ABAP Introduction with HANA

2018/09/13
NTT Data Taiwan Co., Ltd
Michael Chuang

© 2018 NTT DATA Taiwan Co., Ltd.

Agenda

- Use SAP GUI for Windows
- Flow of an ABAP Program
- Introduction to the ABAP Workbench
- Introduction to the ABAP Dictionary
- Basic ABAP Language Elements
- Modularization
- Data Retrieval
- Introduction to the Eclipse/ADT
- Code-to-Data with ABAP
- User Dialog

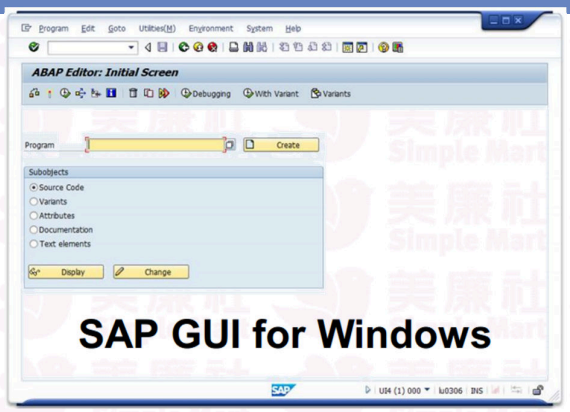
© 2018 NTT DATA Taiwan Co., Ltd.

Use SAP GUI for Windows

© 2018 NTT DATA Taiwan Co., Ltd.

SAP GUI Family

- **SAP GUI for Windows**
- SAP GUI for Java
- SAP GUI for HTML

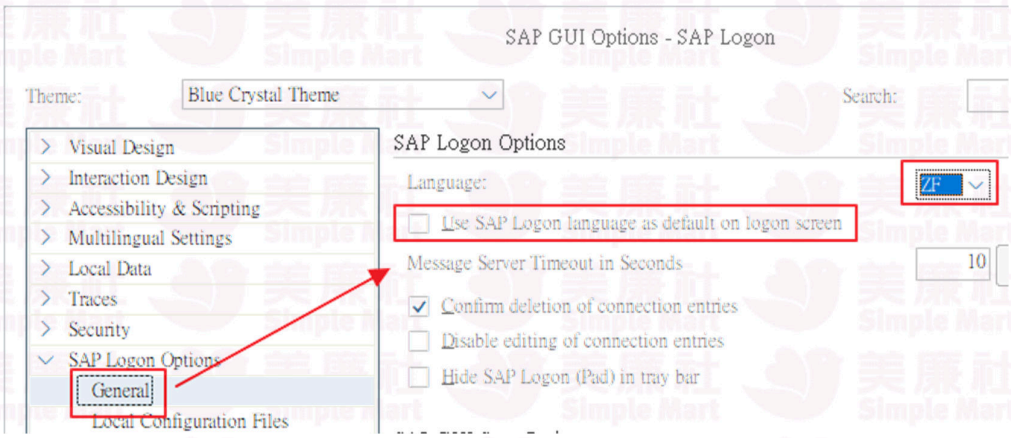


© 2018 NTT DATA Taiwan Co., Ltd.

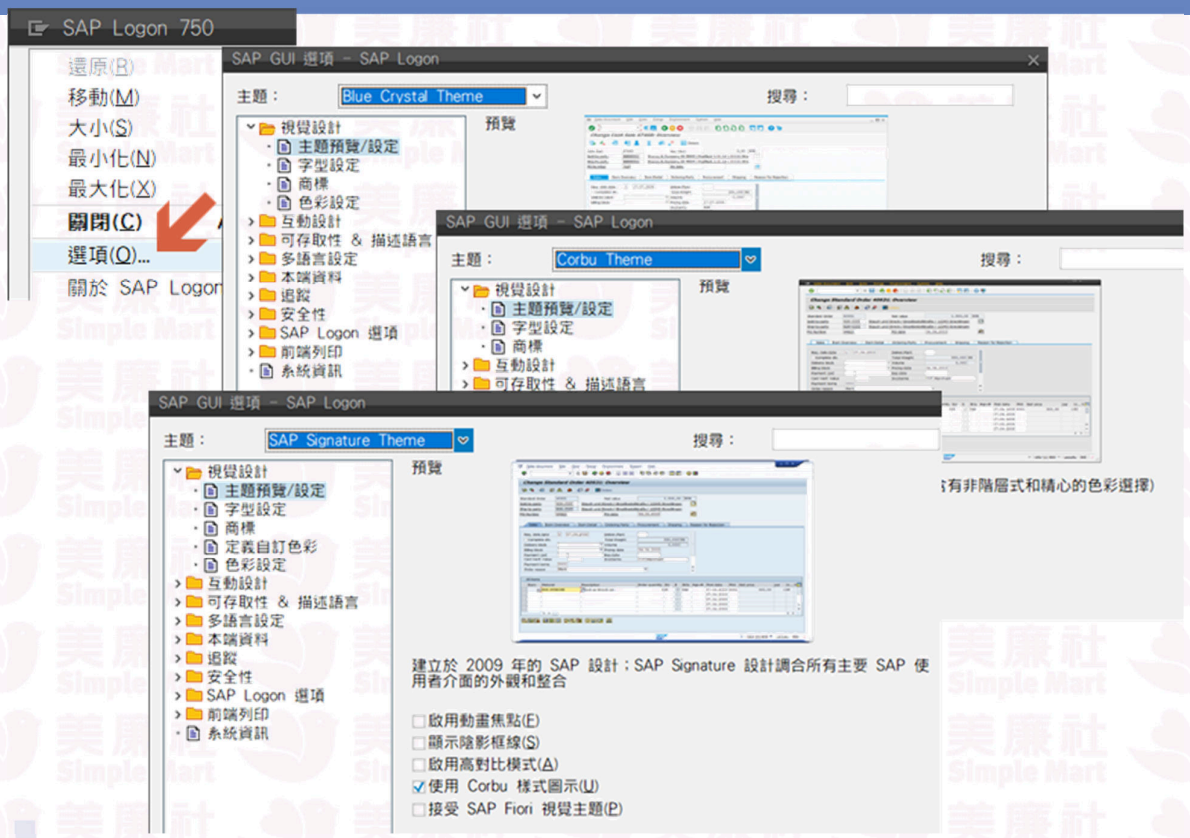
SAP Logon – 變更佈景主題

若使用 SAP Logon 750 版本，因預設啟用的 SAP Fiori / Belize 佈景主題，會隱藏部分開發環境的功能按鈕，建議取消

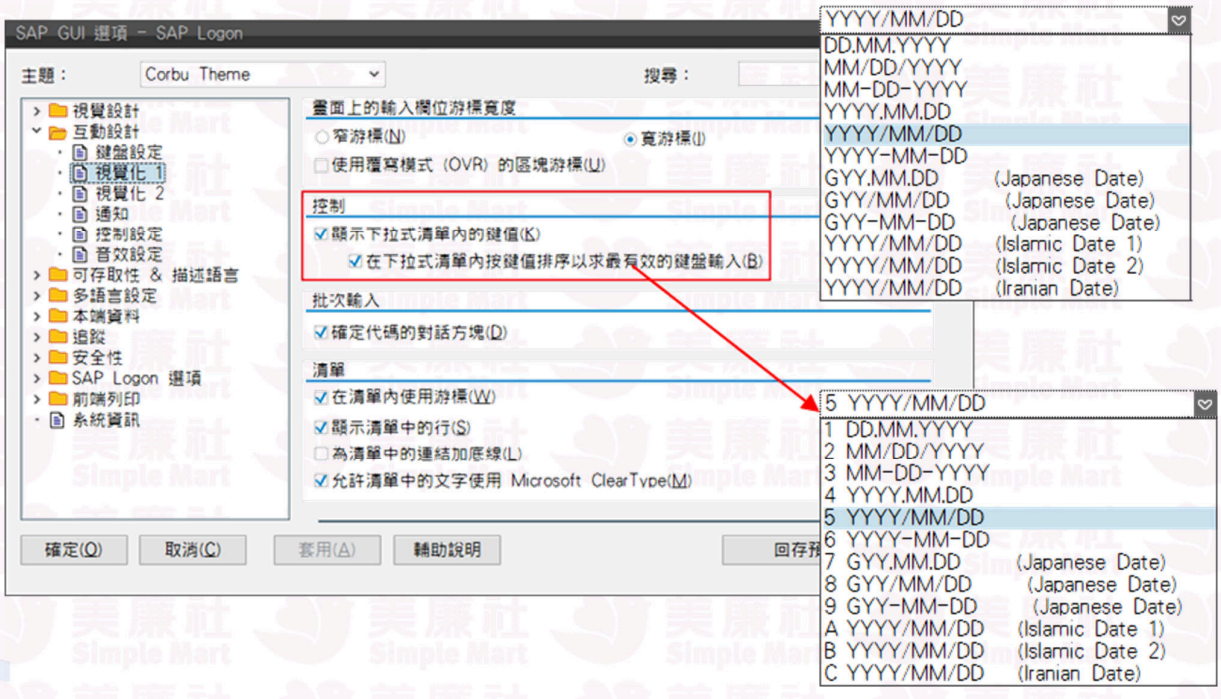
SAP Logon – 變更語言



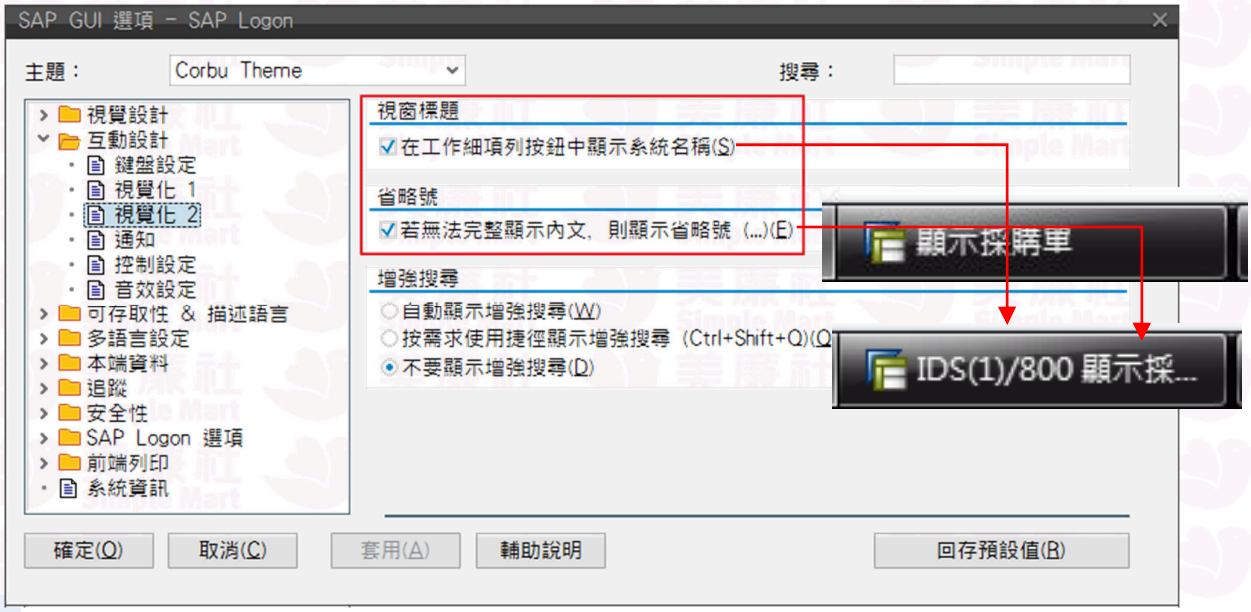
SAP GUI Theme



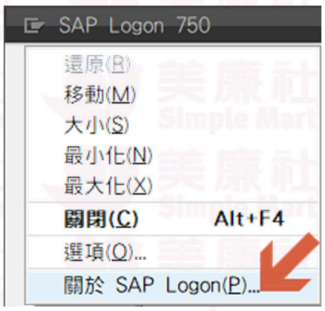
SAP GUI Option – Visualization 1



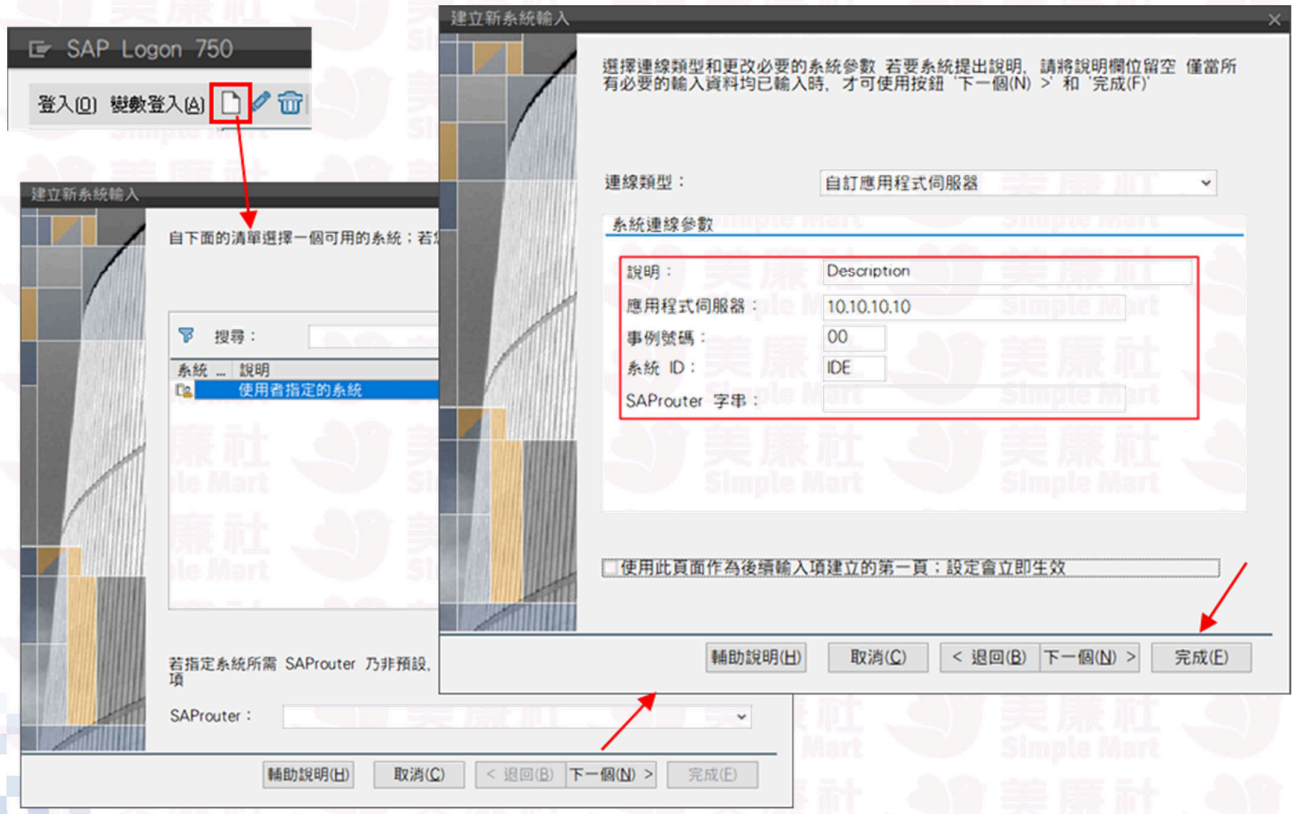
SAP GUI Option – Visualization 2



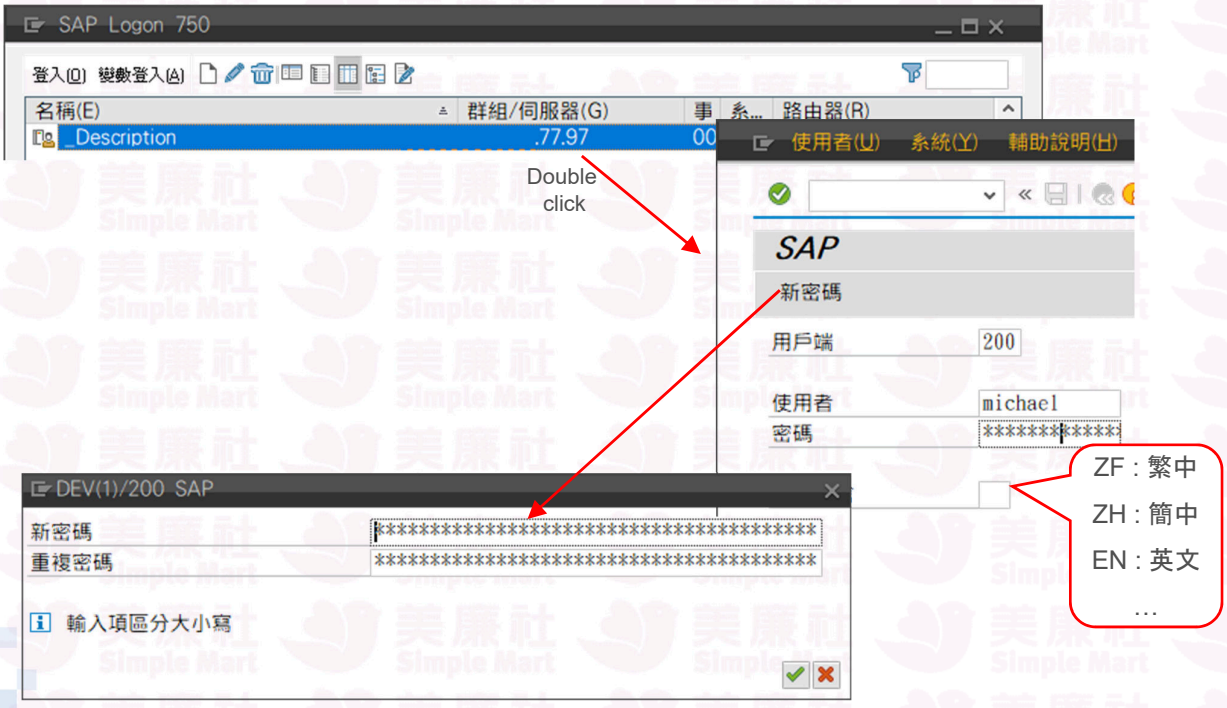
SAP GUI Option – System Information



SAP Logon – 建立Server連線參數



登入畫面與變更密碼

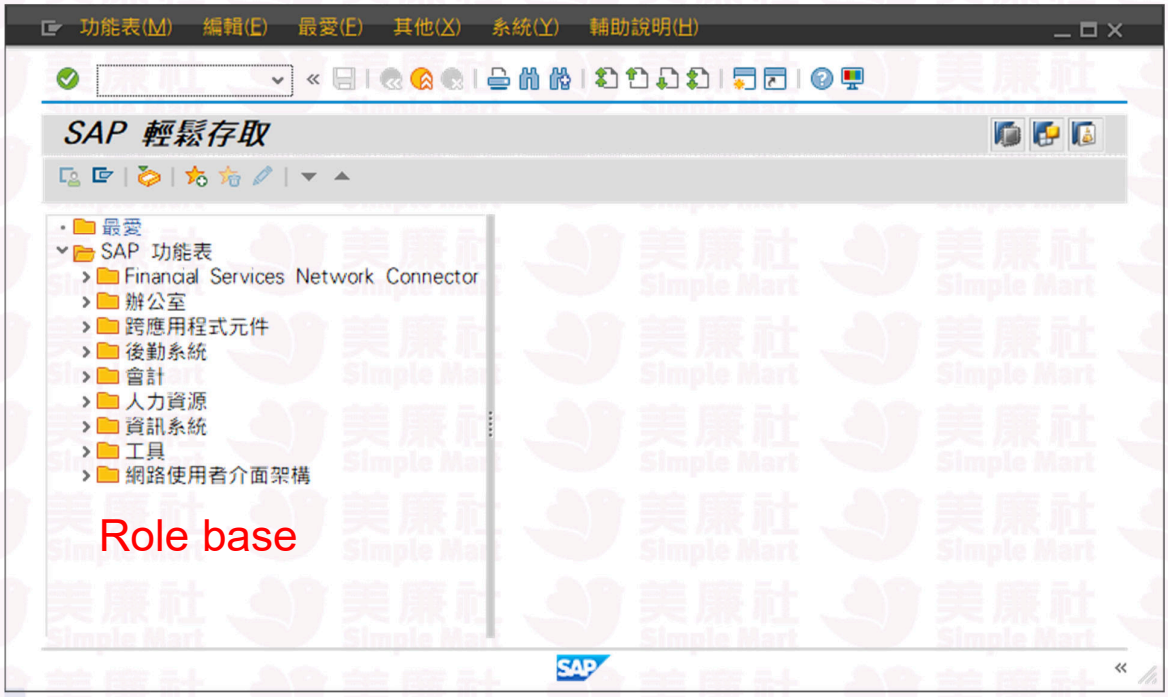


重複登入

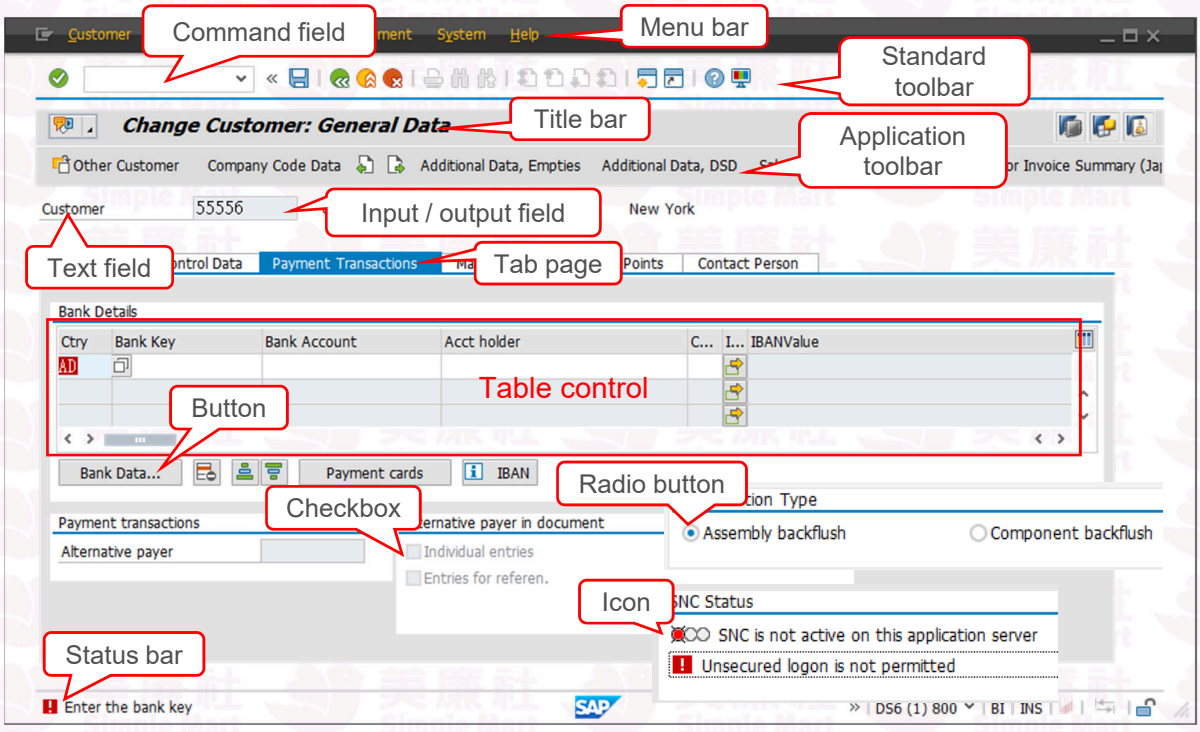
SAP “Client” is ?

