OF-PROGRAM	Program constructor event
START-OF-SELECTION	Reporting event
TOP-OF-PAGE	List event
Declarative Statements (data types and data objects)	
CONSTANTS	Declaration of a constant
DATA	Declaration of a variable
FIELD-SYMBOLS	Declaration of a field symbol
INCLUDE	integration of a structure
NODES	Declaration of a table work area
STATICS	Declaration of a static variable
TABLES	Declaration of a table work area
TYPE-POOLS	integration of a type pool
TYPES	Declaration of an independent data type
Declarative Statements (classes and interfaces)	
ALIASES	Declaration of an alias name
CLASS ENDCLASS	Declaration of a class
CLASS-DATA	Declaration of a static attribute
CLASS-EVENTS	Declaration of a static event
CLASS-METHODS	Declaration of a static method
EVENTS	Declaration of an instance event
INTERFACE ENDINTERFACE	Declaration of an interface
INTERFACES	integration of an interface
METHODS	Declaration of an instance method
PRIVATE SECTION	Introduction of the private visibility area
PROTECTED SECTION	Introduction of the protected visibility area
PUBLIC SECTION	Introduction of the public visibility area

Declarative Statements (object creation)	
CREATE DATA	Creation of an anonymous data object
CREATE OBJECT	Creation of an object
Calling and Exiting Program Unit (calling programs)	
CALL TRANSACTION	Calls a transaction
LEAVE TO TRANSACTION	Calls a transaction
SUBMIT	Calls an executable program
Calling and Exiting Program Unit (calling processing blocks)	
CALL FUNCTION	Calls a function module
CALL METHOD	Calls a method
PERFORM	Calls a subroutine
PUT	Triggers a reporting event
RAISE EVENT	Triggers an event
SET HANDLER	Registers an event
SET USER-COMMAND	Triggers a list event
Calling and Exiting Program Unit (exiting program units)	
CHECK	Exits a loop pass
CONTINUE	Exits a loop pass
EXIT	Exits a loop
LEAVE PROGRAM	Exits a ABAP program
REJECT	Exits a processing block
RETURN	Exits a processing block
STOP	Exits a processing block
Program Flow Control (control structure)	
DO ENDDO	Loop
CASE WHEN ENDCASE	Branch
IF ELSEIF ELSE ENDIF	Branch
WHILE ENDWHILE	Loop
Program Flow Control (program interruption)	
WAIT UP TO	Program interruption for a specific period of time
Program Flow Control (exception handling)	
CATCH SYSTEM-EXCEPTIONS	Catches catchable runtime errors
RAISE	Triggers a non-class-based exception
RAISE EXCEPTION	Triggers a class-based exception
TRY CATCH CLEANUP ENDTRY	Handles class-based exception
Assignments (value assignments)	
MOVE	Assignment based on a conversion rule
MOVE-CORRESPONDING	Assignment of structure components
UNPACK	Unpacks a packed number
WRITE TO	Formatted assignment
Assignments (setting references)	
ASSIGN	Sets a field symbol
UNASSIGN	Initializes a field symbol
GET REFERENCE	Sets a data reference
Assignments (initializations)	
CLEAR	Initializes a data object
FREE	Initializes a data object
REFRESH	Initializes an internal table
Processing Internal Data (calculation expressions)	
COMPUTE	Calculates an arithmetic or bit expression
Processing Internal Data (calculation statements)	
ADD	Adds numerical data objects
DIVIDE	Divides numerical data objects
MULTIPLY	Multiplies numerical data objects
SUBTRACT	Subtracts numerical data objects
Processing Internal Data (processing byte and character strings)	
CONCATENATE	Concatenate byte or character strings
CONDENSE	Condenses a character string

