

Lection 4 Workbook

Development start

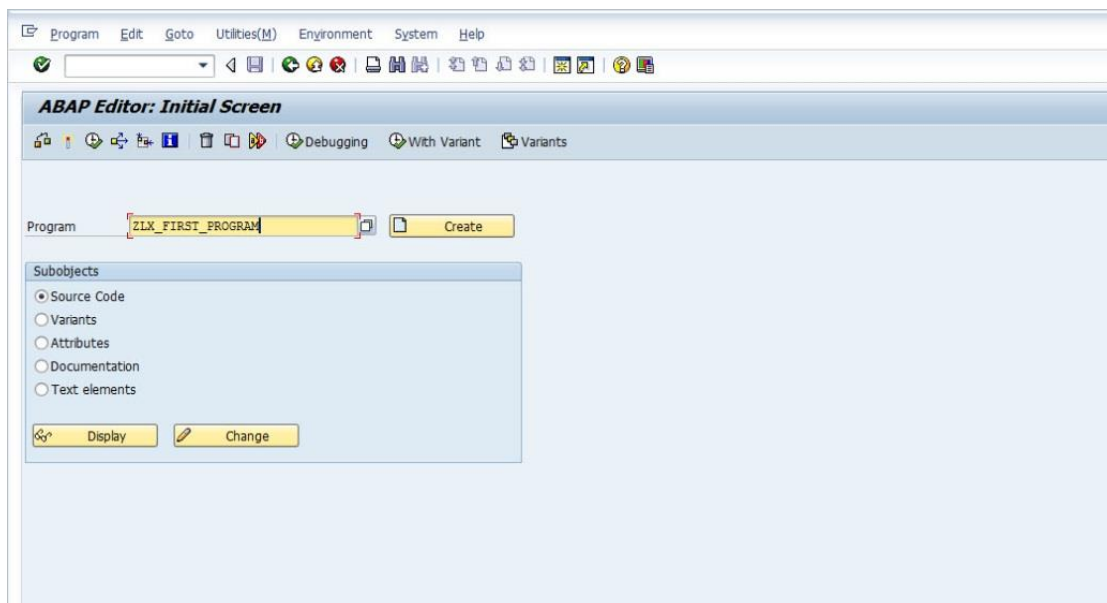
Contents

HOW TO CREATE AND EXECUTE SIMPLE REPORT "HELLO WORLD"	3
APPENDIX 1 - IMPORTANT SY-* FIELDS	6
<i>System fields for screens</i>	<i>6</i>
<i>Selection screens</i>	<i>6</i>
<i>System fields.....</i>	<i>6</i>
<i>System fields for date and time.....</i>	<i>7</i>
<i>System fields in lists.....</i>	<i>7</i>
APPENDIX 2 - ABAP TYPES.....	8
<i>Predefined Elementary ABAP Types with Fixed Length.....</i>	<i>8</i>
<i>Predefined Elementary ABAP Types with Variable Length.....</i>	<i>8</i>
APPENDIX 3 - DATA ELEMENTS ELEMENTARY TYPES	9
APPENDIX 4 - ABAP STATEMENTS OVERVIEW	10
APPENDIX 5 - NAMING CONVENTIONS IN ABAP CODE (EXAMPLE).....	15