- In Business => 財務報表
  - 正式環境僅需一個 client ，即可支援集團內多公司的系統運作
- In Tech. => 完整的開發/測試環境
  - **DEV 100: IMG (Golden Client)**
  - **DEV 200: Development**
  - *DEV 400: test 1 (Sandbox)*
  - *DEV 500: test 2*
  - *QAS 300: User test 1*
  - *QAS 400: User test 2*
  - **PRD 888: 正式環境**

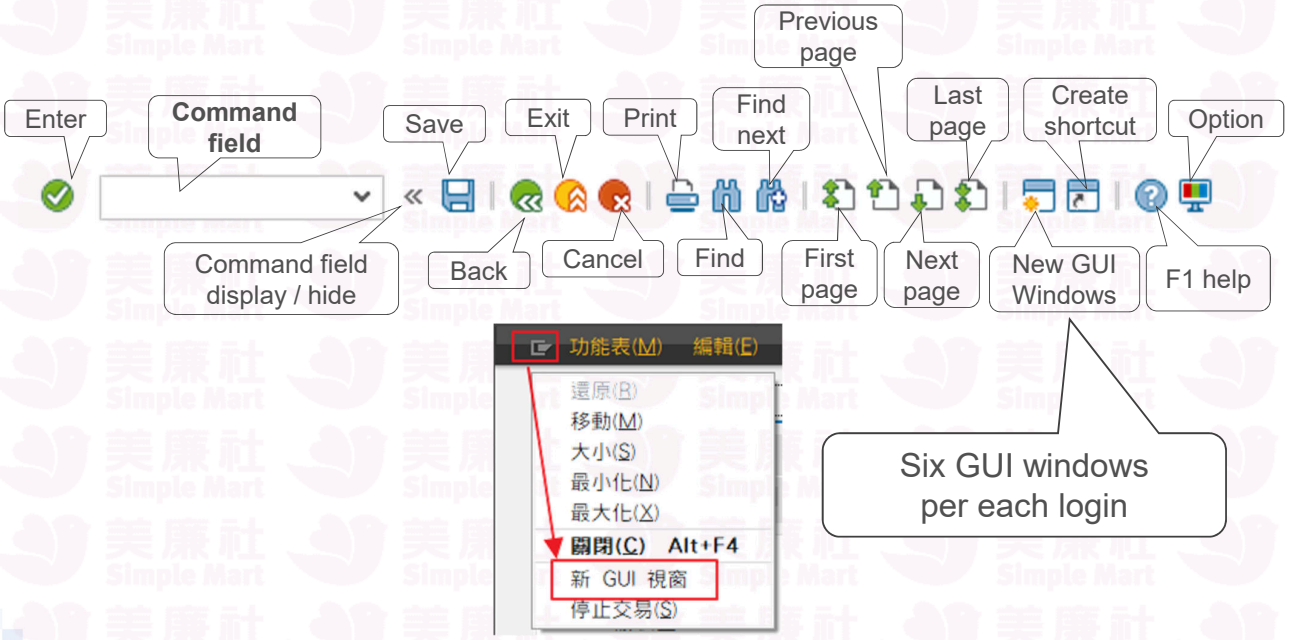
登入後系統初始畫面



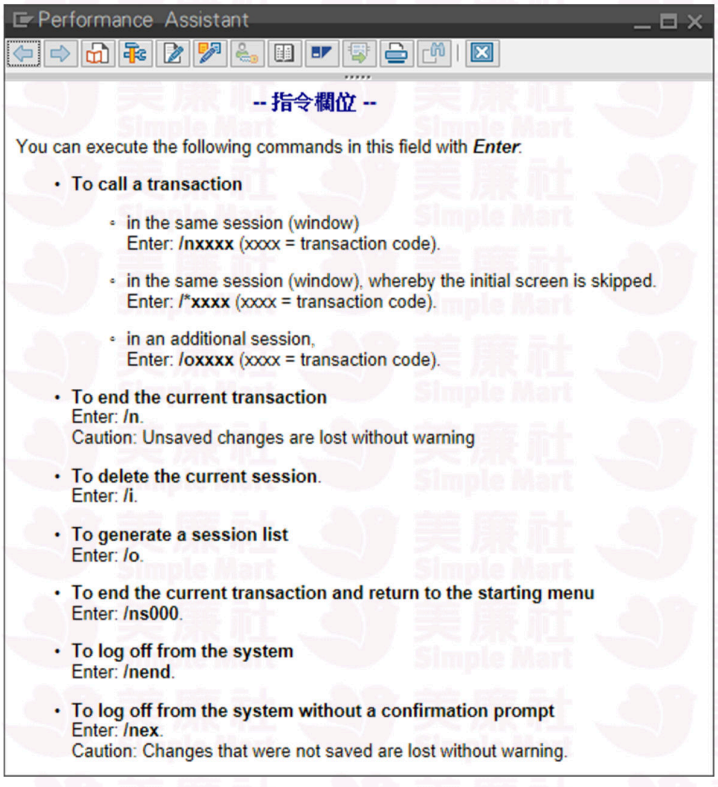
SAP GUI畫面元件介紹



Menu bar and Standard Toolbar

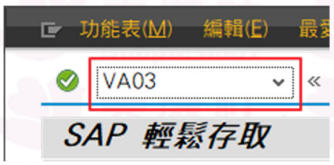


Command field



Transaction Code

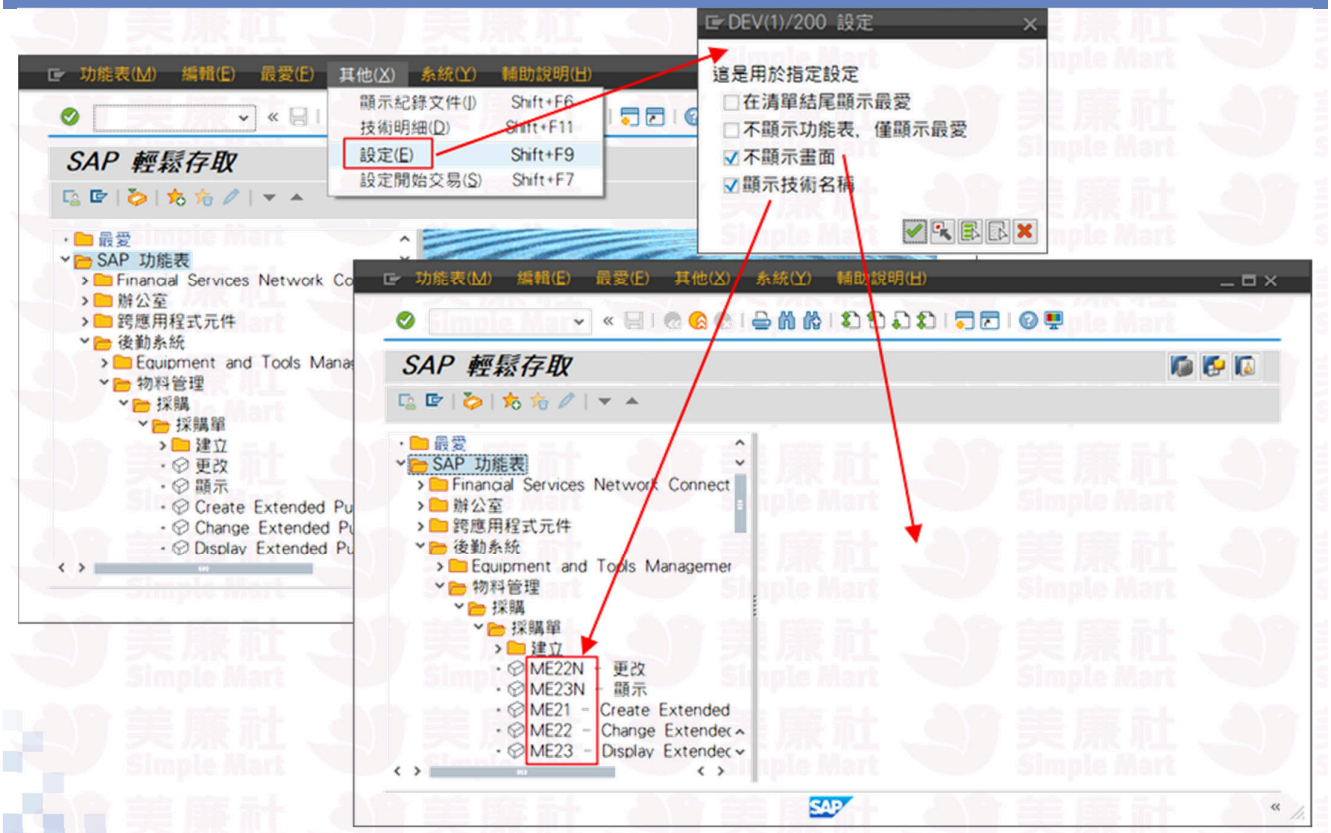
- What is Transaction Code?
  - Shortcut to the application
  - e.g. Create Sales Order➔ VA01
- Transaction Code = T-Code = Technical Name
- Where to key in?
  - Command field
- How to find T-Code?
  - 1. Easy access
  - 2. Status Setting
  - 3. System Menu



© 2018 NTT DATA Taiwan Co., Ltd.

NTT DATA

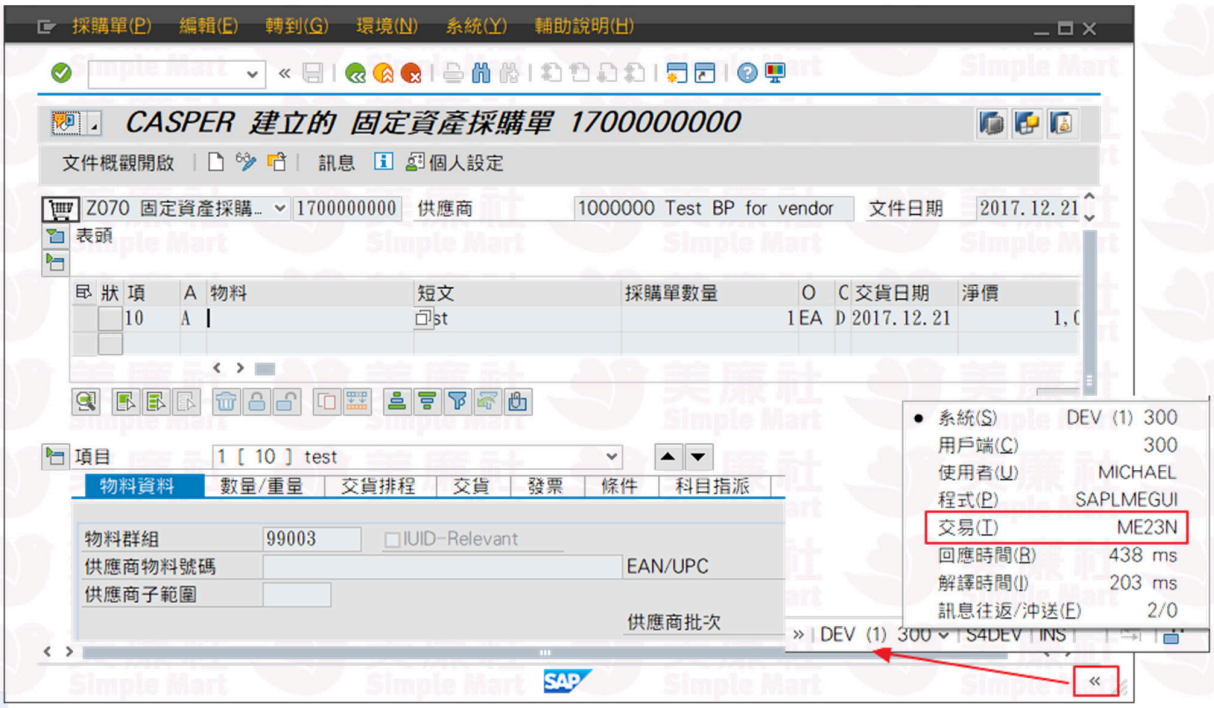
Selection Functions



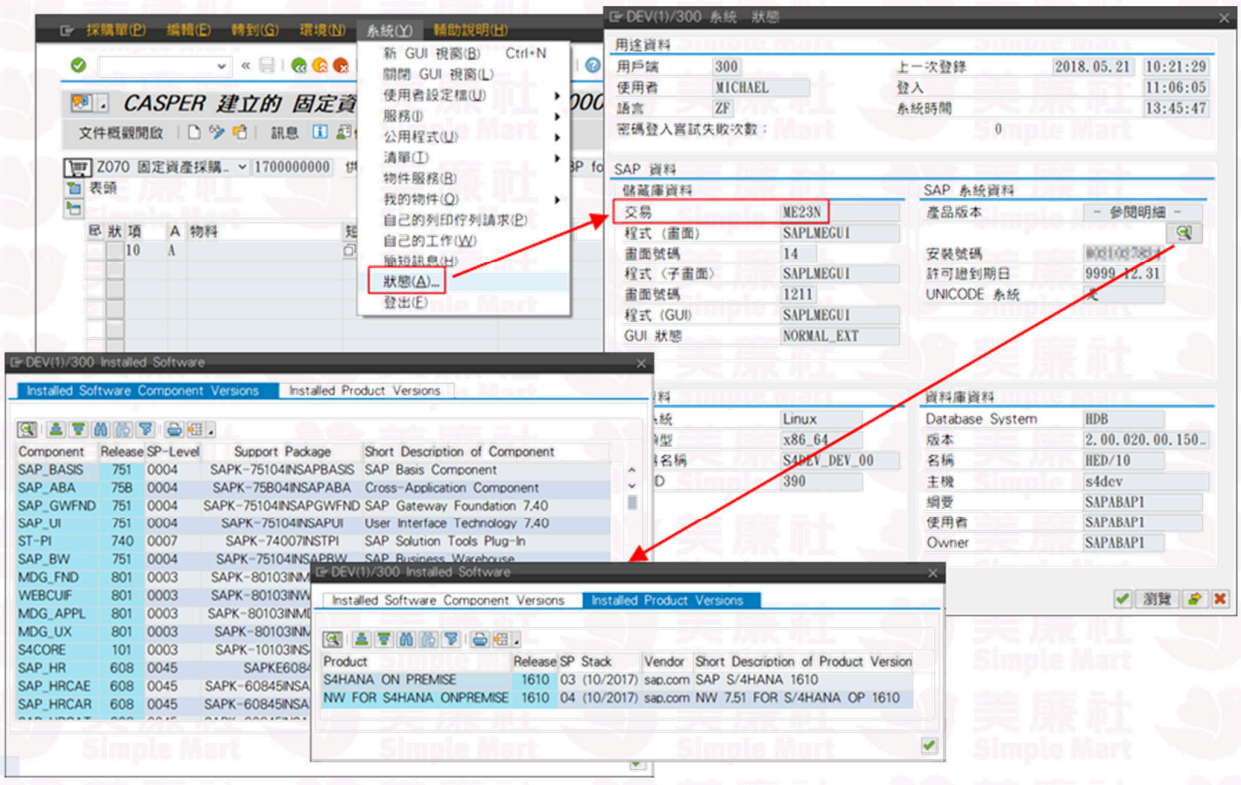
© 2018 NTT DATA Taiwan Co., Ltd.

NTT DATA

Find T-Code from Status Setting



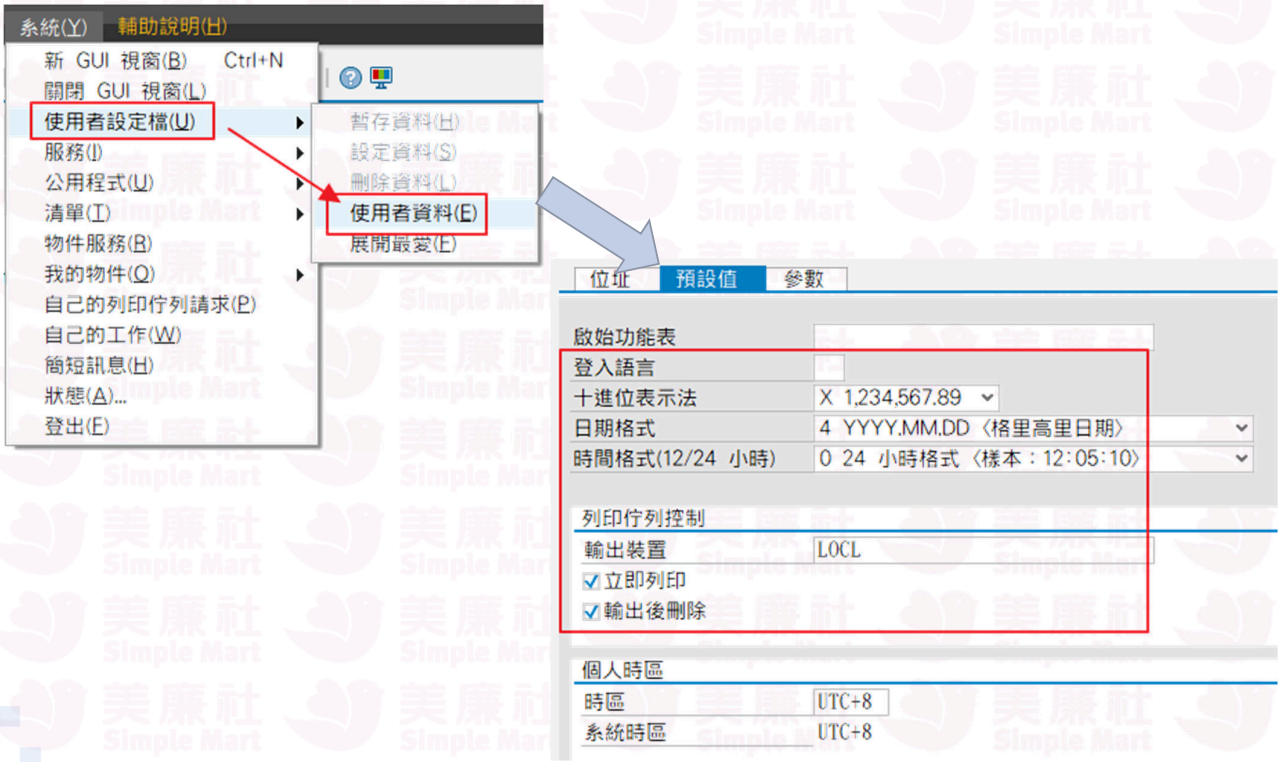
Find T-Code from System Menu



The F1 help (Field Help)

The F4 help (Input Help)

User Profile



© 2018 NTT DATA Taiwan Co., Ltd.