CONVERT TEXT	Converts a character string
FIND	Searches within a byte or character string
GET BIT	Reads single bits within a byte string
OVERLAY	Carries out replacements within a character string
REPLACE	carries out replacements within a byte or a character string
SET BIT	Sets single bits within a byte string
SHIFT	Shifts a byte or a character string
SPLIT	Splits a byte or a character string
TRANSLATE	Translates a character string
Processing Internal Data (internal tables)	
APPEND	Appends single lines to an internal table
AT	Control level processing of an internal table
COLLECT	Insertion of collected lines into an internal table
DELETE	Delete lines from an internal table
FIND IN TABLE	Performs search run in an internal table
INSERT	Inserts lines into an internal table
LOOP AT ENDLOOP	Loop at an internal table
MODIFY	Modifies lines in an internal table
PROVIDE ENDPROVIDE	Loop at several internal tables
READ TABLE	Reads a row of an internal table
REPLACE IN TABLE	Carries out a replacement in an internal table
SORT	Sorts an internal table
SUM	Sums up numerical fields in an internal table
Processing Internal Data (Extracts)	
AT	Control level processing of the extract dataset
EXTRACT	Fills the extract dataset
FIELD GROUPS	Declaration of a field group
INSERT	Builds up a field group
LOOP ENDLOOP	Loop at the extract dataset
SORT	Sorts the extract dataset
Processing Internal Data (Attributes of data objects)	
DESCRIBE	Determines the attributes of a data object
User Dialogs (General Dynpros)	
CALL SCREEN	Calls a dynpro sequence
CONTROLS	Declaration of a control
EXIT FROM STEP-LOOP	Exists a table control or step loop
SET CURSOR	Reads the cursor position
GET PF-STATUS	Determines the GUI status
LEAVE [TO] SCREEN	Exists a dynpro
LOOP AT SCREEN ENDLOOP	Loop at screen elements
MODIFY SCREEN	Modifies a screen element
REFRESH CONTROL	Initializes a table control
SET CURSOR	Sets the cursor position
SET HOLD DATA	Switches standard menu items on and off
SET PF-STATUS	Sets the GUI status
SET SCREEN	Sets the next dynpro
SET TITLEBAR	Sets the GUI title
SUPPRESS DIALOG	Suppresses the screen layout
User Dialogs (Selection Screens)	
PARAMETERS	Defines a parameter
SELECTION-SCREEN	Defines a selection screen or a screen element
SELECT-OPTIONS	Defines a selection criterion
User Dialogs (Classic Lists)	
BACK	Relative positioning of list cursor
DESCRIBE LIST	Determine the properties of a list in the list buffer
FORMAT	Formats a list
GET CURSOR	Reads the cursor position
HIDE	Stores a data object in the list level
LEAVE TO LIST-PROCESSING	Calls the list processing
LEAVE LIST-PROCESSING	Exist the list processing
MODIFY LINE	Modifies a list in the list buffer

NEW-LINE	Line break in a list
NEW-PAGE	Page break in a list
POSITION	Positions the list cursor
PRINT-CONTROL	Formats a print list
READ LINE	Reads a list in the list buffer
RESERVE	Relative page break in a list
SCROLL LIST	Scrolls a list
SET BLANK LINES	Controls blank characters in a list
SET CURSOR	Sets the cursor position
SET MARGIN	Sets the margin of a print list
SET PF-STATUS	Sets the GUI status
SET LEFT SCROLL-BOUNDARY	Sets the scrollable area of a list
SET TITLEBAR	Sets the GUI title
SKIP	Positions the list cursor
ULINE	Outputs a horizontal line in a list
WINDOW	Outputs a list in a dialog window
WRITE	Outputs data to a list
User Dialogs (Messages)	
MESSAGE	Sends a message
Processing External Data (open SQL)	
CLOSE CURSOR	Closes a database cursor
DELETE	Deletes rows from a database table
FETCH NEXT CURSOR	Reads rows through a database table
INSERT	Inserts rows into a database table
MODIFY	Modifies or inserts rows in a database table
OPEN CURSOR	Opens a database cursor
SELECT ENDSELECT	Reads rows from a database table
UPDATE	Modifies rows in a database table
Processing External Data (Native SQL)	
EXEC SQL ENDEXEC	Defines a coding area from Native SQL
EXIT FROM SQL	Exists the Native SQL processing
Processing External Data (Data Clusters)	
DELETE	Deletes data clusters
EXPORT	Exports data objects into a data cluster
FREE MEMORY	Delete data clusters from the ABAP memory
IMPORT	Imports data objects from a data cluster
IMPORT DIRECTORY	Creates the table of contents of a data cluster
Processing External Data (File Interface)	
CLOSE DATASET	Closes a file
DELETE DATASET	Deletes a file
GET DATASET	Determines the properties of a file
OPEN DATASET	Opens a file
READ DATASET	Reads a file
SET DATASET	Sets the properties of a file
TRANSFER	Fills a file
TRUNCATE DATASET	Changes the size of a file
Processing External Data (Data Consistency)	
AUTHORITY-CHECK	Checks an authorization
COMMIT WORK	Completes an SAP LUW
ROLLBACK WORK	Cancels an SAP LUW
SET UPDATE TASK LOCAL	Sets a local update task
Program Parameters (SAP Memory)	
GET PARAMETER	Reads an SPA/GPA parameter
SET PARAMETER	Sets an SPA/GPA parameter
Program Parameters (Language Environment)	
GET LOCALE	Determines the text environment
SET COUNTRY	Sets output formats in lists
SET LANGUAGE	Loads a text pool
SET LOCALE	Sets the text environment
Program Parameters (Date and Time information)	
CONVERT DATE	Converts data and time into a time stamp
CONVERT TIME STAMP	Converts a time stamp into a date and time
GET TIME	Fills the system fields for data and time