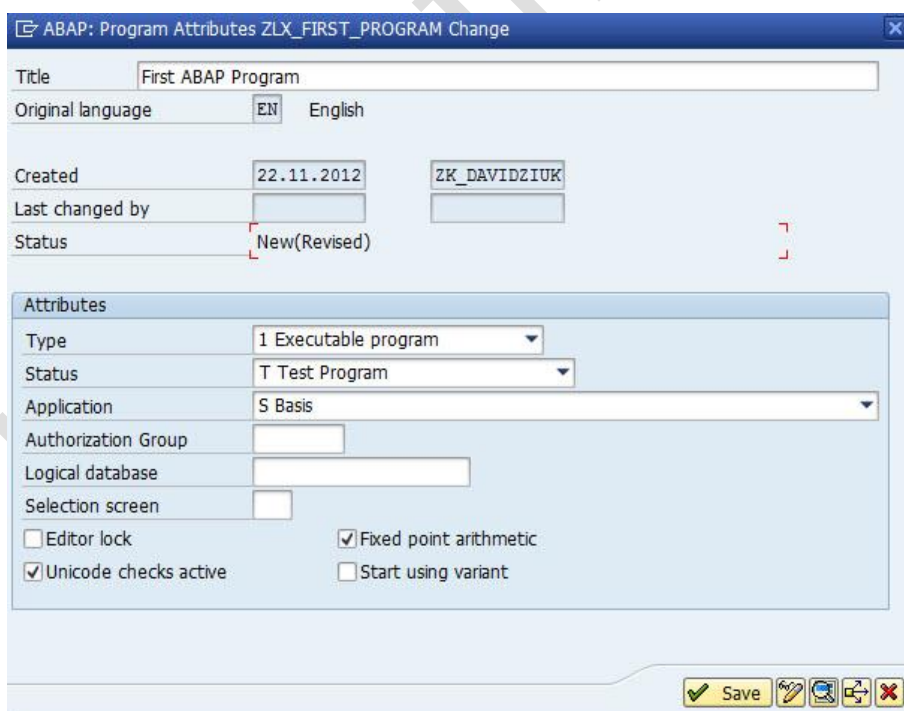
Leverx International

How to create and execute simple report “Hello World”


1. To create ABAP programs directly from within the ABAP Editor, select the menu path Tools → ABAP Workbench → Development → ABAP Editor from the SAP menu of SAP Easy Access (or start Transaction SE38).
2. Type Program Name select “Source Code” in subobjects and press button “Create”.



3. Type Title and select following Attributes: Type – “Executable program”, Status – “Test Program” and Application – “Basis”. Press “Save”.