NTT DATA

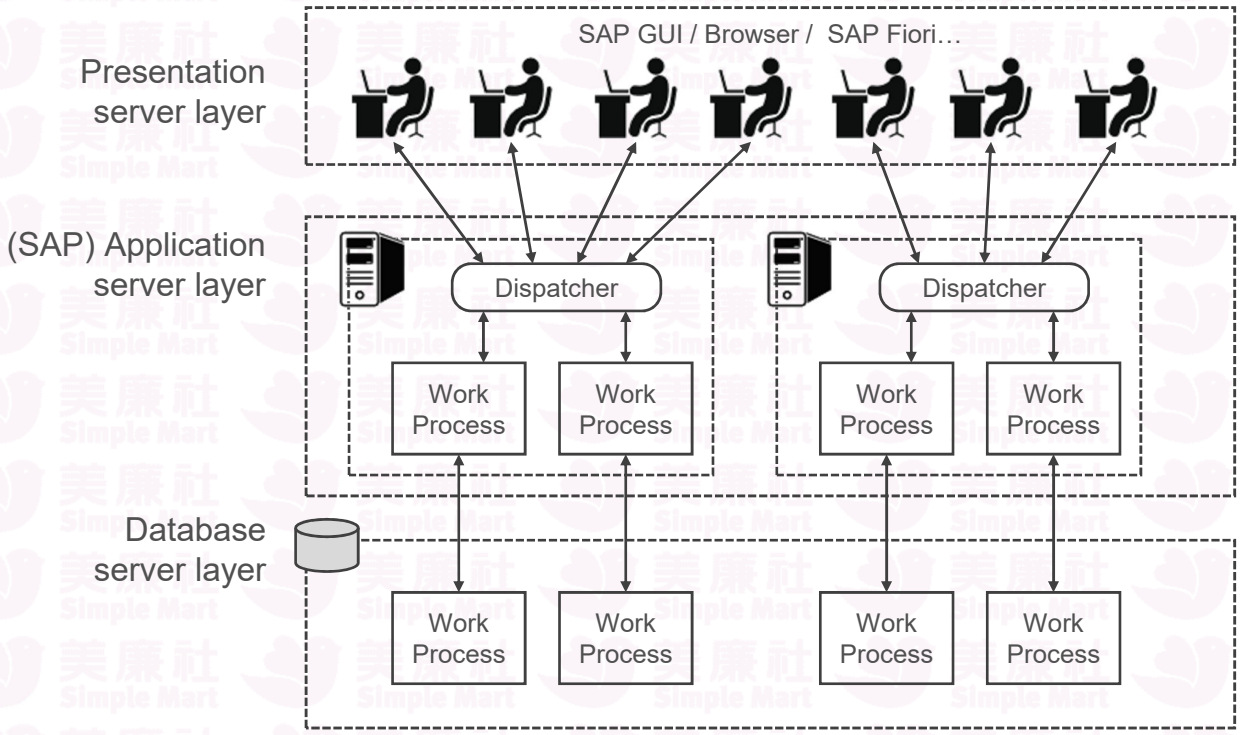
Flow of an ABAP Program

- System Architecture and ABAP Program Using Example of Selection Screen and List

© 2018 NTT DATA Taiwan Co., Ltd.

NTT DATA

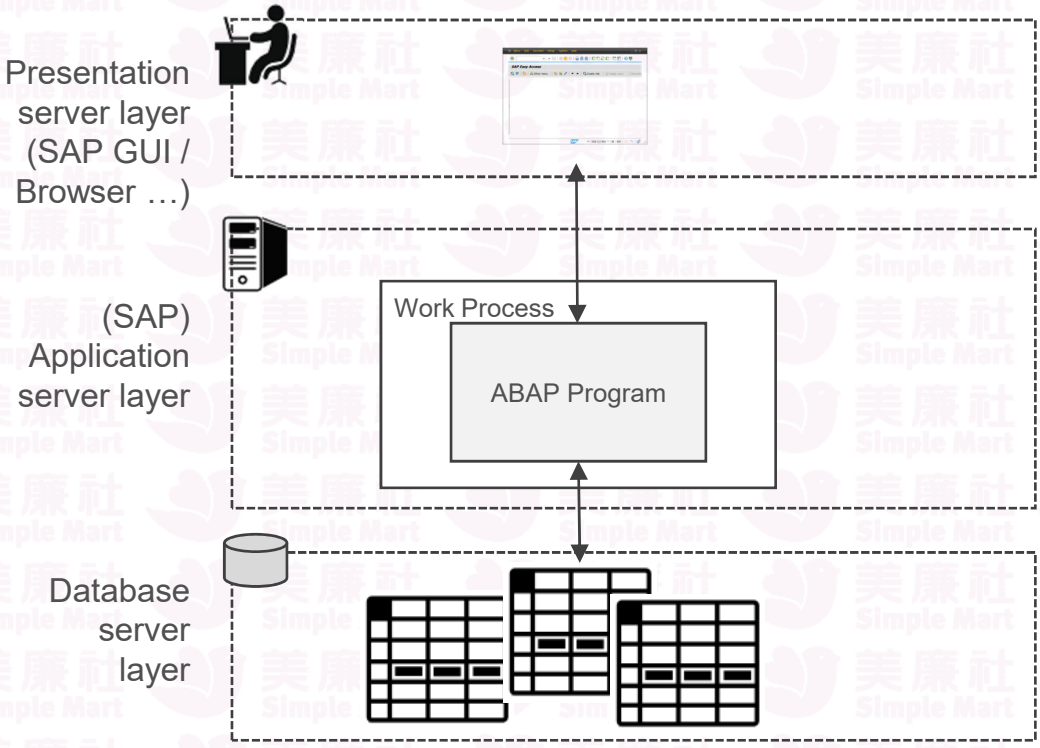
Client/Server Architecture of the SAP NetWeaver Application Server



© 2018 NTT DATA Taiwan Co., Ltd.

NTT DATA

Excerpt for an ABAP Program

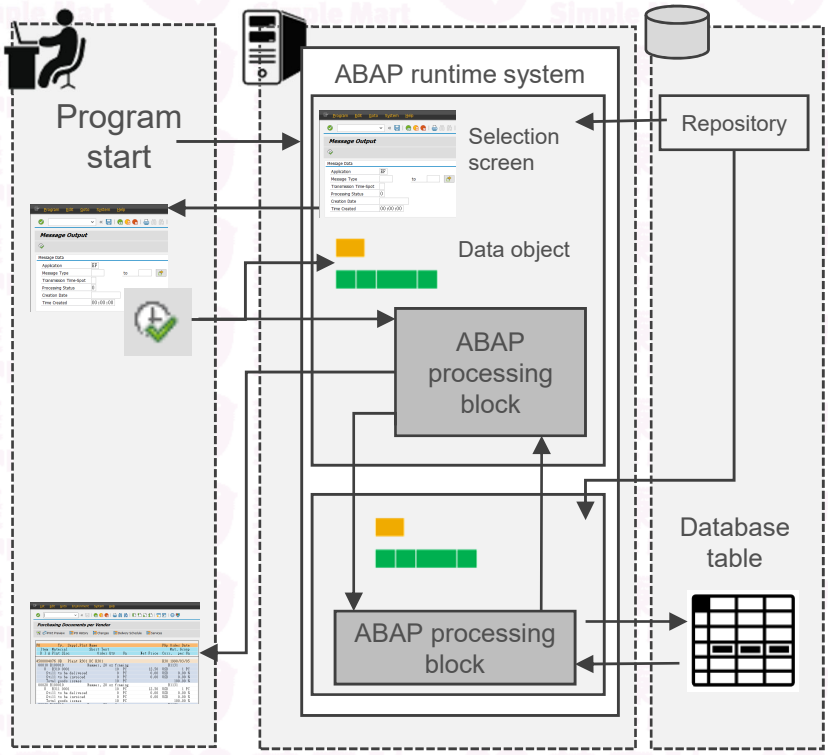


© 2018 NTT DATA Taiwan Co., Ltd.

NTT DATA

Interaction Between Server Layers

1. Program start
2. System loads Program Context
3. Runtime System Sends Selection Screen
4. Input Values Are Inserted into Data Objects
5. Reuse Unit Loading
6. Reuse Unit is Called
7. Program Requests Data Record from the Database
8. Database Supplies the Data Record(s)
9. Reuse Unit Returns the Data
10. Runtime System Sends List



© 2018 NTT DATA Taiwan Co., Ltd.

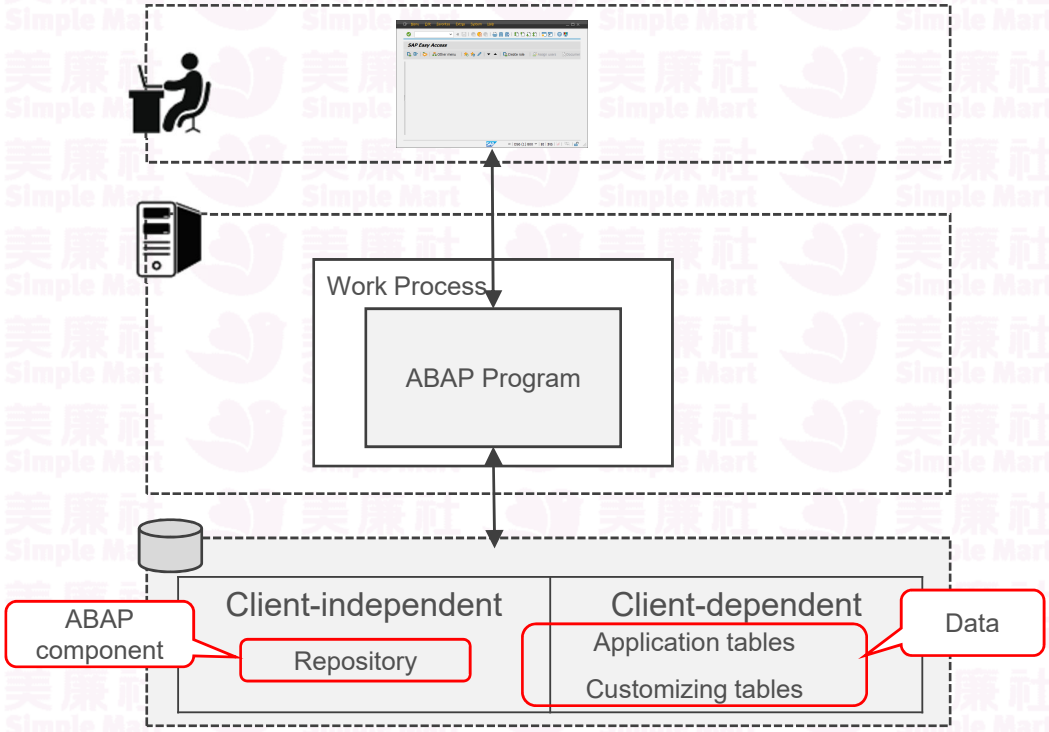
Introduction to the ABAP Workbench

✓**Repository and Object Navigator**

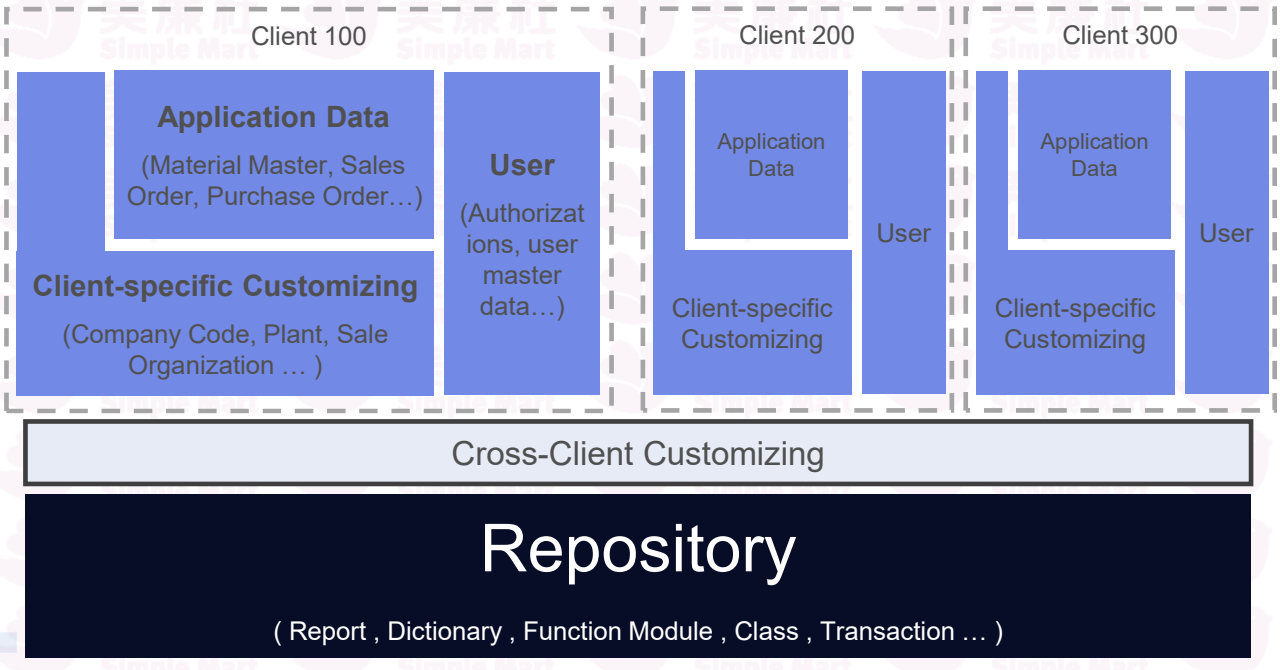
✓**Developing Programs and Organizing Developments**

© 2018 NTT DATA Taiwan Co., Ltd.

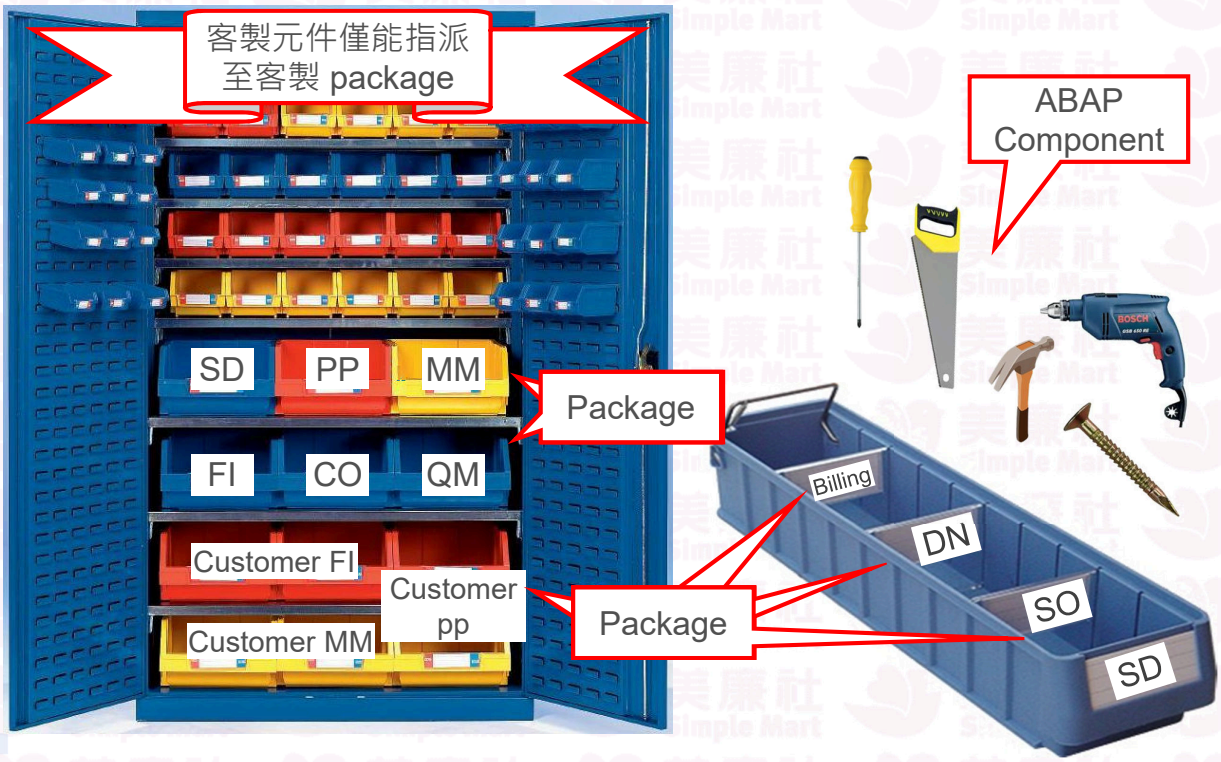
The Cross-Client Capability of the Repository



Data Structure of SAP ERP Systems



Structure of the Repository



© 2018 NTT DATA Taiwan Co., Ltd.

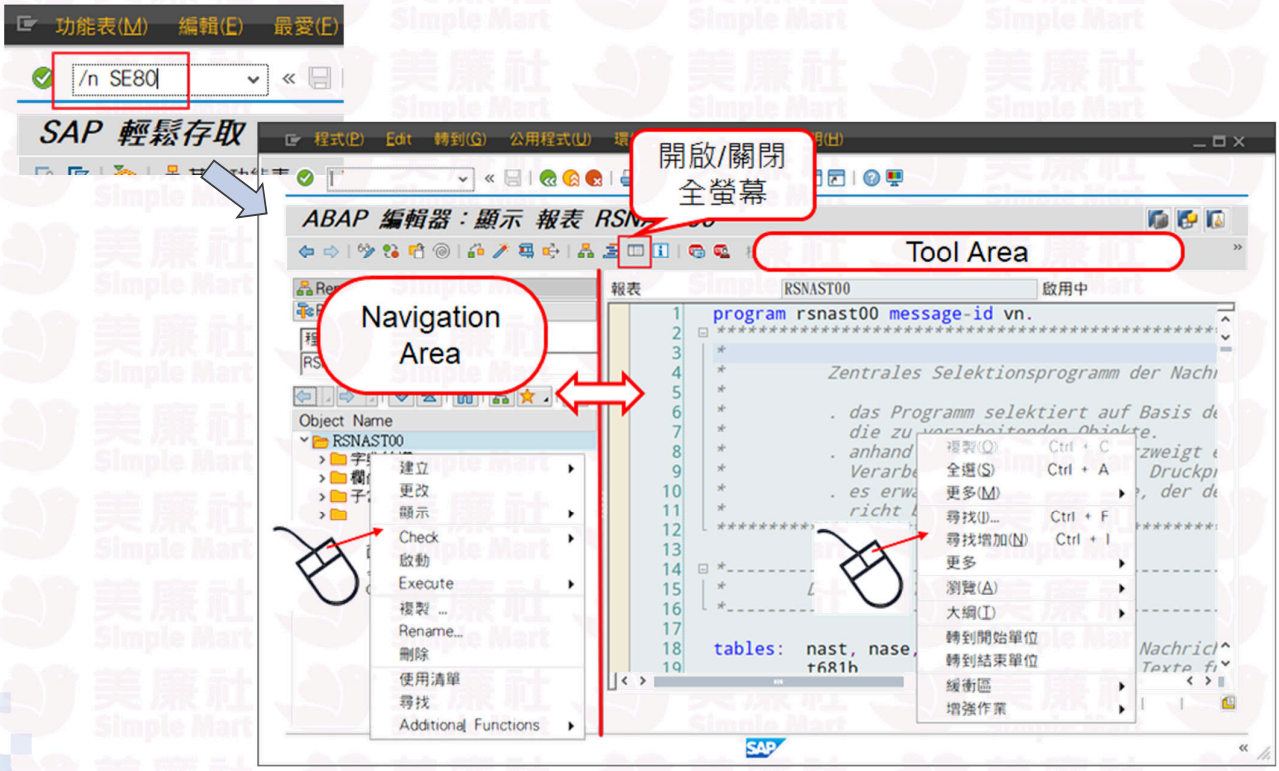
NTT DATA

ABAP Workbench Tools

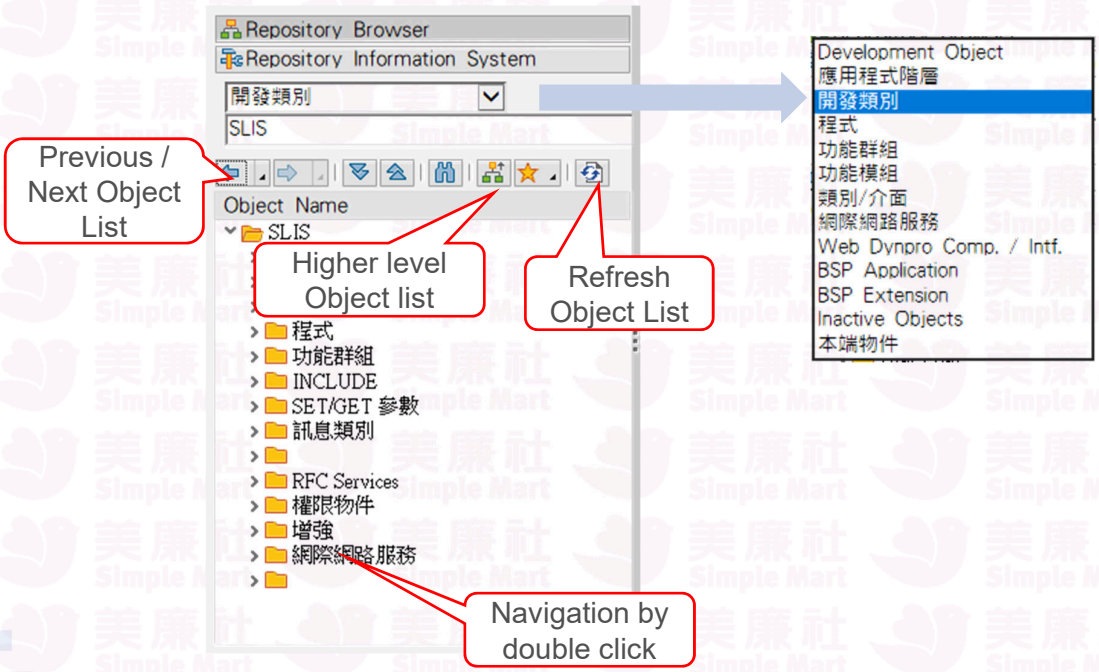
© 2018 NTT DATA Taiwan Co., Ltd.