GET TIME STAMP	Creates a time stamp
Program Processing (Testing and Checking Programs)	
ASSERT	Defines an assertion
SET EXTENDED CHECK	Controls the extended program check
BREAK-POINT	Defines a break point
LOG-POINT	Defines a log point
GET RUN TIME	Determines a measurement interval
SET RUN TIME ANALYZER	Controls the runtime analysis
SET RUN TIME CLOCK	Sets the measurement accuracy for measurement intervals
Program Processing (Dynamic Program Development)	
EDITOR-CALL FOR REPORT	Calls the ABAP Editor
GENERATE SUBROUTINE POOL	Generates a subroutine pool
INSERT REPORT	Stores an ABAP program
INSERT TEXTPOOL	Stores a text pool
READ REPORT	Reads an ABAP Program
READ TEXTPOOL	Reads a test pool
SYNTAX-CHECK	Calls the syntax check
ABAP Data and Communication Interfaces (Remote Function Call)	
CALL FUNCTION DESTINATION	Remote function call
RECEIVE	Receives parameters for a remote function call
WAIT UNTIL	waits until a remote function call has finished
ABAP Data and Communication Interfaces (ABAP and XML)	
CALL TRANSFORMATION	Calls an XSLT program or a simple transformation
ABAP Data and Communication Interfaces (OLE Interface)	
CALL METHOD	Calls an OLE automation method
CREATE OBJECT	Creates an OLE automation object
FREE OBJECT	Releases memory in the OLE automation
GET PROPERTY	Reads an attribute in the OLE automation
SET PROPERTY	Sets an attribute in the OLE Automation
Enhancements (Source code enhancements)	
ENHANCEMENTS ENDENHANCEMENTS	Implements an enhancement to the source code
ENHANCEMENT-POINT	Defines an enhancement to the source code
ENHANCEMENT-SECTION END-ENHANCEMENT-SECTION	Defines an enhancement to the source code
Enhancements (Enhancements via BAdIs)	
GET BADI	Creates a BAdI object
CALL BADI	Calls BAdI methods

System Fields

Information on the current AS ABAP	
sy-dbsys	Central database system of AS ABAP example Oracle,etc
sy-host	Name of the computer in the network
sy-opsys	Operating system of the application server
sy-saprl	Release status of AS ABAP for example 700, 710
sys-sysid	Name of AS ABAP such as NSP, PF1
Information on the current user session	
sy-langu	One digit language key for example D, E, F
sy-mandt	Client ID used by the user to log on for example 000, 200
sy-modno	Indexing of external sessions, set to 0 in the first session. Increased by 1 in new sessions created with <i>create session</i> function or by calling a transaction with /o in the input field in the standard toolbar
sy-uname	Users logon name
Date and Time Information	
sy-datlo	Local date of the user
sy-datum	date of the application server
sy-dayst	Set to 'X' during daylight saving time, otherwise leave blank
sy-fdayw	Factory calendar weekday, Monday=1, Friday=5, etc
sy-timlo	Local time of the user
sy-tzone	Time difference to UTC reference time in terms of seconds
sy-zeit	Time of the application server
sy-zonlo	Users time zone
Information on the current ABAP Program	
sy-calld	Contains a blank character in the first program of a call sequence otherwise set to 'X'
sy-cprog	In externally called procedure the name of the calling program otherwise the name of the current program.