4. In window "Create Object Directory Entry" Select Package, entry Personal Responsible and press "Save".

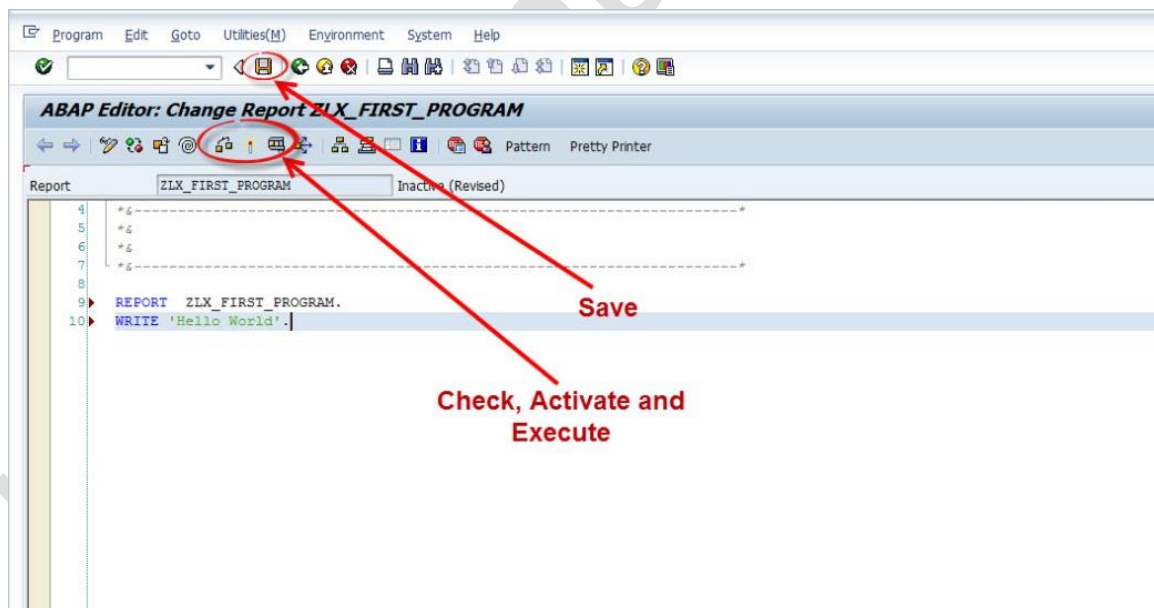


5. In ABAP Editor type our first Program.

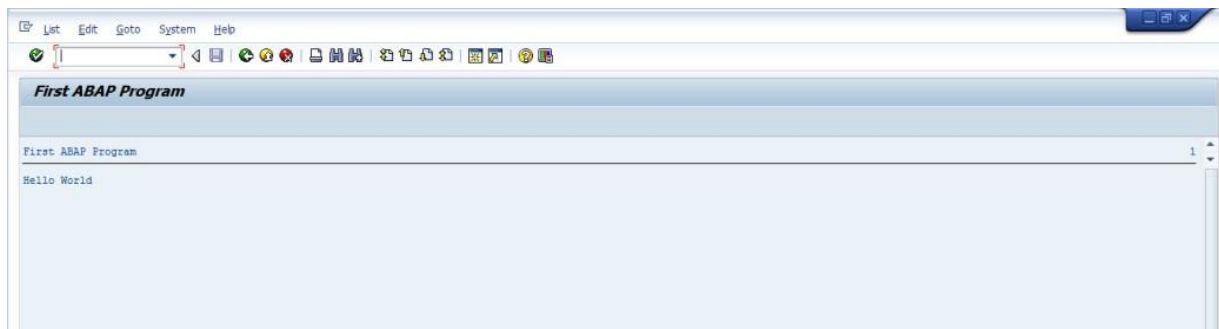
REPORT zlx_first_program.

WRITE 'Hello World'.

6. Check, Save, Activate and Execute it.



7. Output screen is follow:



Appendix 1 - Important SY-* Fields

System fields for screens

sy-cucol	Horizontal cursor position counting starts at column 2
sy-curow	Vertical cursor position counting starts at row 1
sy-datar	Set to X at time of PAI if at least one screen input field was changed by the user or other data transfer, otherwise empty.
sy-loopc	Number of rows currently displayed in a screen table (table control).
sy-scpls	Number of columns in current screen.
sy-srows	Number of rows in current screen.
sy-stepl	Index of current row in a screen table (table control). Is set at every loop pass. Outside a loop, for example during the POV (Process On Value Request) event for a table row, SY-STEPL is not set appropriately.
sy-title	Text that appears in the title bar of the screen. For selection screens and lists this is the program name, otherwise SAP R/3. Can be set during PBO (Process Before Output) using SET TITLEBAR.
sy-ucomm	Function code that triggered the event PAI. Every user action that triggers PAI is assigned a unique function code, with one exception. Choosing Enter triggers PAI and different function codes can be transferred to SY-UCOMM: <ul style="list-style-type: none"> • If there is an entry in the command field of the standard toolbar, this is transferred to SY-UCOMM as the function code. • If there is no entry in the command field and a function code is assigned to the ENTER key, this function code is transferred to SY-UCOMM. • If the command field does not contain an entry and no function code is assigned to the ENTER key, it is empty and the content of SY-UCOMM is not affected.

Selection screens

sy-slset	Variant used to fill the selection screen.
-----------------	--

System fields

sy-subrc	Return value after ABAP statements
sy-index	Loops, current loop pass
sy-tabix	Internal table, current line index
sy-tfill	Internal table, current no. of lines
sy-lilli	List processing, current list line
sy-lsind	List processing, details list index
sy-dbcnt	DB operations, no. of table lines processed
sy-cprog	ABAP program, caller in external procedures
sy-datum	Displays current date
sy-dynnr	ABAP program, no. of current screen
sy-tleng	Line length
sy-stepl	Loop information in table control
sy-loopc	Loop information in table control
sy-fdpos	Contains off sets for the found string
sy-dbsys	Database system
sy-dyngr	Screen group of current screen
sy-dynnr	Number of current screen
sy-msgid	Message ID
sy-msgno	Message number
sy-msgty	Message type (E,I,W,...)
sy-msgv1	Message variable
sy-msgv2	Message variable
sy-msgv3	Message variable
sy-msgv4	Message variable
sy-pagno	Runtime: current page in list
sy-pfkey	Current GUI Status
sy-colno	Current list column
sy-linct	Page length of list

sy-linno	Current line in list
sy-linsz	Line width of list
sy-macol	Number of columns on left margin of print list
sy-binpt	Program running under batch input
sy-modno	Index of External Session