NTT DATA

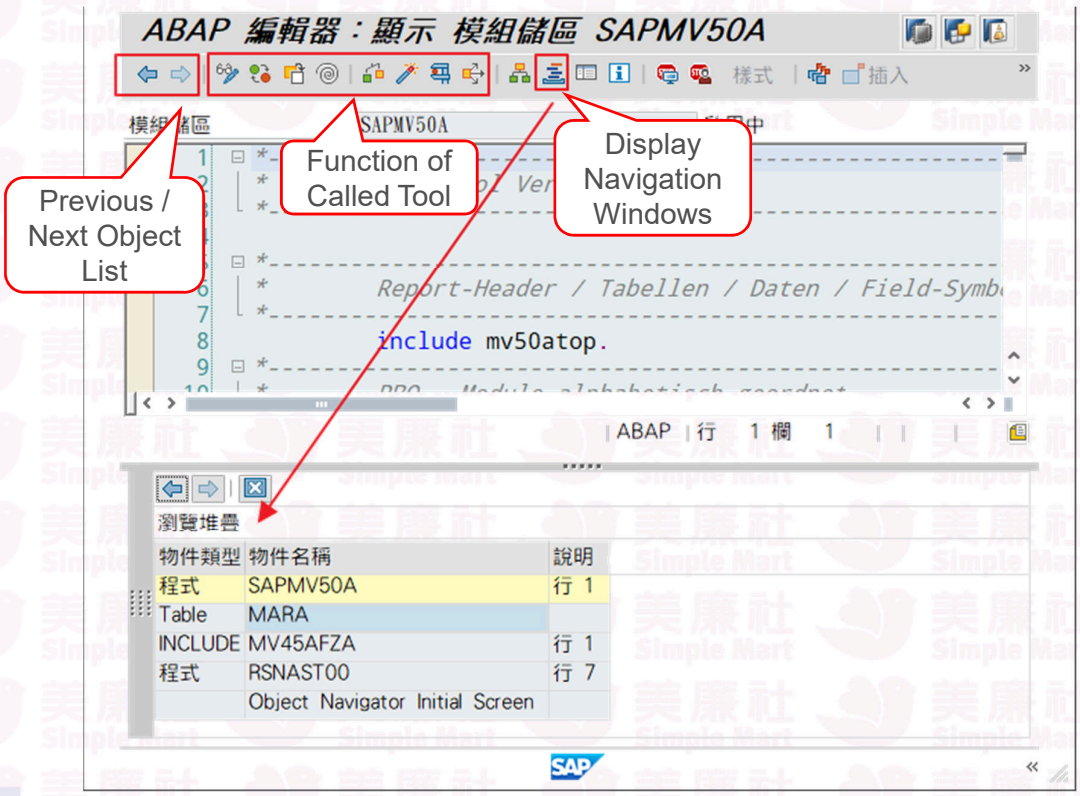
Screen Layout in the Object Navigator (SE80)



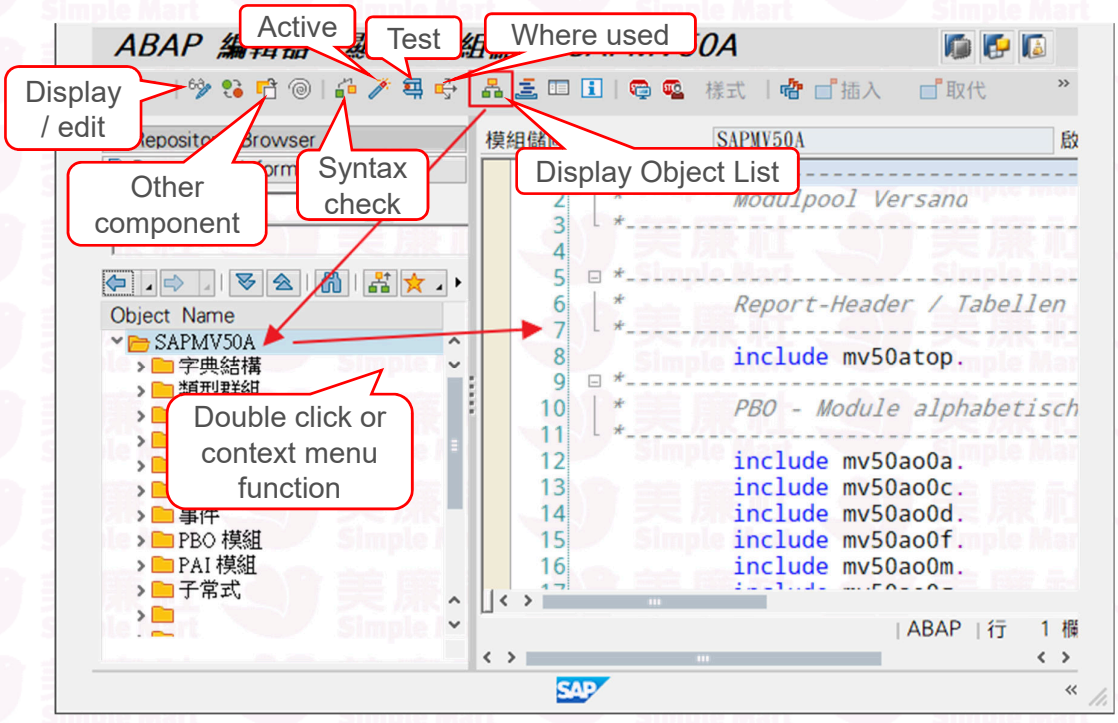
Working with the Navigation Area



Functions in the Tool Area



Synchronizing the Navigation and Tool Areas

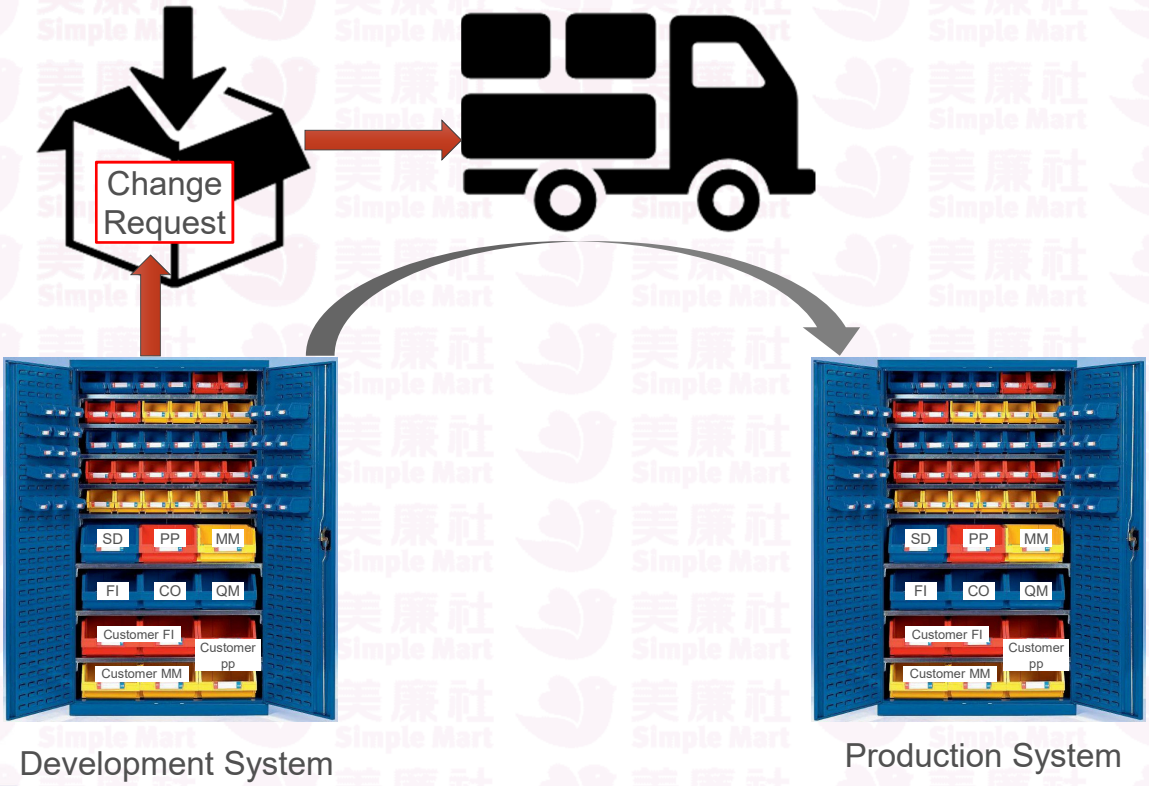


Introduction to the ABAP Workbench

- ✓Repository and Object Navigator
- ✓**Developing Programs and Organizing Developments**

© 2018 NTT DATA Taiwan Co., Ltd.

Transporting Development Objects

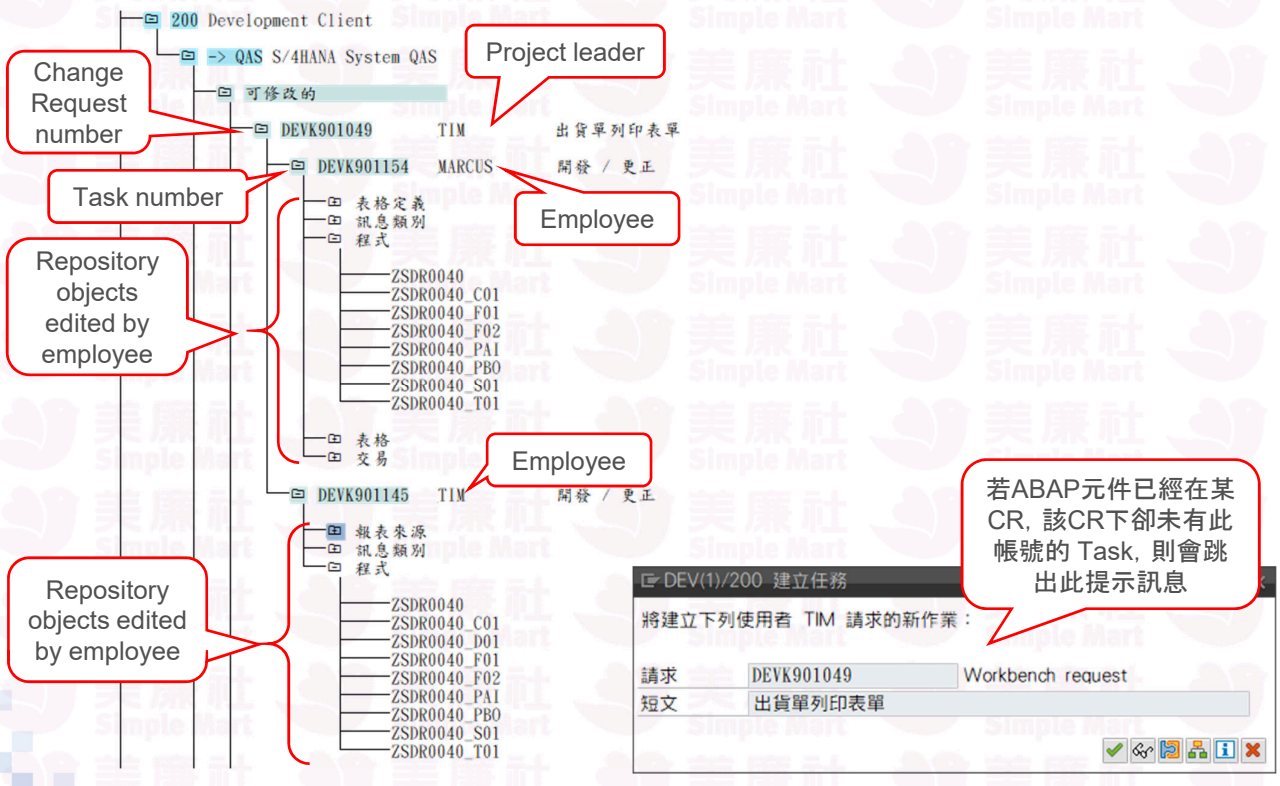


Development System

Production System

© 2018 NTT DATA Taiwan Co., Ltd.

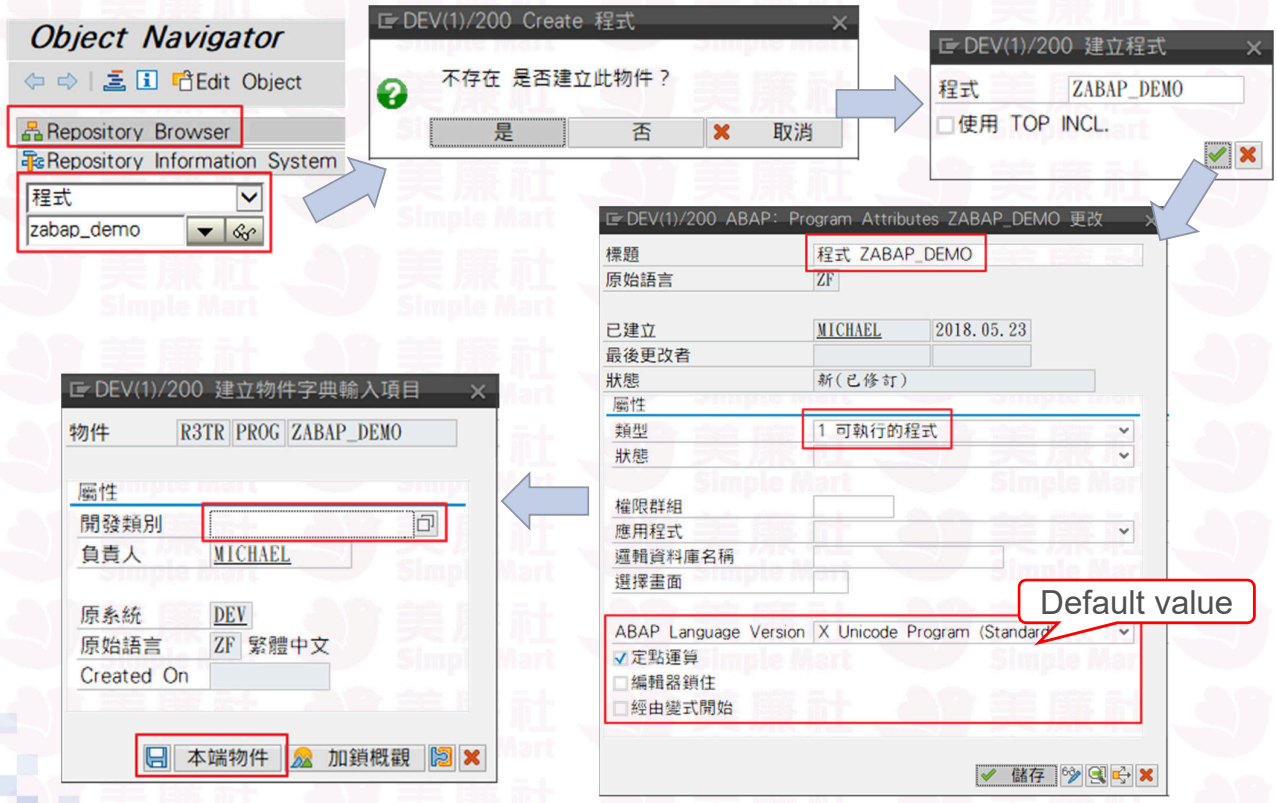
Organization of a Development Project in the Change Request(CR)



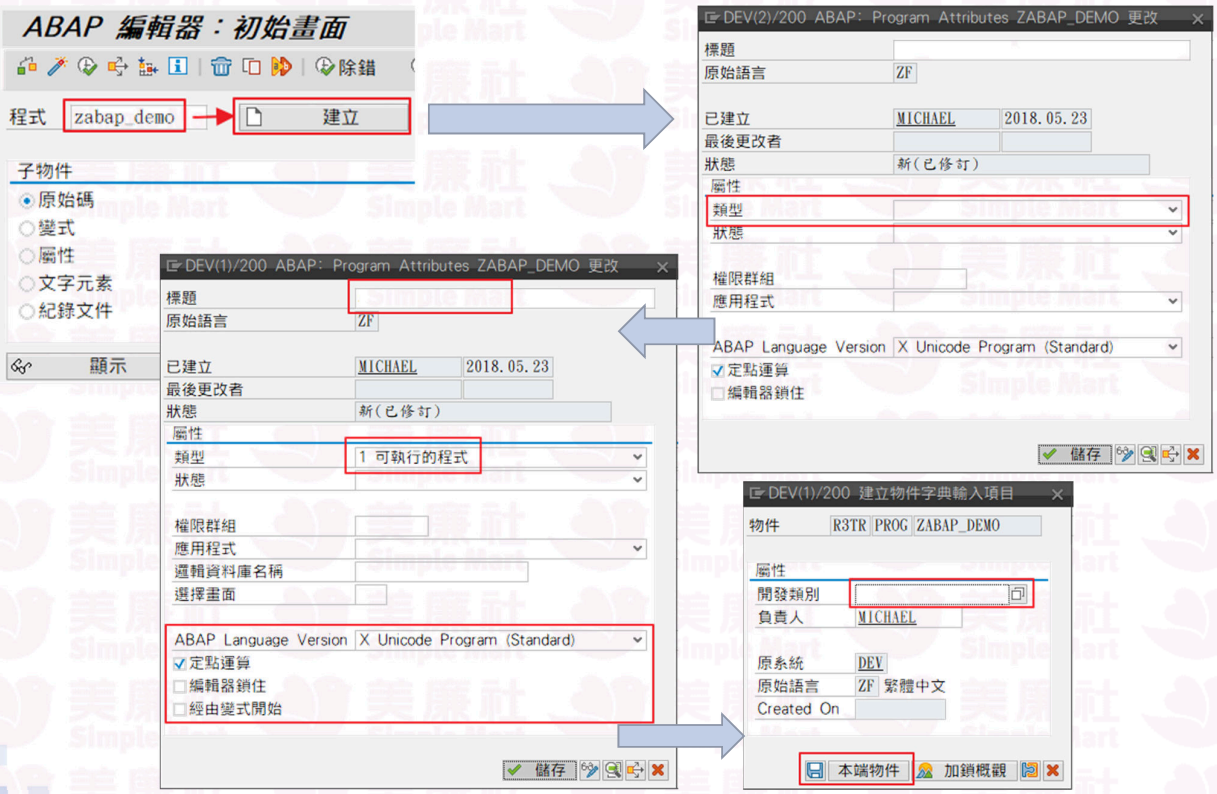
SAP Naming Rule

- SAP ABAP 客製元件，必須為 Y 或 Z 開頭，其餘字母保留給 SAP 使用。
- SAP 不希望客戶自行修改 SAP 元件。若有需要，可能需與 SAP 申請 access key，並附上理由。若因程式修改所造成的錯誤，SAP 不負責。