sy-dbnam	The linked logical database for executable programs
sy-dyngr	Screen group of the current dynpro.
sy-dynnr	The current selection screen during selection screen processing, the number of the subscreen container during list processing. During the processing of a subscreen dynpro the number of that subscreen dynpro
sy-ldbpg	The database program of the linked logical database for executable programs
sy-repid	Name of the current ABAP program
sy-tcode	Name of the current transaction code
Information on Batch and Batch input processing	
sy-batch	Set to 'X' in ABAP programs running in the background, otherwise initial
sy-binpt	Set to 'X' during processing of batch input session and in ABAP programs called via <i>call transaction</i> using otherwise initial
ABAP Programming - Constants	
sy-abcde	Contains the alphabet
sy-uline	Contains a horizontal line of 255 characters length for list outputs
sy-vline	Contains a vertical line 'I' for list outputs
ABAP Programming - Loop Processing	
sy-index	Contains the number of loop passes, including the current loop in <i>do</i> and <i>while</i> loops
ABAP Programming - Character String Processing	
sy-fdpos	Position information in operations with character type fields
ABAP Programming - Internal Tables	
sy-tabix	Line of an internal (index) table that has been addressed last, is set to 0 when accessing a hashed table.
sy-tfill	During statements <i>describe table</i> , <i>loop at</i> and <i>read table</i> , it is filled with the number of lines in the addressed internal table.
sy-dbcnt	During statements <i>describe table</i> , <i>loop at</i> and <i>read table</i> , it is filled with the line width of the addressed internal table.
ABAP Programming - Database Accesses	
sy-dbcnt	SQL statements set the content of <i>sy-dbcnt</i> to the number of processed table rows
ABAP Programming - Return Value	
sy-subrc	Return value that is set by many ABAP statements, in general 0 means that the statement executed without problems
ABAP Programming - Dynpros	
sy-cucol	Horizontal cursor position, the count begins at column 2
sy-curow	Vertical cursor position, the count begins at row 1
sys-datar	Set to 'X' during PAI if at least one input field of a dynpro was changed due to a user action or other data transfer, otherwise blank
sy-loopc	Number of lines currently displayed in a table control
sy-pfkey	GUI status of the current dynpro
sy-scols	Number of columns in the current screen
sy-srows	Number of rows in the current screen
sy-step1	Index of the current row in a table control, is set during each loop pass
sy-title	Text that is displayed in the title bar of the dynpro
sy-ucomm	Function code that is triggered during the PAI event
ABAP Programming - Selection Screens	
sy-slset	Variant used to fill a selection screen
ABAP Programming - List Creation	
sy-colno	Current column during list creation, the count begins at 1
sy-linct	Page length of the list
sy-linno	Current line during list creation
sy-linsz	Line width of the list
sy-pagno	Current page during list creation
sy-tvar0 ... st-tvar9	You can assign values to these system fields in your programs, during the TOP-OF-PAGE event the contents of the variables replace the placeholders in the list and column headers of the program
sy-wtitl	Is set to 'N' in the <i>report</i> , <i>program</i> and <i>function-pool</i> statements if the addition <i>no standard page heading</i> was used otherwise blank
ABAP Programming - Interactive List Processing	
sy-cpage	Page number of the top page from the list in which the list event was triggered, the count begins at 1
sy-lilli	Line in which the list event was triggered, the count begins at 1 and contains the page header
sy-lisel	Content of the line in which the list event was triggered
sy-listi	Index of the list in which the list event was trigger
sy-lsind	Index of the list that is being created
sy-staco	Number of the first displayed column of the list in which the list event was triggered, the count begins at 1
sy-staro	Number of the top displayed row in the top page of the list in which the list event was triggered, the count begins at 1 and the page header is not included in the count.
ABAP Programming - Printing Lists	
sy-callr	Contains a value indicating where the printout was started, such as <i>new-page</i> for program driven printing or <i>rsdbrunt</i> for printing from the selection screen.
sy-prdsn	Contains the name of the spool file
sy-spono	Contains the number of the spool file
sy-marow	Contains the number of rows in the top margin

sy-macol	Contains the number of columns in the left margin
ABAP Programming - Messages	
sy-msgid	Contains the message class after the <i>message</i> statement
sy-msgno	Contains the message number after the <i>message</i> statement
sy-msgty	Contains the message type after the <i>message</i> statement
sy-msgv1 ... sy-msgv4	Contains the field contents that were used for the placeholder of the message after the <i>message</i> statement.

ABAP Program Types

The following table provides an overview of all ABAP program types as well as program components and properties supported by those types.

	Executable program	Module Pool	Function Group	Subroutine Pool	Type Group	Class Pool	Interface Pool
Global data types and data objects	X	X	X	X	X	X	
Dynpros	X	X	X				
Dialog Modules	X	X	X				
Program constructor event (load-of-program)	X	X	X	X			
Selection screen events	X	X	X				
Classic list events	X	X	X				
Reporting events	X						
Classes	X	X	X	X		X	
Interfaces	X	X	X	X		X	X
Methods	X	X	X	X		X	
Function Modules			X				
Subroutines	X	X	X	X			
Executable via <i>submit</i>	X						
Executable via transaction code	X	X	X	X		X	

Selectors

The following table provides an overview of all selectors that can be used in names

-	Structure component selector for <i>struct-comp</i>
->	Object component selector for <i>ref->comp</i>
=>	Class component selector for <i>class=>comp</i>
-	Interface component selector for <i>intf-comp</i>
-	Column selector for <i>dbtab-col</i>