System fields for date and time

sy-datlo	Local date of user, for example 19981129, 1999062
sy-datum	Current (application server) date, for example 19981130, 19990627
sy-dayst	During daylight saving time X, otherwise empty
sy-fdayw	Factory calendar weekday, Sunday 0 ... Saturday 6
sy-timlo	Local time of user, for example 154353, 225312
sy-tzone	Time difference to Greenwich Mean Time (UTC) in seconds, for example 3600, 10800
sy-uzeit	Current (application server) time, for example 164353, 215312
sy-zonlo	Time zone of user, for example CET, UTC

System fields in lists

sy-lsind	Index of the list for current event for basic list = 0
sy-listi	Index of the list level from which the event was triggered
sy-lilli	Absolute number of the line from which the event was triggered
sy-lisel	Contents of the line from which the event was triggered
sy-curow	Position of the line in the window from which the event was triggered (counting starts with 1)
sy-cucol	Position of the column in the window from which the event was triggered (counting starts with 2)
sy-cpage	Page number of the first displayed page of the list from which the event was triggered
sy-staro	Number of the first line of the first page displayed of the list from which the event was triggered (counting starts with 1). This line may contain the page header
sy-staco	Number of the first column displayed in the list from which the event was triggered (counting starts with 1)
sy-ucomm	Function code that triggered the event
sy-pfkey	Status of the list currently being displayed

Appendix 2 - ABAP Types

Predefined Elementary ABAP Types with Fixed Length

Data Type	Initial field length	Initial value	Meaning
Numeric types			
I	4	0	Integer (whole number)
F	8	0	Floating point number
P	8	0	Packed number
Character types			
C	1	' '	Text field (alphanumeric characters)
D	8	00000000	Date field (Format: YYYYMMDD)
N	1	0	Numeric text field (numeric characters)
T	6	000000	Time field (format: HHMMSS)
Hexadecimal type			
X	1	X'0'	Hexadecimal field

Predefined Elementary ABAP Types with Variable Length

- **STRING** for character strings
A string is a sequence of characters with variable length. A string can contain any number of alphanumeric characters. The length of a string is the number of characters multiplied by the length required for the internal representation of a single character.
- **XSTRING** for byte strings
A byte string is a hexadecimal type with variable length. It can contain any number of bytes. The length of a byte string is the same as the number of bytes.

Appendix 3 - Data Elements Elementary Types

Dictionary type	ABAP type	Maximum length n	Meaning
DEC	P((n+1)/2)	1-31, 1-17 in tables	Calculation/amount field
INT1	Internal only	3	Single-byte integer
INT2	Internal only	5	Two-byte integer
INT4	I	10	Four-byte integer
CURR	P((n+1)/2)	1-17	Currency field
CUKY	C(5)	5	Currency key
QUAN	P((n+1)/2)	1-17	Quantity
UNIT	C(n)	2-3	Unit
PREC	Internal only	16	Accuracy
FLTP	F(8)	16	Floating point number
NUMC	N(n)	1-255	Numeric text
CHAR	C(n)	1-255	Character
LCHR	C(n)	256-max	Long character
STRING	STRING	1-max	String of variable length
RAWSTRING	XSTRING	1-max	Byte sequence of variable length
DATS	D	8	Date
ACCP	N(6)	6	Accounting period YYYYMM
TIMS	T	6	Time HHMMSS
RAW	X(n)	1-255	Byte sequence
LRAW	X(n)	256-max	Long byte sequence
CLNT	C(3)	3	Client
LANG	C(1)	internal 1, external 2	Language