Creating an ABAP Program by SE80



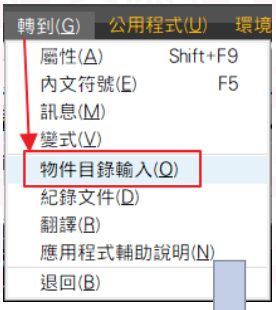
Creating an ABAP Program by SE38



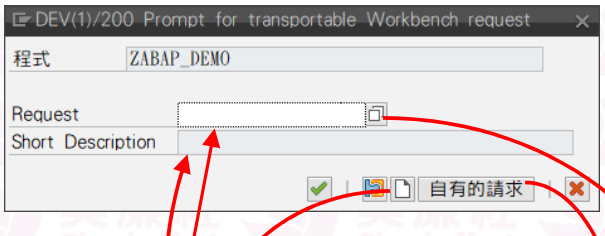
Compare “Local Object” and “Package”

- Local object(本端物件 / $TMP):
  - 僅能在建立時的 Server 環境測試用
  - 無法綁到 Change Request ，所以無法上傳至 QAS / PRD 環境
  - 可重新指定至特定的 package
- Package
  - 強制要綁入 Change Request
  - 可重新指定到另一個 package，但無法更改為 local object

- 重新指定 package

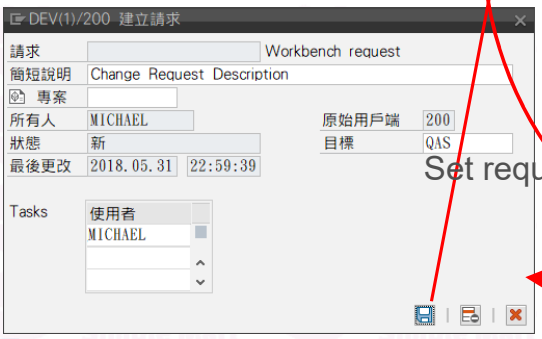


Assignment to a Change Request

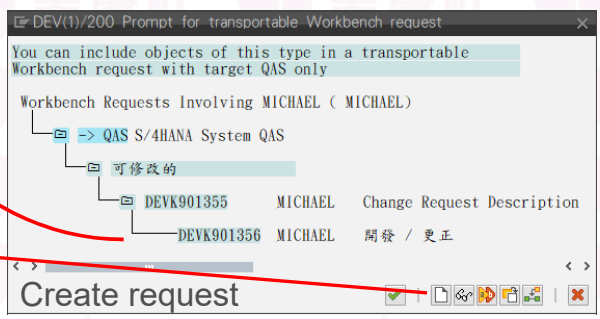


Create request

Find exist request



Set request



Create request

Introduction to the ABAP Programming Language

- The ABAP programming language…
  - Is typed
  - Enables multi-language applications
  - Enables SQL access
  - Has been enhanced as a object-oriented language
  - Is platform-independent
  - Is upward-compatible

© 2018 NTT DATA Taiwan Co., Ltd.

General ABAP Syntax I

General structure of an ABAP statement

XXX	YYY	.
ABAP keyword	Extras and operands (keyword-specific)	Period as end of the statement

```
DATA wa_scarr TYPE scarr.

SELECT * INTO @wa_scarr
         FROM scarr.
  NEW-LINE.
  WRITE wa_scarr-carrid.
  WRITE wa_scarr-carrname.
  WRITE wa_scarr-url.
ENDSELECT.
```

不區分大小寫

© 2018 NTT DATA Taiwan Co., Ltd.

General ABAP Syntax II

```
* 程式註解1：整行註解
DATA wa_scarr TYPE scarr.

SELECT * INTO @wa_scarr "程式註解2
         FROM scarr.
  NEW-LINE.
  WRITE: wa_scarr-carrid, "Chained Statement
         wa_scarr-carrname,
         wa_scarr-url.
ENDSELECT.
```

© 2018 NTT DATA Taiwan Co., Ltd.

Return Code of the ABAP Statements

```
PARAMETERS p_carrid TYPE scarr-carrid.

DATA wa_scarr TYPE scarr.

SELECT SINGLE * INTO @wa_scarr
  FROM scarr
  WHERE carrid = @p_carrid.
IF sy-subrc = 0.
  WRITE: wa_scarr-carrid,
         wa_scarr-carrname,
         wa_scarr-url.
ELSE.
  WRITE 'Sorry, record not found'.
ENDIF.
```

© 2018 NTT DATA Taiwan Co., Ltd.

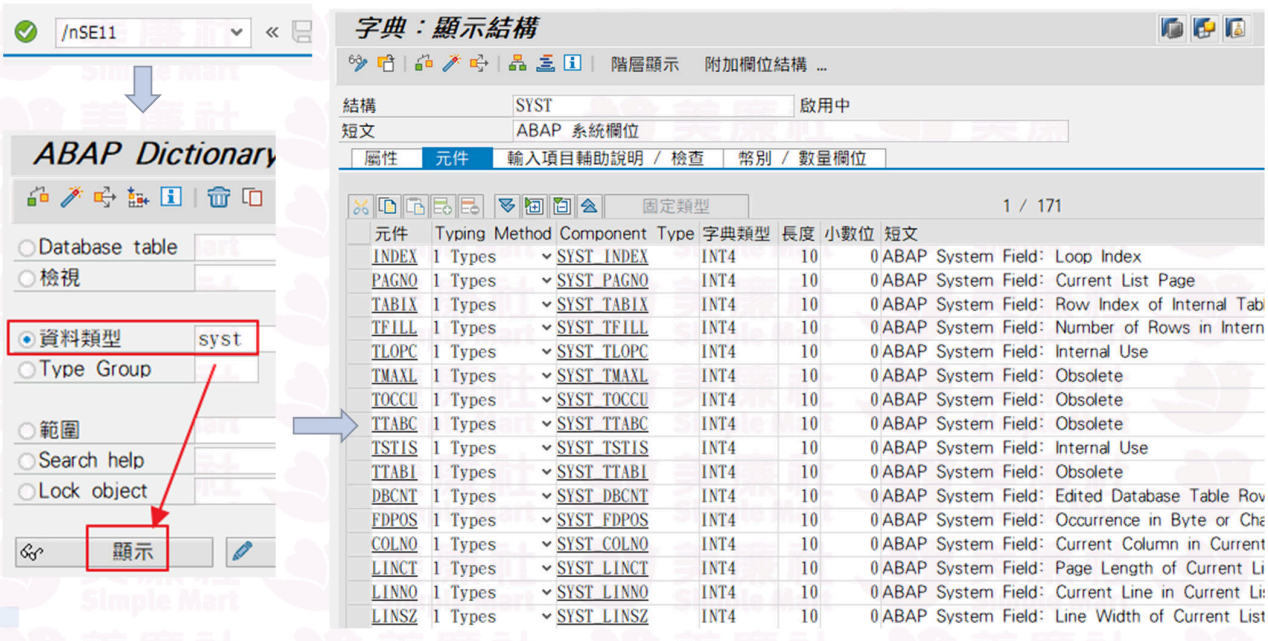
Some interesting System Fields

System field	Meaning
SY-MANDT	Logon client
SY-UNAME	Logon name of the user
SY-LANGU	Logon language of the user
SY-DATUM	Server date
SY-UZEIT	Server time
SY-DATLO	Local date
SY-TIMLO	Local time
SY-ZONLO	Local Time Zone
SY-TCODE	Current t-code
SY-REPID	Name of the current ABAP program (=> SY-CPROG)
SY-INDEX	Loop counter at DO and WHILE loops
...	...

© 2018 NTT DATA Taiwan Co., Ltd.

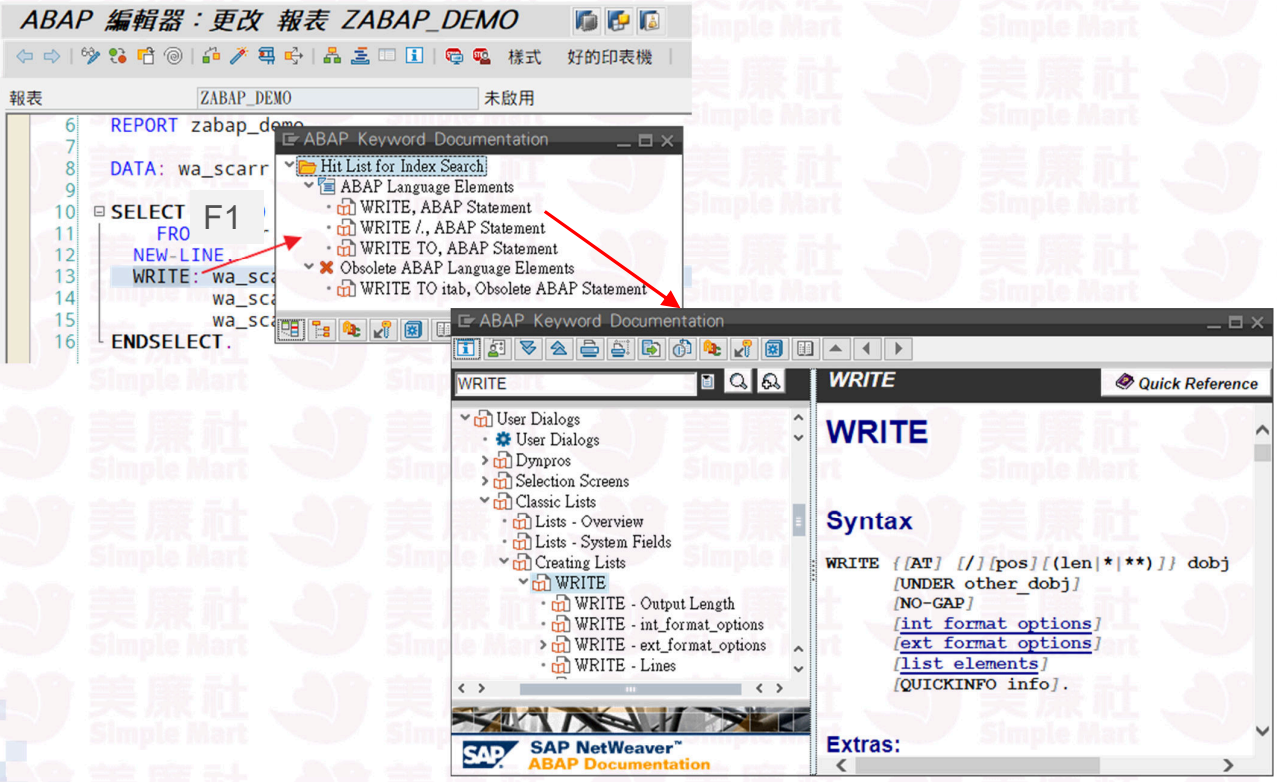
System Fields in Dictionary

- 唯一有“簡稱”的變數
  - 本名：SYST；程式可縮寫為 SY



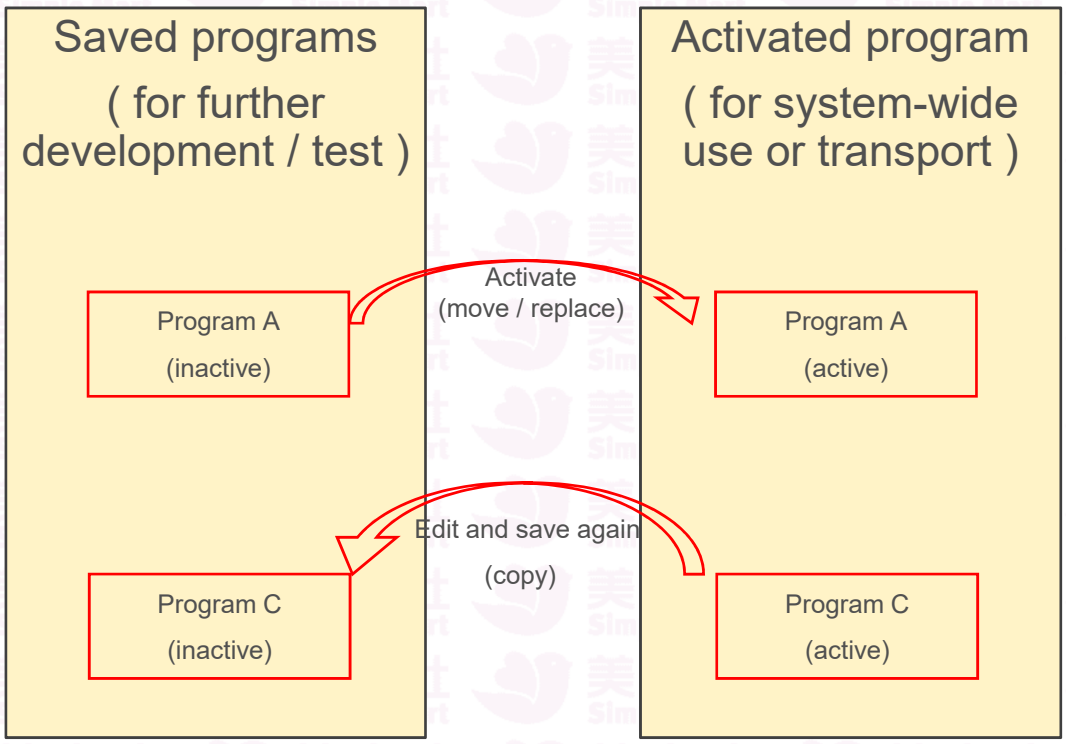
© 2018 NTT DATA Taiwan Co., Ltd.

Keyword Documentation in the ABAP Editor (SE38 / SE80)

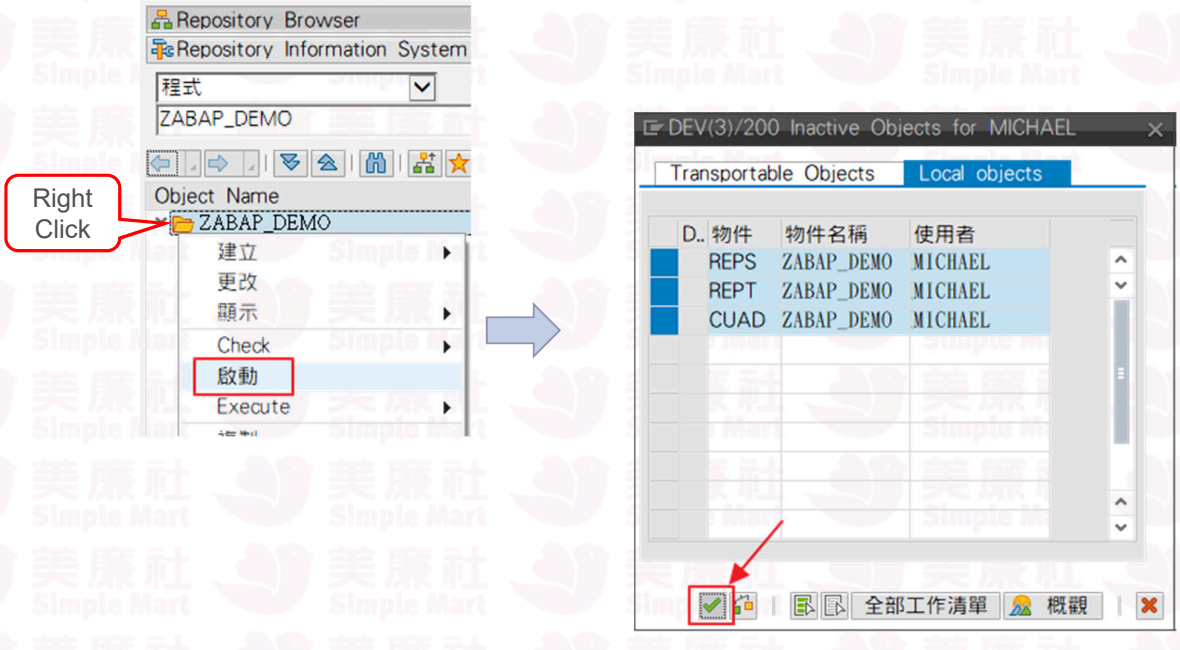


ABAP Workbench Setting – ABAP Editor

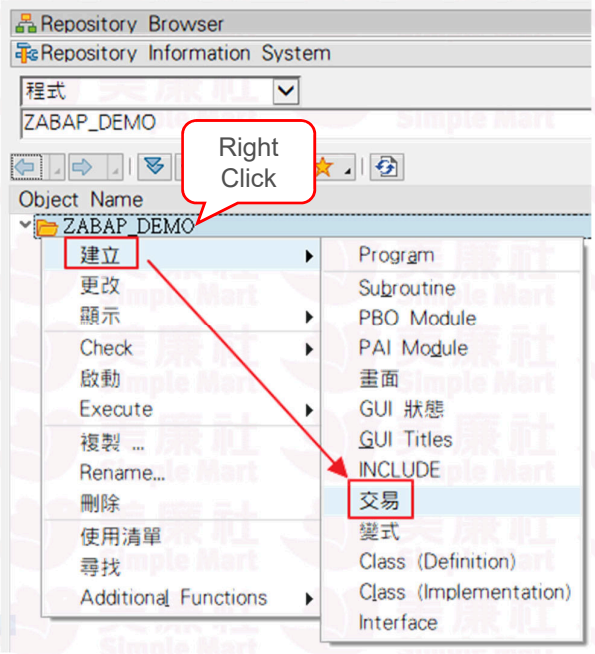
Inactive and Active Development Objects



Active ABAP Report



Creating a Transaction (Step 1)



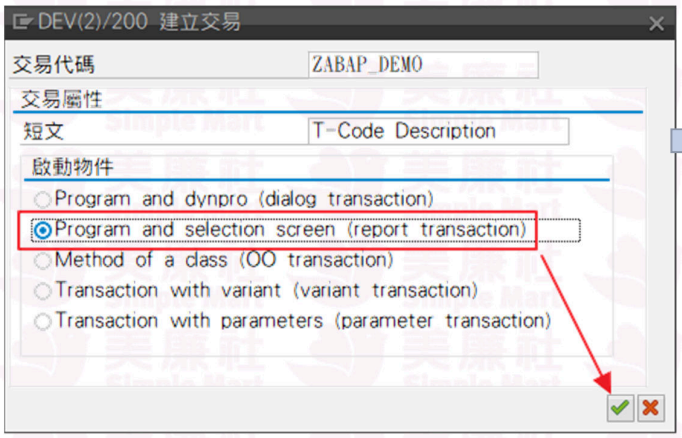
or

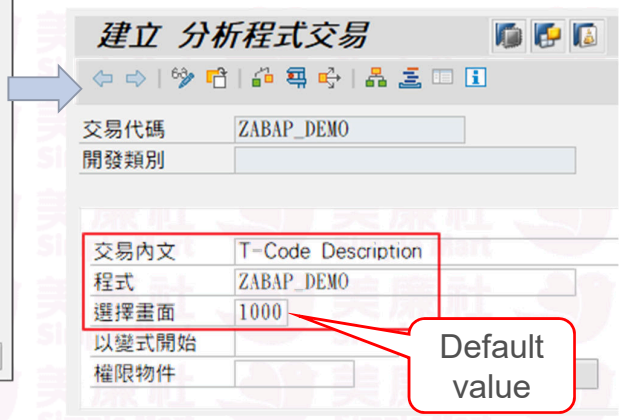
T-code:SE93

© 2018 NTT DATA Taiwan Co., Ltd.

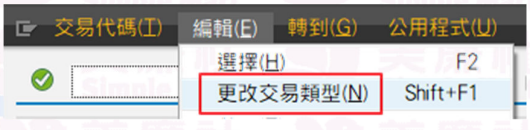
NTT DATA

Creating a Transaction (Step 2)





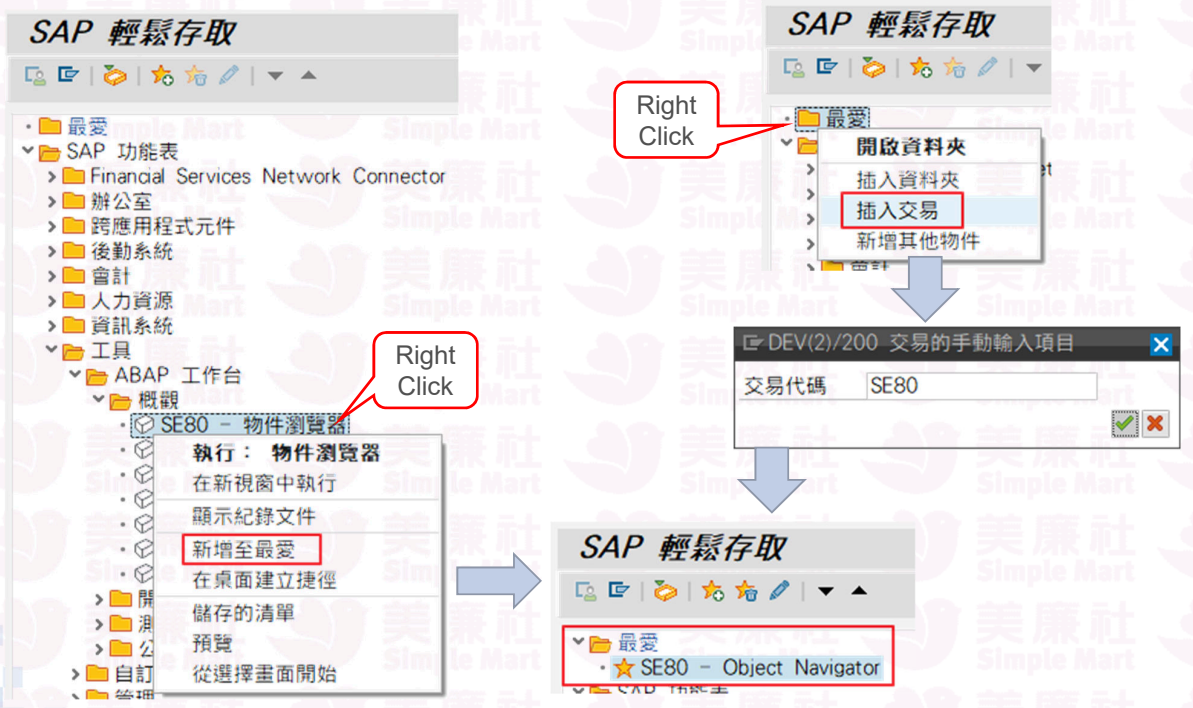
If transaction type is error



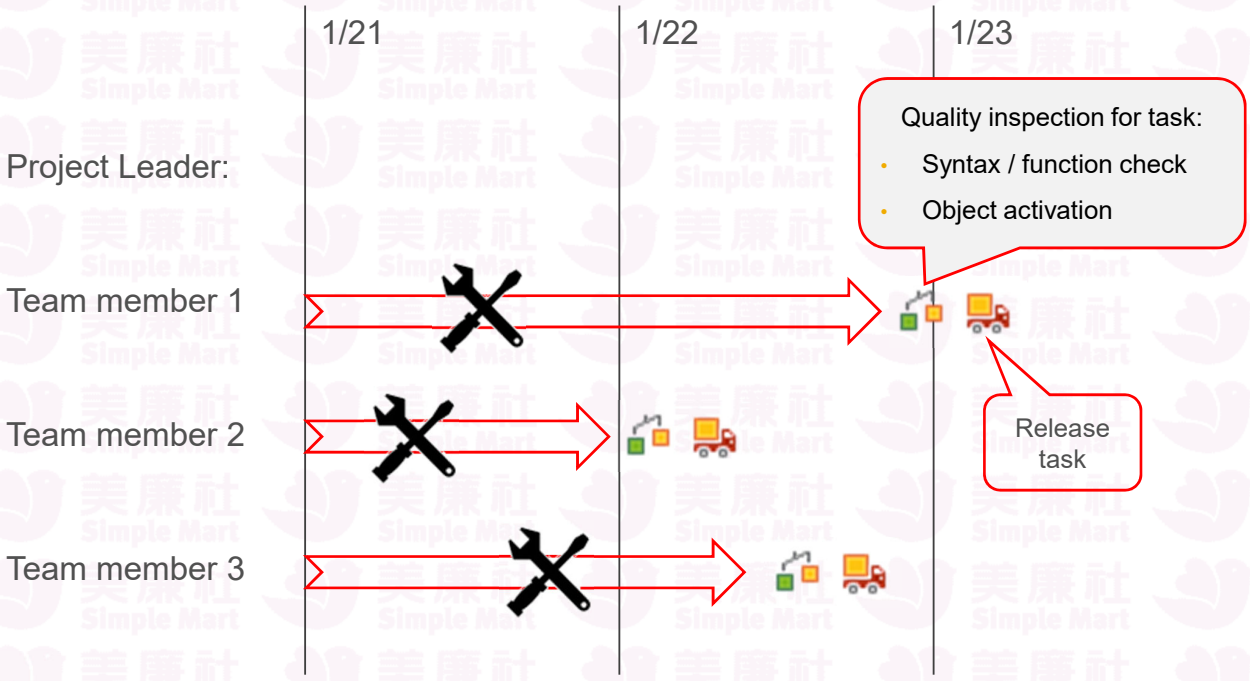
© 2018 NTT DATA Taiwan Co., Ltd.

NTT DATA

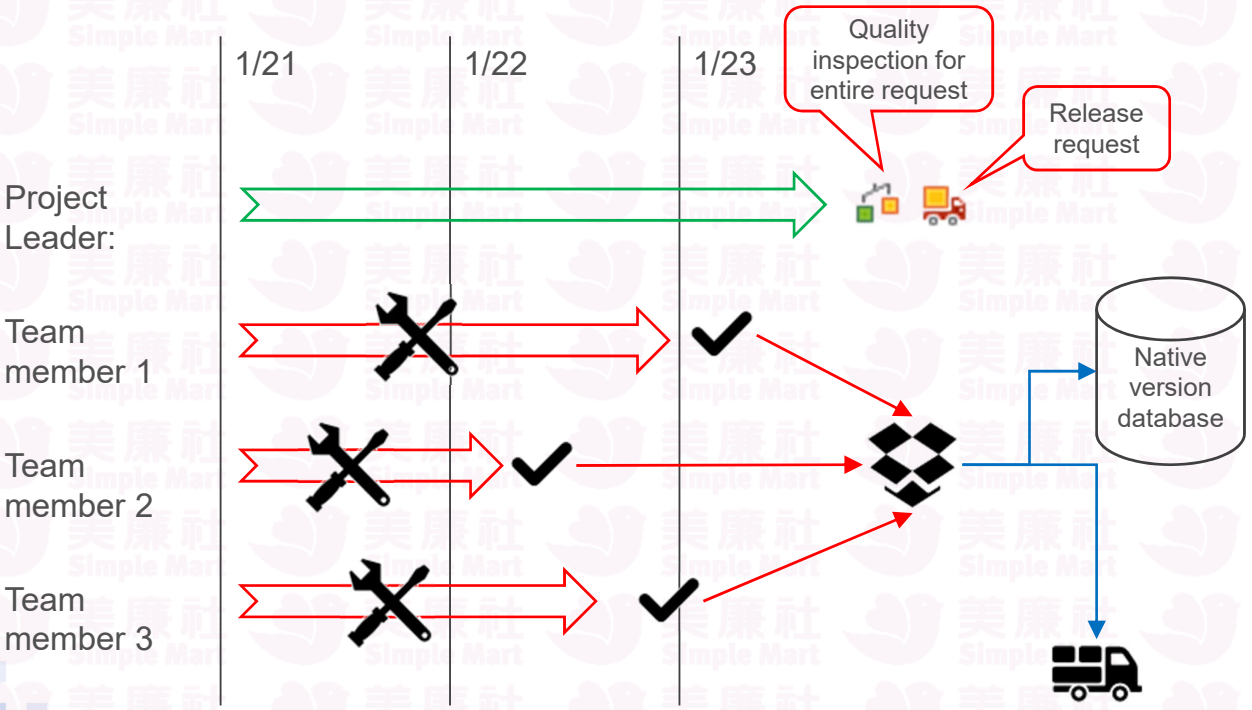
Inserting a T-Code into the Personal Favorites



The Developer Releases his Task



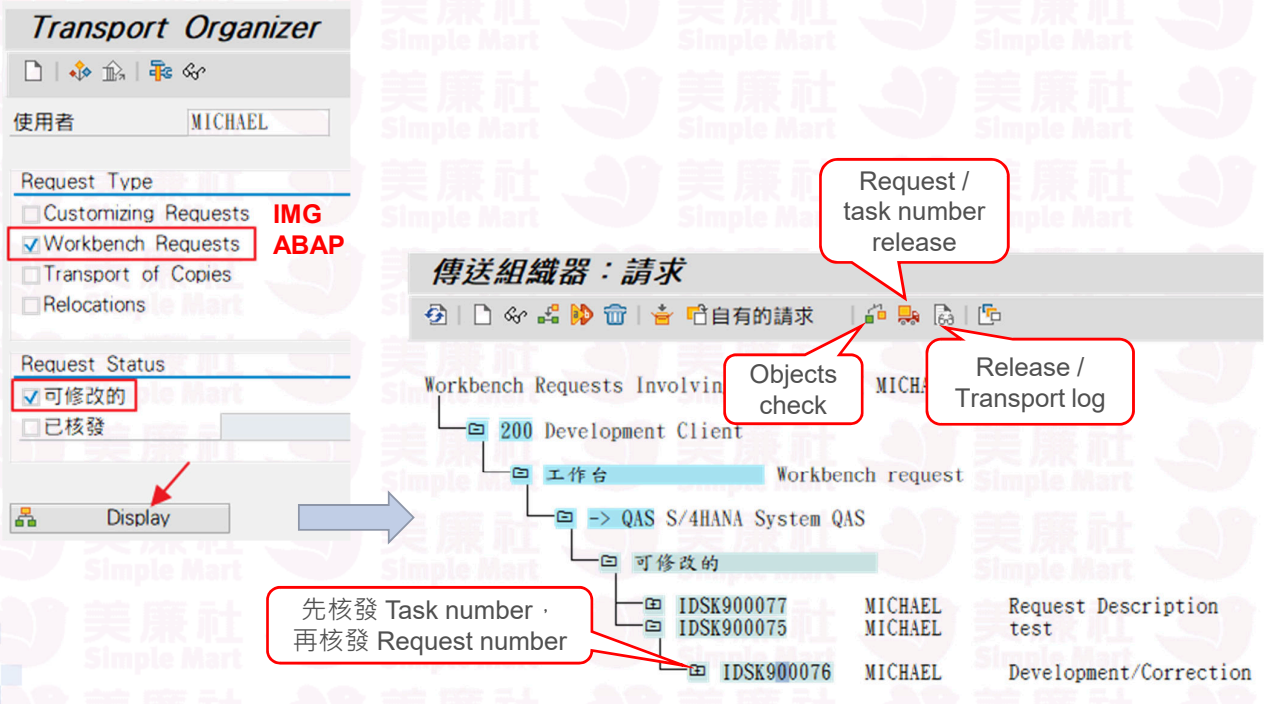
Closing the Project by Releasing the Request



© 2018 NTT DATA Taiwan Co., Ltd.

NTT DATA

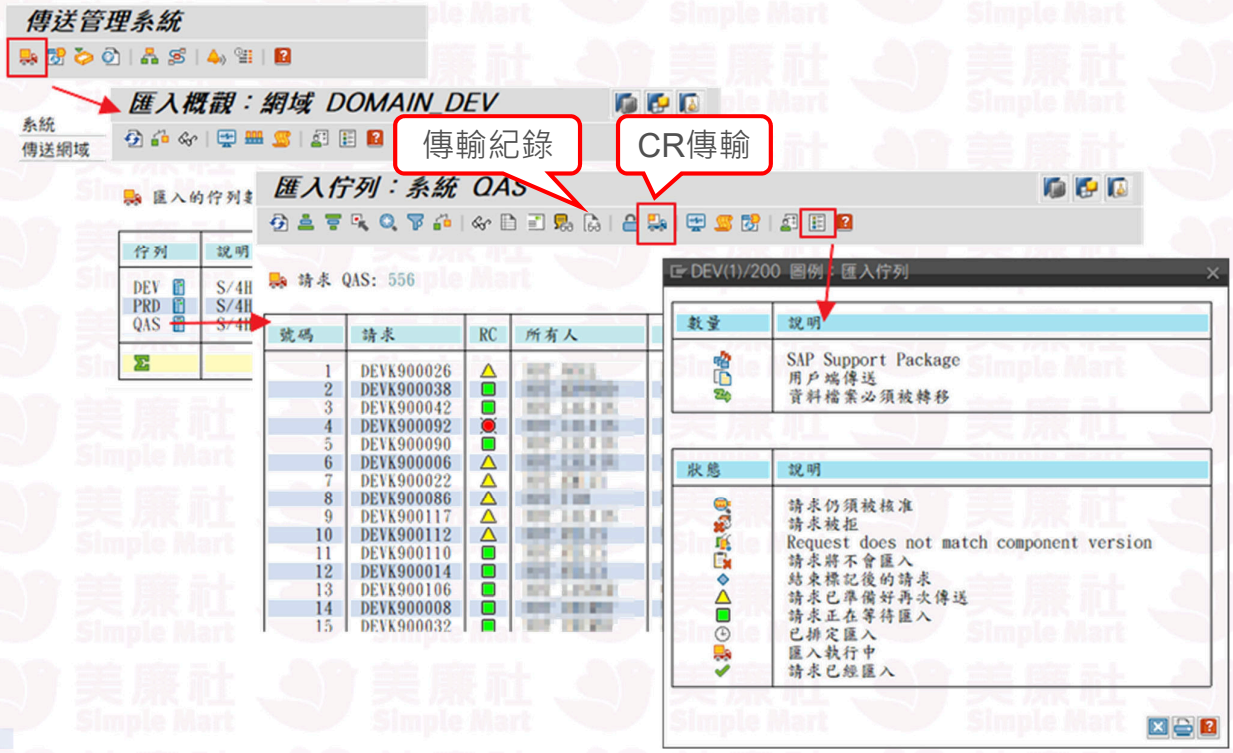
T-Code:SE09/SE10
Release CR



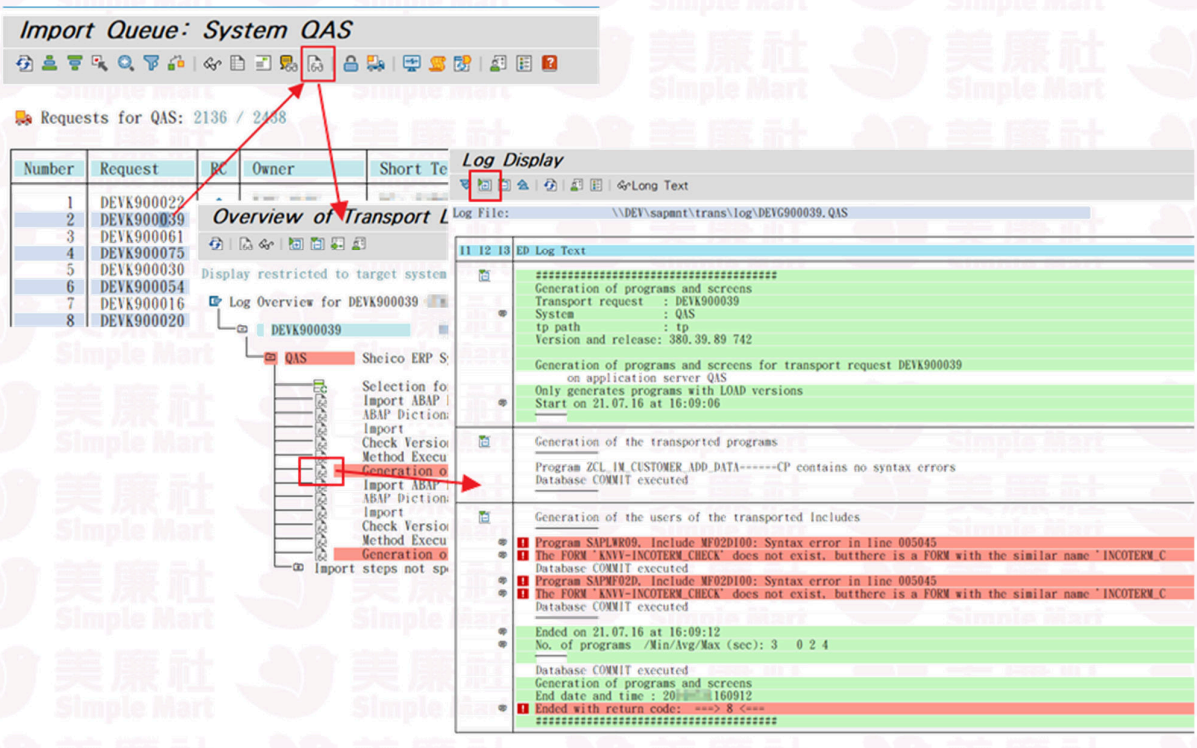
© 2018 NTT DATA Taiwan Co., Ltd.

NTT DATA

T-code : STMS
Transport CR to QAS/PRD



CR Transport Log in SAP GUI



CR Transport Sequence - 1

DEV CR Version	ABAP Component Version				
DEVK900001	A1	B1	C1		
DEVK900003	A2	B2			
DEVK900005	A3		C2	D1	
DEVK900007			C3		E1

DEV Final version: A3, B2, C3, D1, E1

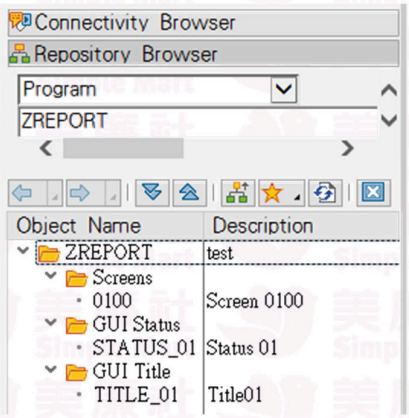
QAS Final version: A3, B2, C3, D1, E1

PRD Final version: ??

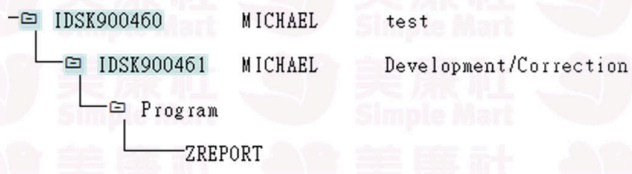
CR Transport Sequence - 2

- Re-package CR?

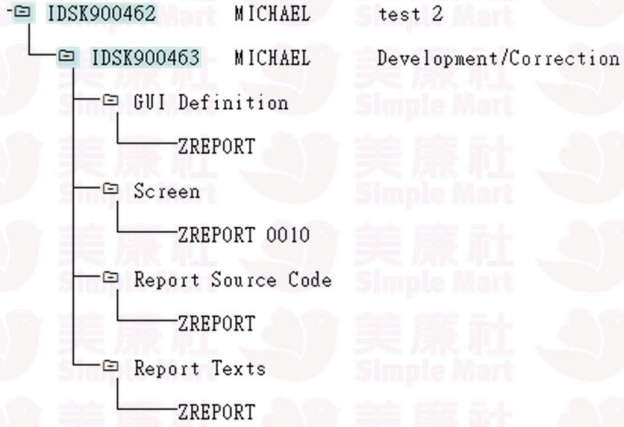
程式元件架構



Ex: 第一版的 CR 內容

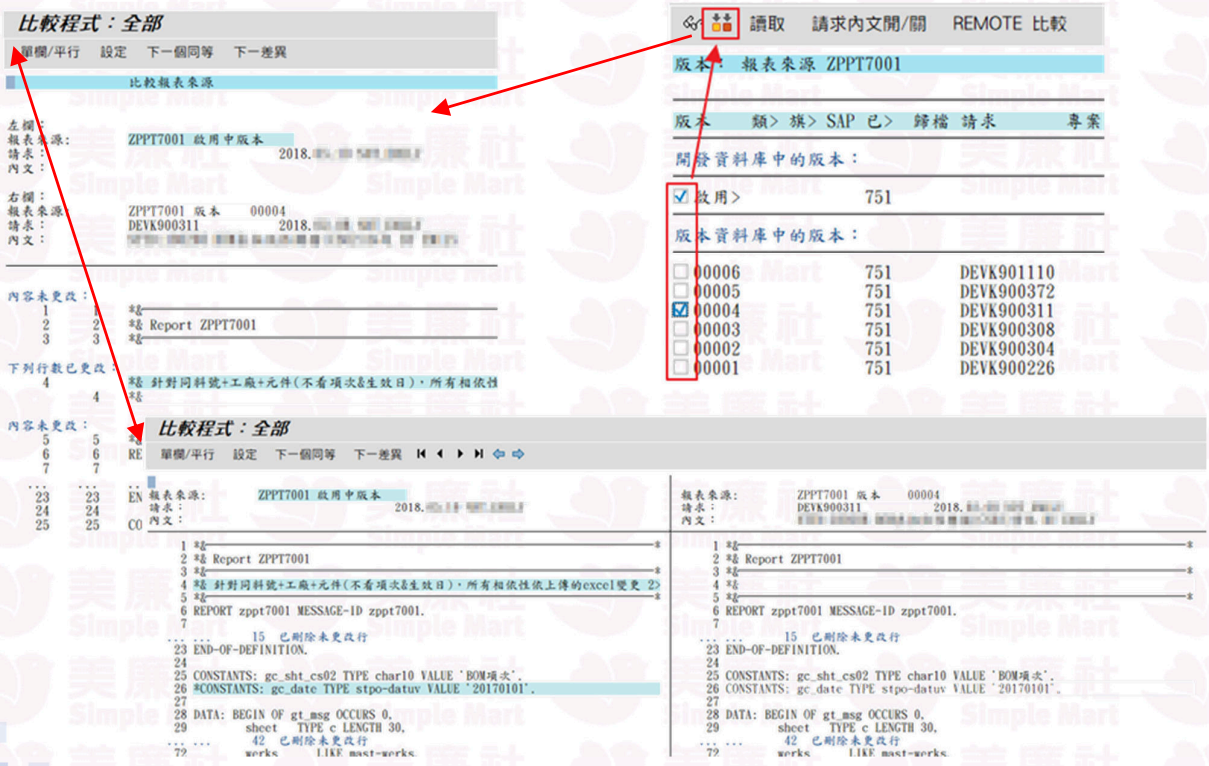


Ex: 若需要重包CR，則需...