Appendix 4 - ABAP Statements Overview

ADD for single fields	Adds two single fields
ADD for field sequences	Adds sequences of fields in storage
ADD-CORRESPONDING	Adds subfields of structures
ALIASES	Defines class-specific alias names for an interface component in ABAP objects
APPEND	Appends a line or multiple lines to the end of an index table
ASSIGN	Assigns a field to a field symbol
AT for event blocks	Event keywords for defining event blocks for screen events
AT for group change	Change of group when processing loops of extracts and internal tables
AUTHORITY-CHECK	Checks user authorization
BACK	Relative position of the output in a list
BREAK-POINT	Calls the Debugger
CALL CUSTOMER-FUNCTION	Calls the customer function modules
CALL FUNCTION	Calls the function modules
CALL DIALOG	Calls a dialog module
CALL METHOD	Calls a method in ABAP Objects
CALL METHOD OF	Calls a method in OLE2 Automation
CALL SCREEN	Calls a screen sequence
CALL SELECTION-SCREEN	Calls a selection screen
CALL TRANSACTION	Call a transaction
CASE	Conditional branch
CATCH	Catches a class-based exception
CATCH SYSTEM-EXCEPTIONS	Catches a catchable runtime error
CHECK	Conditionally leaves a loop or processing block
CLASS Declaration	Declares a class in ABAP Objects
CLASS < > IMPLEMENTATION	Implements a class in ABAP Objects
CLASS-DATA	Declares static attributes of a class or interface
CLASS-METHODS	Declares static methods of a class or interface
CLASS-EVENTS	Declares static events of a class or interface
CLEANUP	Tidies up after class-based exceptions
CLEAR	Sets a variable to its initial value
CLOSE DATASET	Closes a file
CLOSE CURSOR	Closes a database cursor
COLLECT	Aggregates lines and then adds them to an internal table
COMMIT	Closes a SAP LUW
COMMUNICATION	Enables two programs to communicate with each other
COMPUTE	Performs numeric operations
CONCATENATE	Concatenates (chains) several strings to a string
CONDENSE	Removes spaces from a string
CONSTANTS	Declares constant data objects
CONTINUE	Ends a loop pass
CONTEXTS	Declares a context
CONTROLS	Defines a control
CONVERT for Dates	Converts a data into an inverted date form
CONVERT for Timestamps	Converts a timestamp into the correct date and time for the current time zone
CONVERT for Text	Converts a text into a format that can be sorted alphabetically
CREATE DATA	Creates a dynamic data object
CREATE OBJECT in ABAP Objects	Instantiates an object in ABAP Objects
CREATE OBJECT in OLE2 Automation	Instantiates an external object in OLE2 Automation
DATA with Reference to Known Data Types	Declares variables with a previously-declared data type
DATA with Reference to Generic Data Types	Declares variables by completing the description of a generic type

DATA, Creating an Associated Data Type	Declares variables with data types that only exist as an attribute of the variable
DATA statement for Shared Data Areas	Declares shared data areas in a program
DEFINE	Defines a macro
DELETE for Files	Deletes files on the application server
DELETE for Database Table Entries	Deletes entries from database tables
DELETE for Cluster Databases	Deletes data clusters from cluster database tables
DELETE for the Cross-Transaction Application Buffer	Deletes data clusters from the cross-transaction application buffer
DELETE for Lines from an Internal Table	Deletes lines from internal tables of any type
DELETE for Lines from Index Tables	Deletes entries from index tables
DEMAND	Retrieves values from a context instance
DESCRIBE DISTANCE	Gets the distance between two fields
DESCRIBE FIELD	Describes the attributes of a field
DESCRIBE LIST	Describes the attributes of a list
DESCRIBE TABLE	Describes the attributes of an internal table
DIVIDE	Divides one field by another
DIVIDE-CORRESPONDING	Divides matching components of structures
DO	Introduces a loop
EDITOR-CALL	Loads an ABAP program or internal table into a text editor
ELSE	Introduces a statement block in an IF control structure
ELSEIF	Introduces a statement block in an IF control structure
END-OF-DEFINITION	Closes a macro definition
END-OF-PAGE	Event keywords for defining event blocks for list events
END-OF-SELECTION	Event keywords for defining event blocks for reporting events
ENDAT	Closes a statement block in control level processing
ENDCASE	Closes a CASE control structure
ENDCATCH	Closes a CATCH area
ENDCLASS	Closes a class definition
ENDDO	Closes a DO loop
ENDEXEC	Closes a Native SQL statement
ENDFORM	Closes a subroutine
ENDFUNCTION	Closes a function module
ENDIF	Closes an IF control structure
ENDINTERFACE	Closes a interface definition
ENDLOOP	Closes a LOOP
ENDMETHOD	Closes a method
ENDMODULE	Closes a dialog module
ENDON	Closes a conditional statement block
ENDPROVIDE	Closes a PROVIDE loop
ENDSELECT	Closes a SELECT loop
ENDTRY	Closes a TRY area
ENDWHILE	Closes a WHILE loop
EVENTS	Defines events in classes or interfaces
EXEC SQL	Introduces a Native SQL statement
EXIT	Leaves a loop or processing block
EXIT FROM STEP-LOOP	Ends a step loop
EXIT FROM SQL	Ends Native SQL processing
EXPORT	Exports data clusters
EXTRACT	Creates an extract dataset and adds lines to it
FETCH	Uses a cursor to read entries from a database table
FIELD-GROUPS	Declares a field group for an extract dataset
FIELD-SYMBOLS	Declares field symbols
FIND	Searches for patterns
FORM	Defines a subroutine
FORMAT	Sets formatting options for list output