Version Compare in SAP GUI - 1

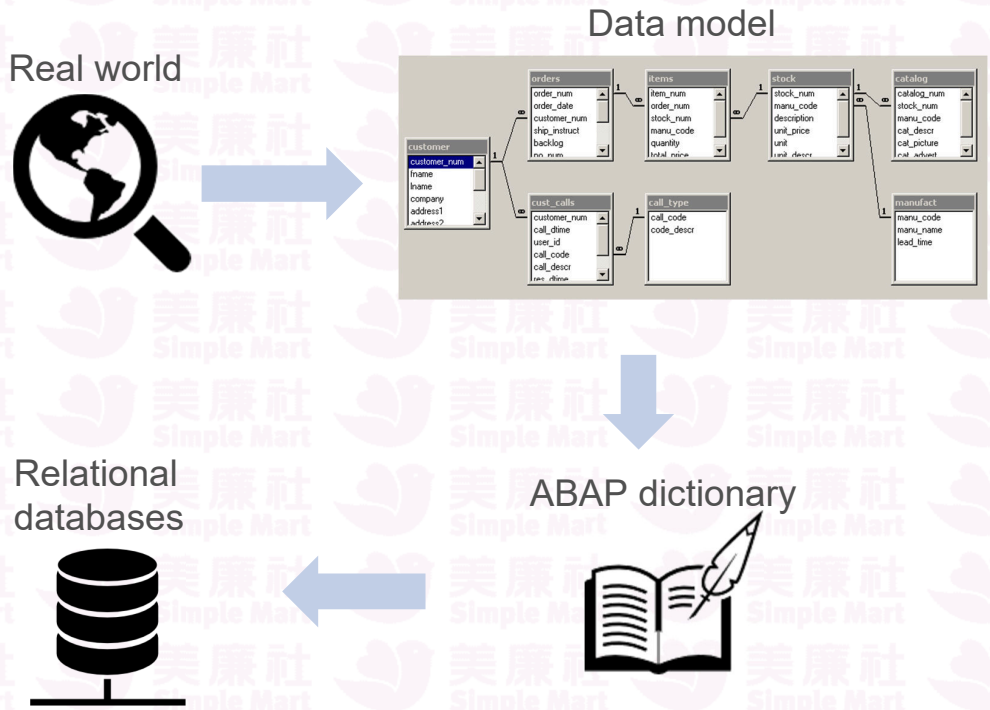
Version Compare in SAP GUI - 2



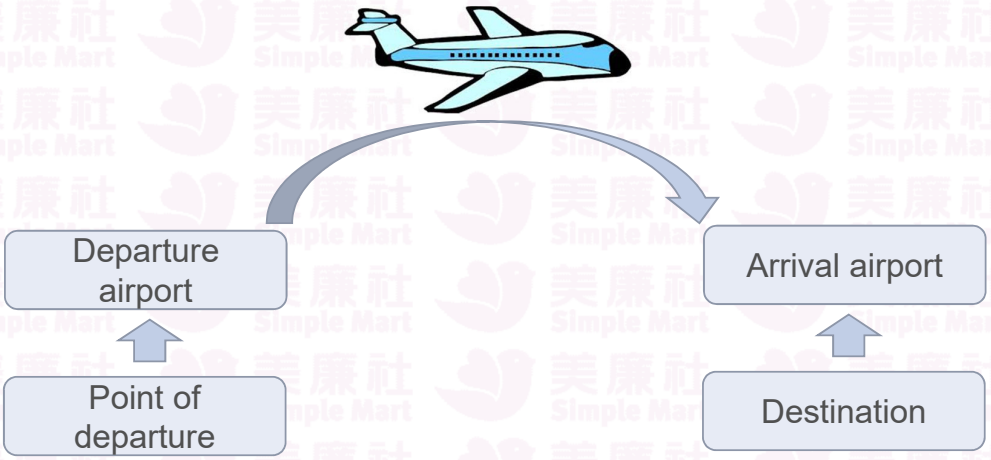
Introduction to the ABAP Dictionary

- Data Modeling and Descriptive Elements in the ABAP Dictionary

Data Modeling



Flight Data Model for ABAP Training Courses

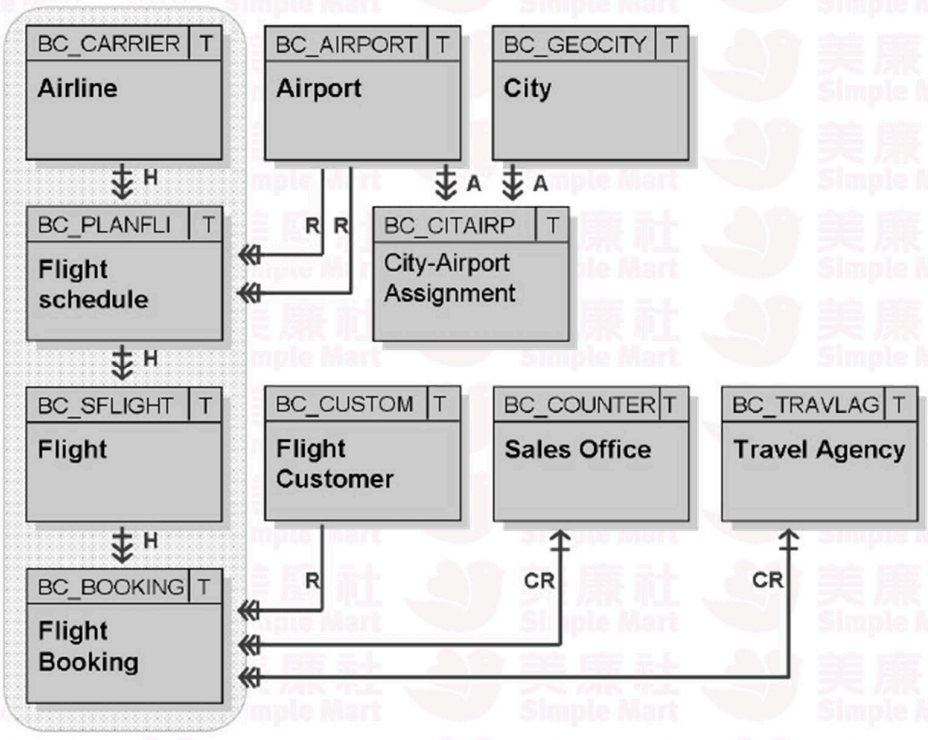


- Typical inquires at the travel agency:
  - Suitable airports
  - Suitable flight connections
  - Suitable flight times
  - Further information about the flights: price, utilization of capacity or availability, and so on.

© 2018 NTT DATA Taiwan Co., Ltd.

NTT DATA

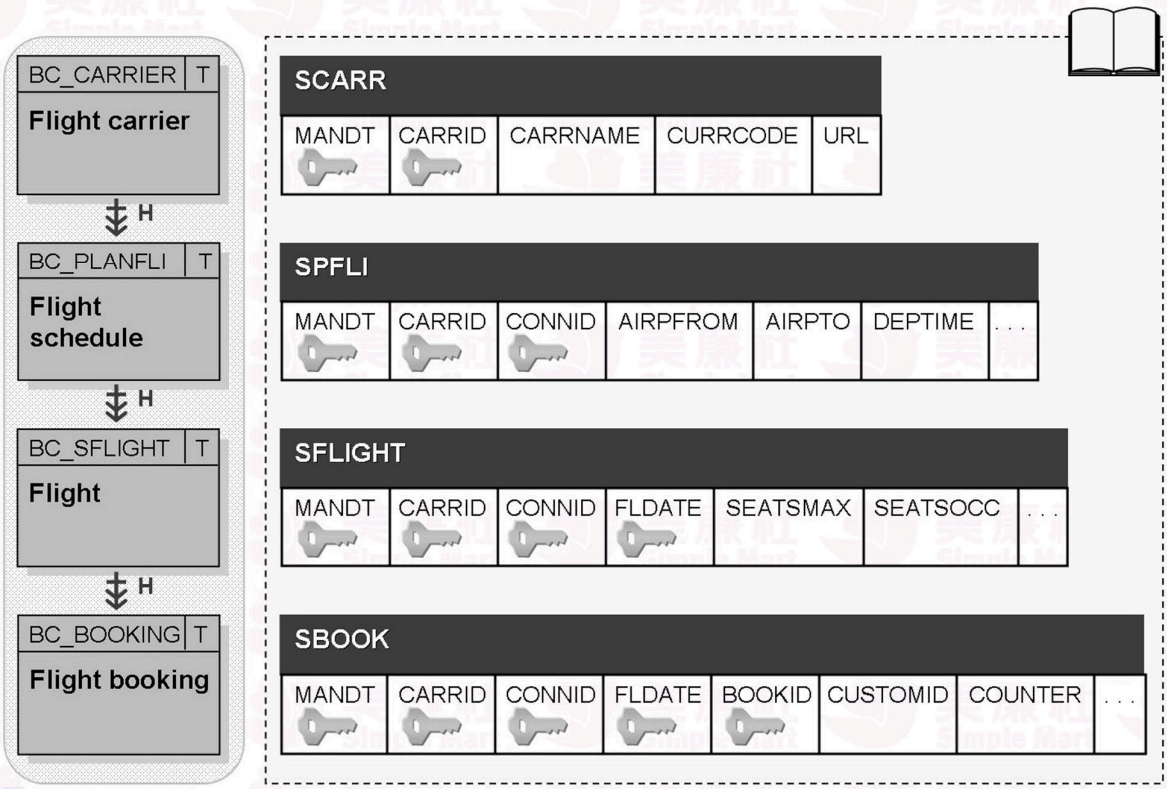
Relational data Model



© 2018 NTT DATA Taiwan Co., Ltd.

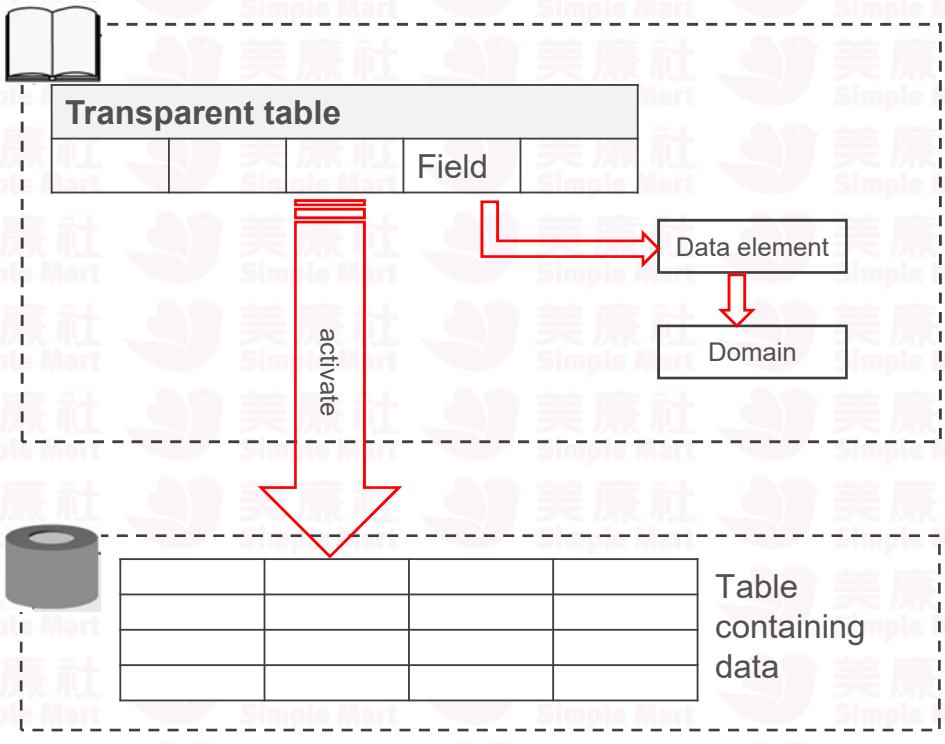
NTT DATA

Implementation Using Transparent Tables



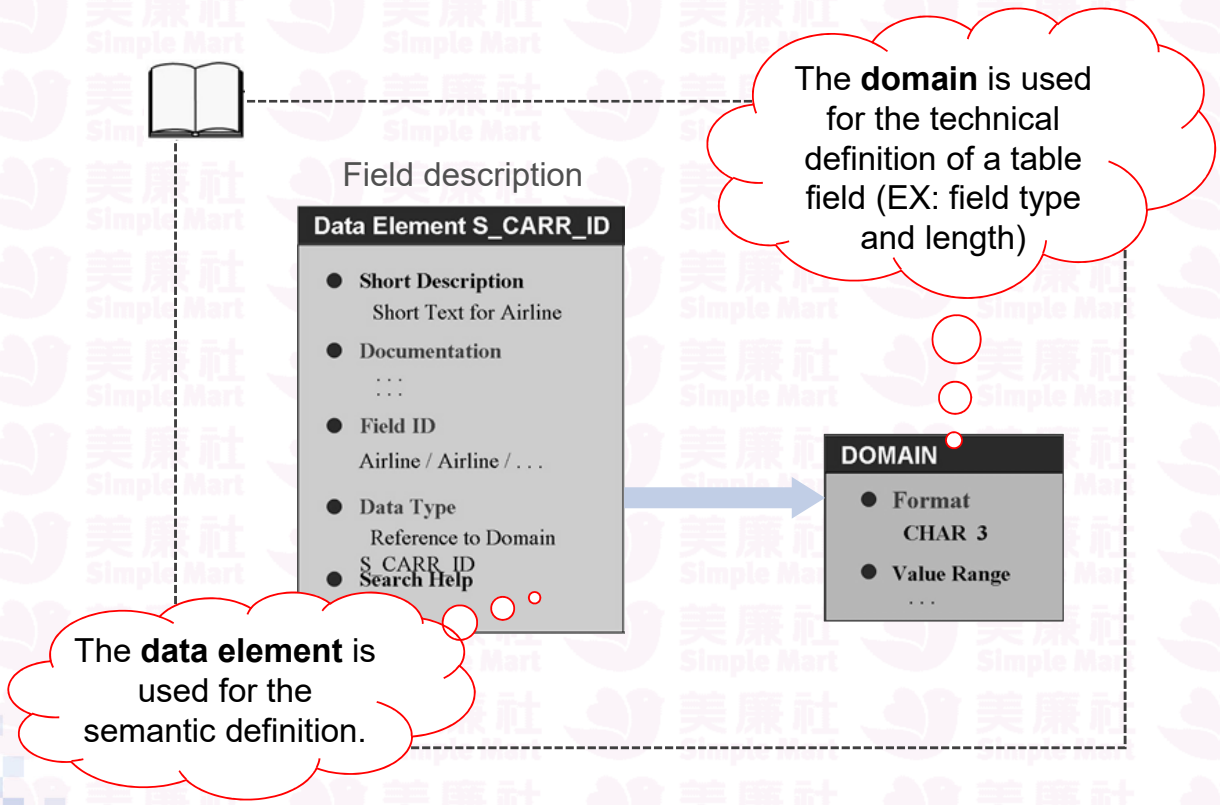
© 2018 NTT DATA Taiwan Co., Ltd.

Technical Structure of a Transparent Table

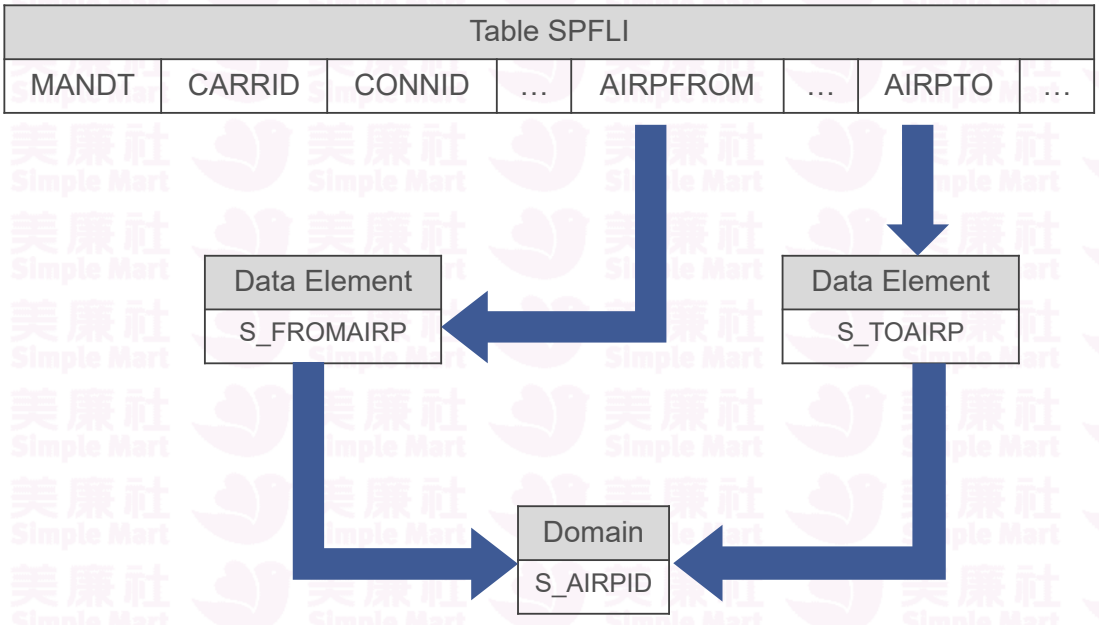


© 2018 NTT DATA Taiwan Co., Ltd.