FREE	Release space in memory
FUNCTION	Defines a function module
FUNCTION-POOL	Introduces a function group
GET	Event keyword for defining event blocks for reporting events
GET BIT	Reads an individual bit
GET CURSOR	Gets the cursor position on a screen or in an interactive list event
GET LOCALE LANGUAGE	Gets the current text environment
GET PARAMETER	Gets an SPA/GPA parameters
GET PF-STATUS	Gets the current GUI status
GET PROPERTY	Gets a property of an OLE2 Automation object
GET REFERENCE	Gets a data reference
GET RUN TIME FIELD	Measures the runtime in microseconds
GET TIME	Synchronizes the time
GET TIME STAMP FIELD	Returns a time stamp
HIDE	Stores information about list lines
IF	Conditional branch. Introduces a new branch.
IMPORT	Imports data clusters
IMPORT DIRECTORY	Creates the directory of a data cluster from a cluster database
INCLUDE	Inserts an include program in another program
INCLUDE TYPE STRUCTURE	Includes a structure within another
INITIALIZATION	Event keywords for defining event blocks for reporting events
INSERT for Database Tables	Inserts entries from database tables
INSERT for Field Groups	Defines the structure of field groups for extract datasets
INSERT for any Internal Table	Inserts lines from internal tables of any type
INSERT for Index Tables	Inserts entries in index tables
INSERT for Programs	Inserts ABAP programs into the program library
INTERFACE	Declares a interface in ABAP Objects
INTERFACES	Implementation or nesting of an interface in ABAP Objects
LEAVE for Screens	Leaves a screen
LEAVE for Lists During Screen Processing	Switches between screen and list processing
LEAVE for Programs	Leaves an ABAP program
LOAD-OF-PROGRAM	Event keywords for defining event blocks for a program constructor
LOCAL	Protects global data against changes
LOOP Through Extracts	Introduces a loop through an extract dataset
LOOP THROUGH internal tables	Starts a loop through an extract dataset
LOOP Through Screen Fields	Starts a loop through the table SCREEN
MESSAGE	Displays a message
METHOD	Introduces the implementation of a method in a class
METHODS	Declares methods in classes and interfaces
MODIFY for Database Tables	Inserts or changes lines in database tables
MODIFY for any Internal Table	Changes the content of lines in internal tables of any type
MODIFY for Index Tables	Changes the content of lines in index tables
MODIFY for Lists	Changes a list line
MODIFY SCREEN	Changes the SCREEN table
MODULE	Introduces a dialog module
MOVE	Assigns values
MOVE-CORRESPONDING	Assigns values between identically-named components of structures
MULTIPLY	Multiplies two single fields
MULTIPLY-CORRESPONDING	Multiplies components of structures
NEW-LINE	Inserts a line break in a list
NEW-PAGE	Inserts a page break in a list
NODES	Declares an interface work area
ON CHANGE	Introduces a new branch
OPEN CURSOR	Opens a database cursor
OPEN DATASET	Opens a file
OVERLAY	Overlays one string with another
PACK	Converts type C variables to type P
PARAMETERS	Declares parameters for a selection screen

PERFORM	Calls a subroutine
POSITION	Absolute position of the display in a list
PRINT-CONTROL for Print Format	Specifies the print format
PRINT-CONTROL for Index Lines	Creates index lines in the spool file
PRIVATE	Defines the private section of a class
PROGRAM	Introduces a program
PROTECTED	Defines the protected section of a class
PROVIDE	Loops through internal tables at given intervals
PUBLIC	Defines the public section of a class
PUT	Triggers a GET event
RAISE for Class-Based Exceptions	Raises an exception
RAISE for Exceptions of Function Modules and Methods	Raises exceptions
RAISE for Events	Triggers events in ABAP Objects
RANGES	Declares a RANGES table
READ for Files	Reads a file
READ for any Internal Table	Reads a line of an internal table
READ for Index Tables	Reads a line of an internal table
READ for Lists	Reads the contents of a line from a list
READ for Programs	Reads ABAP programs into the program library
RECEIVE	Receives results from an asynchronous function module call
REFRESH	Initializes an internal table
REFRESH CONTROL	Initializes a control
REJECT	Leaves an GET processing block
REPLACE by Pattern	Replaces strings in fields with other strings using a pattern
REPLACE by Position	Replaces strings in fields with other strings by position
REPORT	Introduces a program
RESERVE	Inserts a conditional page break in a list
RETURN	Leaves a processing block
ROLLBACK	Undoes the changes in a SAP LUW
SCROLL	Scrolls through lists
SEARCH	Searches for strings
SELECT	Reads data from the database
SELECT-OPTIONS	Declares selection criteria for a selection screen
SELECTION-SCREEN for Selection Screen Formatting	Formats selection screens
SELECTION-SCREEN for Selection Screen Definition	Defines selection screens
SELECTION-SCREEN for Selection Screen Versions	Defines selection screen versions
SELECTION-SCREEN for Logical Databases	Provides special functions
SET BIT	Sets individual bits
SET BLANK LINES	Allows blank lines in lists
SET COUNTRY	Sets output Formats
SET CURSOR	Sets the cursor on the screen
SET DATASET	Sets the attributes of a file
SET EXTENDED CHECK	Affects the extended program check
SET HANDLER	Registers event handlers in ABAP Objects
SET HOLD DATA	Sets a screen attribute
SET LANGUAGE	Sets the display language
SET LEFT SCROLL BOUNDARY	Sets the left boundary for horizontal scrolling
SET LOCALE LANGUAGE	Sets the current text environment
SET MARGIN	Sets the margin of a print page
SET PARAMETER	Sets an SPA/GPA parameters
SET PF-STATUS	Sets the GUI status
SET PROPERTY	Sets a property of an OLE2 Automation object
SET RUN TIME ANALYZER	Controls runtime analysis
SET RUN TIME CLOCK	Controls runtime analysis
SET SCREEN	Sets the next screen

SET TITLEBAR	Sets the screen title
SET UPDATE TASK LOCAL	Switches on local update
SET USER-COMMAND	Triggers a list event
SHIFT	Shifts strings
SKIP for Blank Lines	Generates blank lines in the display list
SKIP for Positioning	Absolute position of the display in a list
SORT for Extracts	Sorts an extract dataset
SORT for Internal Tables	Sorts internal tables
SPLIT	Splits a string
START-OF-SELECTION	Event keywords for defining event blocks for reporting events
STATICS	Defines static variables
STOP	Leaves a reporting event
SUBMIT	Calls an executable program of type 1
SUBTRACT for single fields	Subtracts two single fields
SUBTRACT-CORRESPONDING	Subtracts components of structures
SUM	Calculates sums of groups.
SUPPLY	Fills context instances with values
SUPPRESS DIALOG	Prevents the current screen from being displayed
TABLES	Declares an interface work area
TOP-OF-PAGE	Event keywords for defining event blocks for list events
TRANSFER	Writes to a file
TRANSLATE	Converts characters to strings
TRY	Introduces a TRY block
TYPE-POOL	Introduces a type group
TYPE-POOLS	Declares the types and constants of a type group to a program
TYPES for Single Field Types	Defines a single field type
TYPES for Complex Types	Defines complex types
ULINE	Places horizontal lines in the display list
UNPACK	Converts type P variables to type C
UPDATE	Changes entries in database tables
WHEN	Introduces a statement block in a CASE control structure
WHILE	Introduces a loop
WINDOW	Displays a list as a modal dialog box
WRITE	Displays lists
WRITE TO	Assigns string values

Appendix 5 - Naming conventions in ABAP code (example)

- Global/Local/Member – G[*]_... / L[*]_.../ M[*]_...
- Types - [G/L/M]TY_...
- Variables - [G/L/M]V_...
- Structures - [G/L/M]S_...
- Internal Tables - [G/L/M]T_...
- Constants - [G/L/M]C_...
- Objects - [G/L/M]O_...
- Field Symbols - <[G/L/M][V/S/T/...]>_...
- Parameters - [Exporting/Importing/Etc][V/S/T/...]_...
- ...