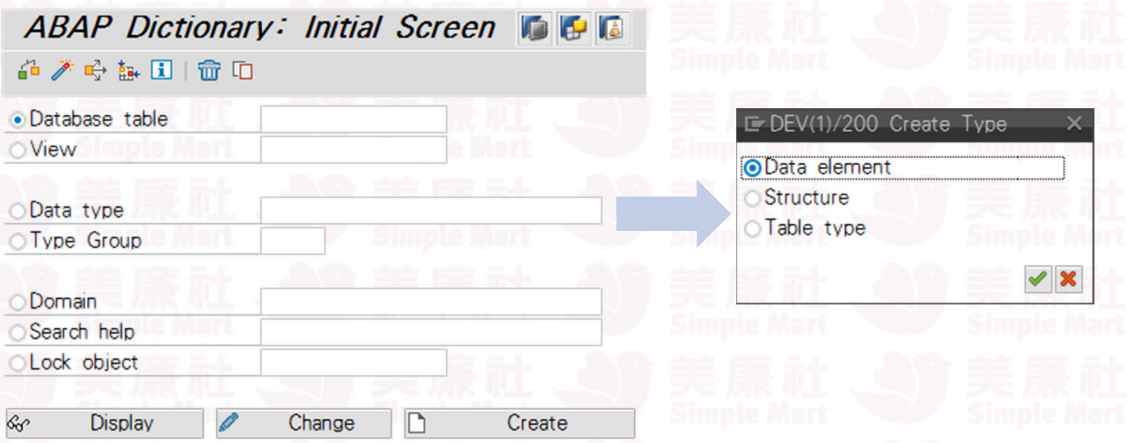
Data Element and Domain



Two-Level Domain Concept: Example



T-Code : SE11
Create ABAP Dictionary Component



Transparent Tables in the ABAP Dictionary (SE11)

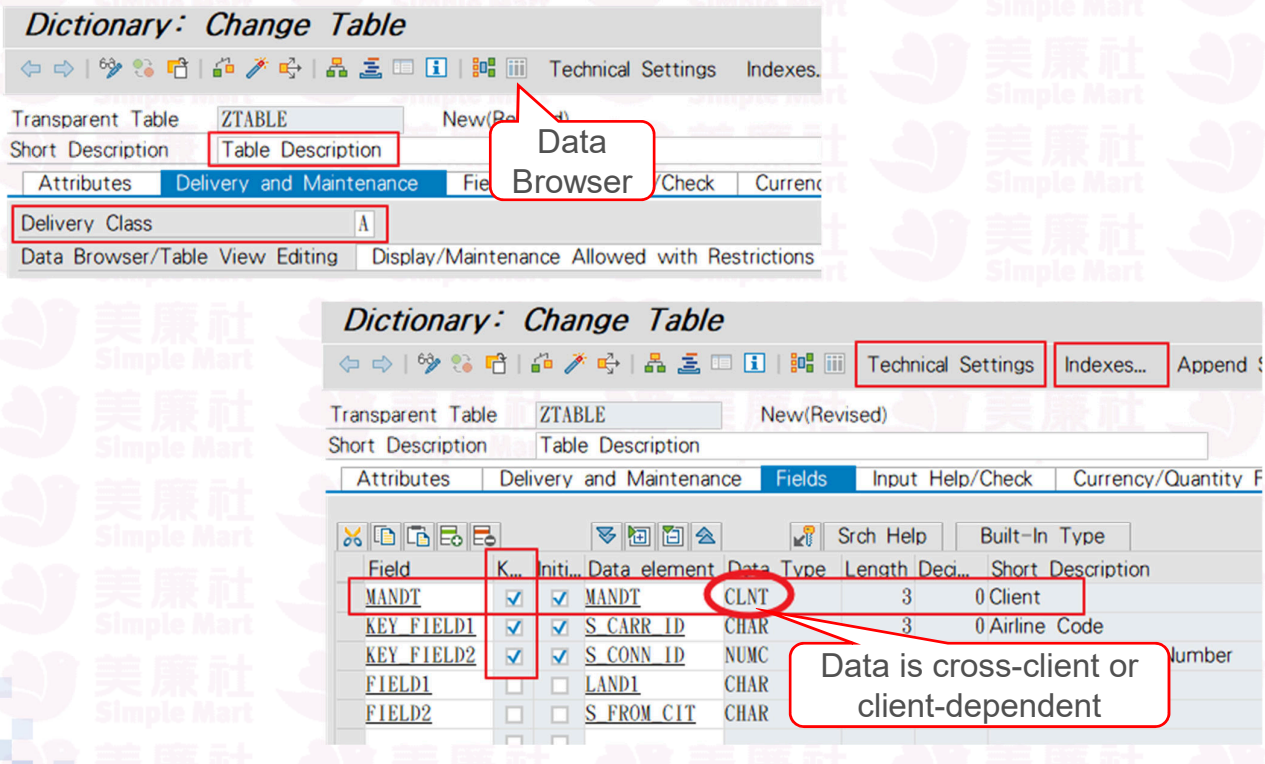
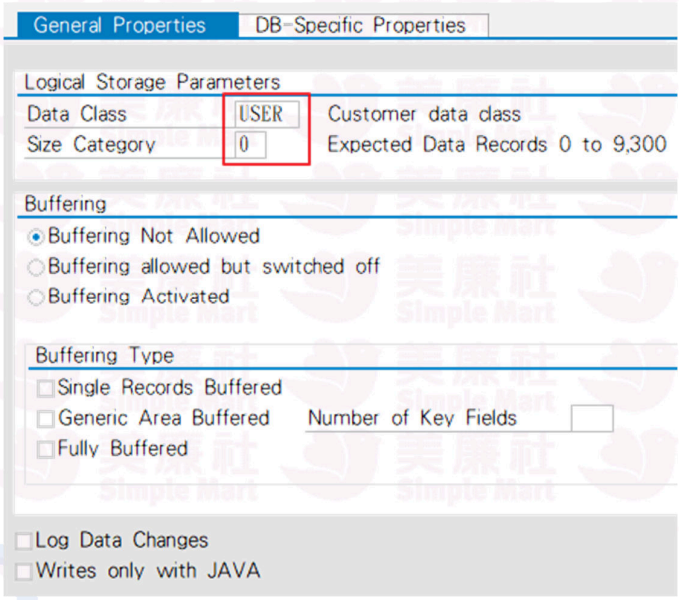
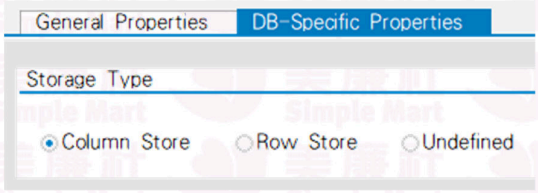
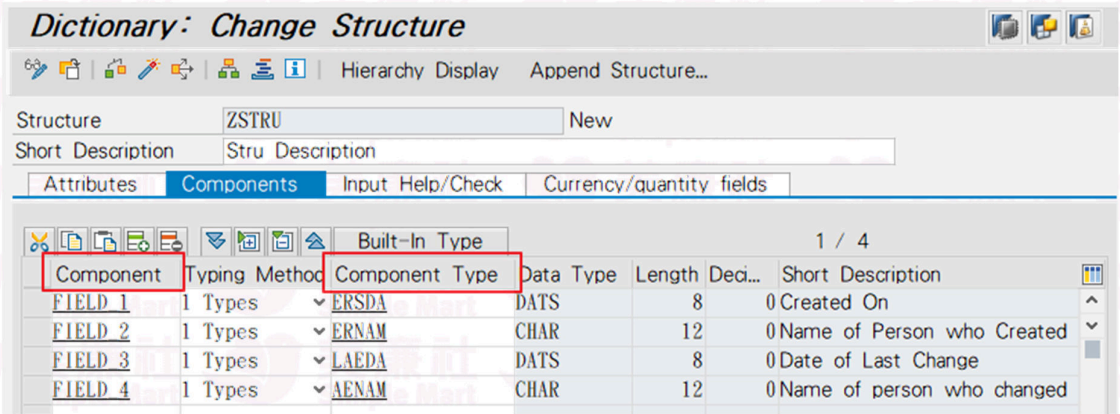


Table – Technical Settings

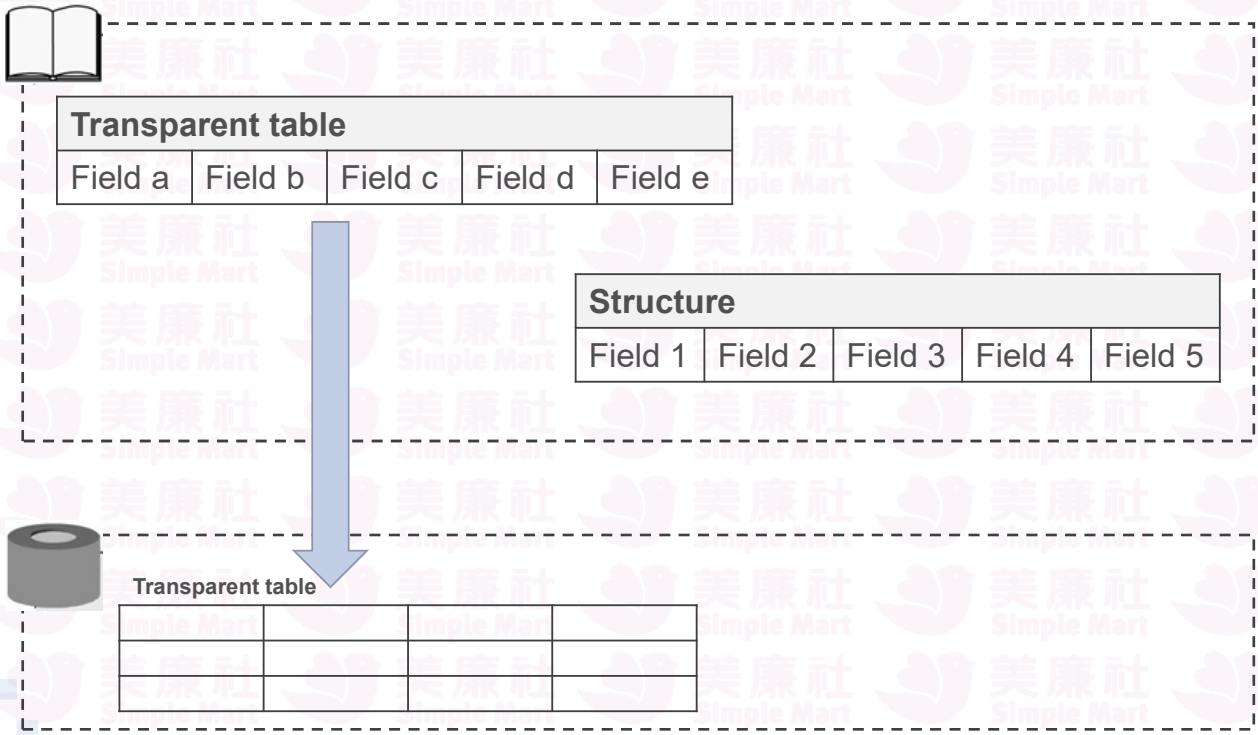




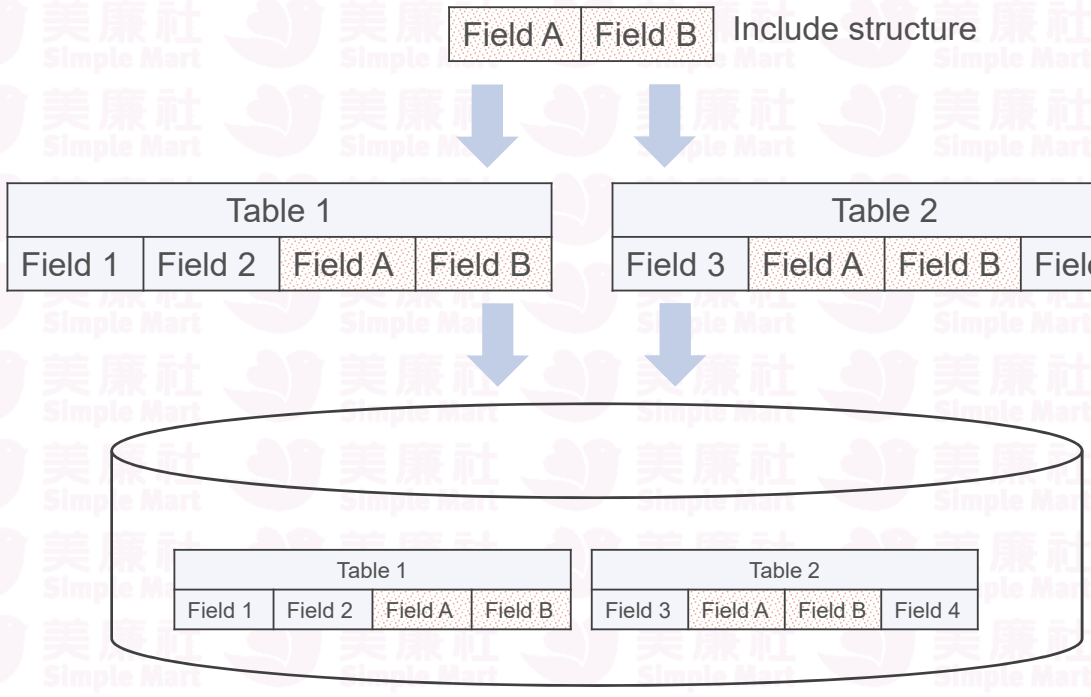
Structures in the ABAP Dictionary (SE11)



Structure & Table in ABAP Dictionary



ABAP Dictionary – include structure



ABAP Dictionary – include structure

Transparent Table ZTABLE New
Short Description Table Description

Attributes | Delivery and Maintenance | Fields | Input Help/Check | Currency/Quantity Fields

Srch Help | Built-In Type | 1 / 10

Field	K...	Initi...	Data element	Data...	Length	Deci...	Short Description
MANDT	☑	☑	MANDT	CLNT	3	0	Client
KEY_FIELD1	☑	☑	S_CARR_ID	CHAR	3	0	Airline Code
KEY_FIELD2	☑	☑	S_CONN_ID	NUMC	4	0	Flight Connection Number
FIELD1	☐	☐	LAND1	CHAR	3	0	Country Key
FIELD2	☐	☐	S_FROM_CIT	CHAR	20	0	Departure city
.INCLUDE	☐	☐	ZSTRU	STRU	0	0	Stru Description
FIELD_1	☐	☐	ERSDA	DATS	8	0	Created On
FIELD_2	☐	☐	ERNAM	CHAR	12	0	Name of Person who Created the Object
FIELD_3	☐	☐	LAEDA	DATS	8	0	Date of Last Change
FIELD_4	☐	☐	AENAM	CHAR	12	0	Name of person who changed object

Data Element (SE11)

Data element ZBUKRS
Short Description Company Code

Attributes | Data Type | Further Characteristics

Elementary Type
Domain BUKRS Company code
Data Type CHAR Character String
Length 4

Built-in type Data Type
Length 0

Data element ZBUKRS Inactive
Short Description Company Code

Attributes | Data Type | Further Characteristics | Field Label

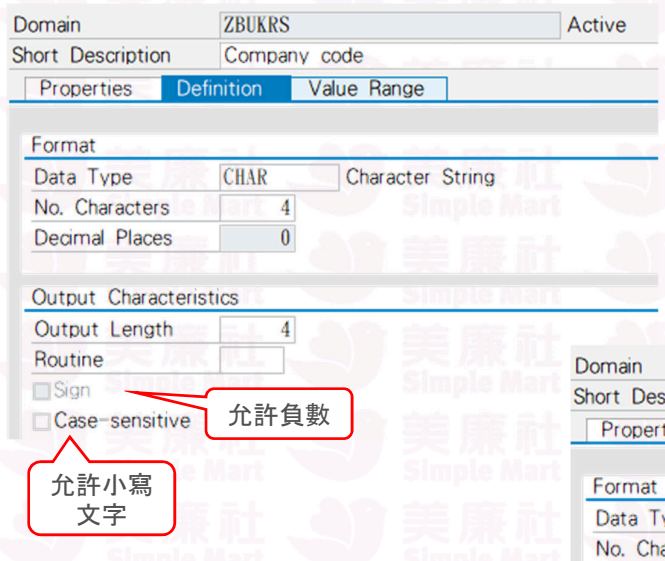
	Length	Field Label
Short	6	CoCode
Medium	15	Company Code
Long	15	Company Code
Heading	4	CoCd

CoCd	Company Name
0001	SAP A.G.

CoCode	Company Name
0001	SAP A.G.

Company Code	Company Name
0001	SAP A.G.

Domain (SE11)

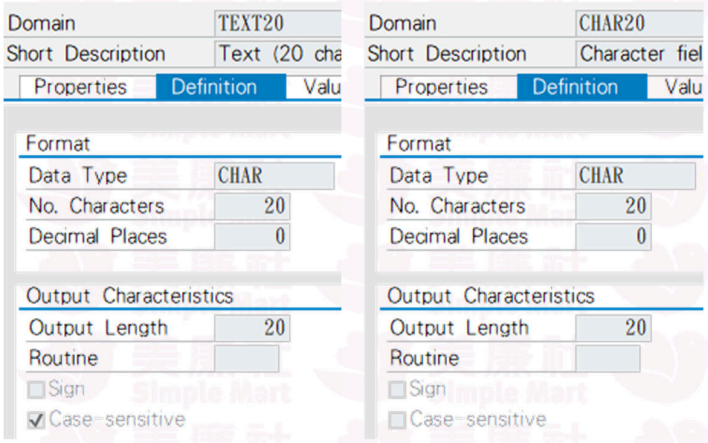


常用 Domain：

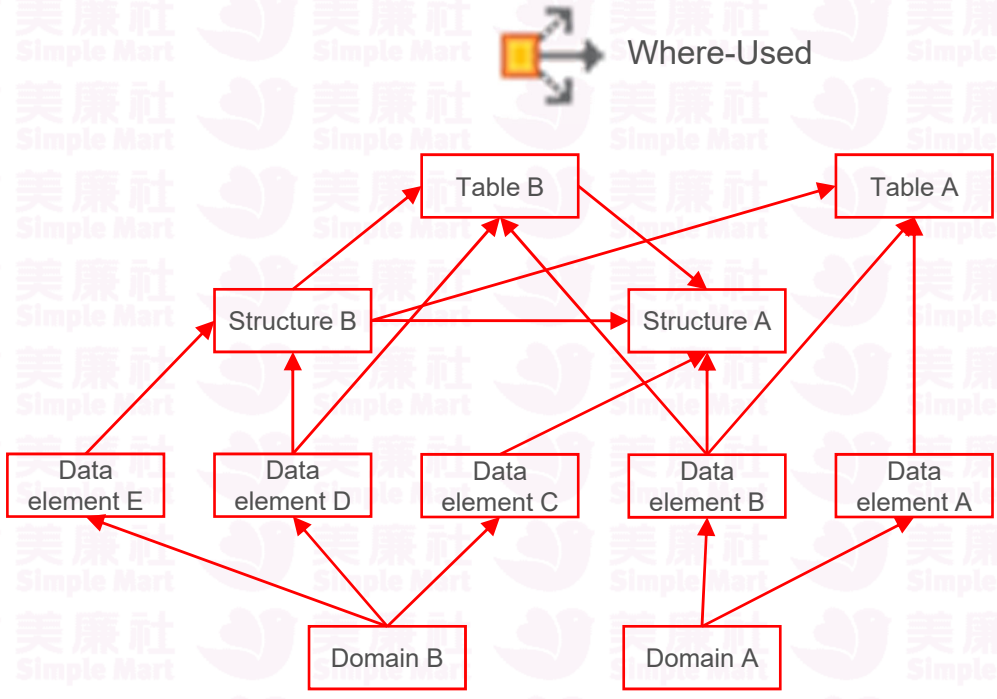
TEXT*：指定長度並區分大小寫的字串。Ex: TEXT20

CHAR*：指定長度但一律轉大寫的字串。Ex: CHAR20

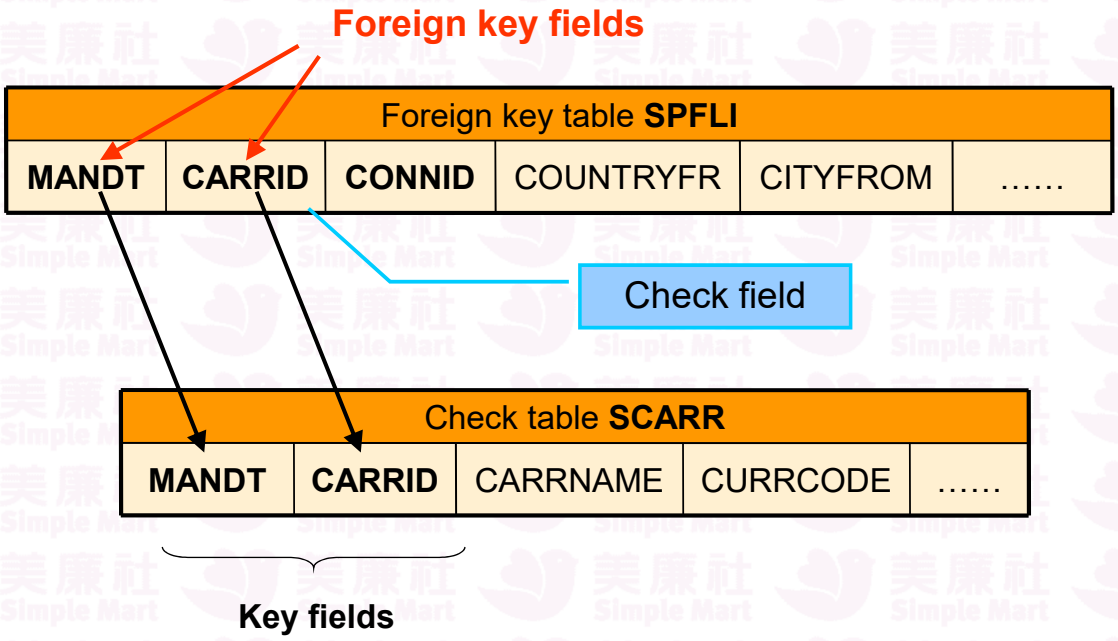
NUMC*、DEC*…



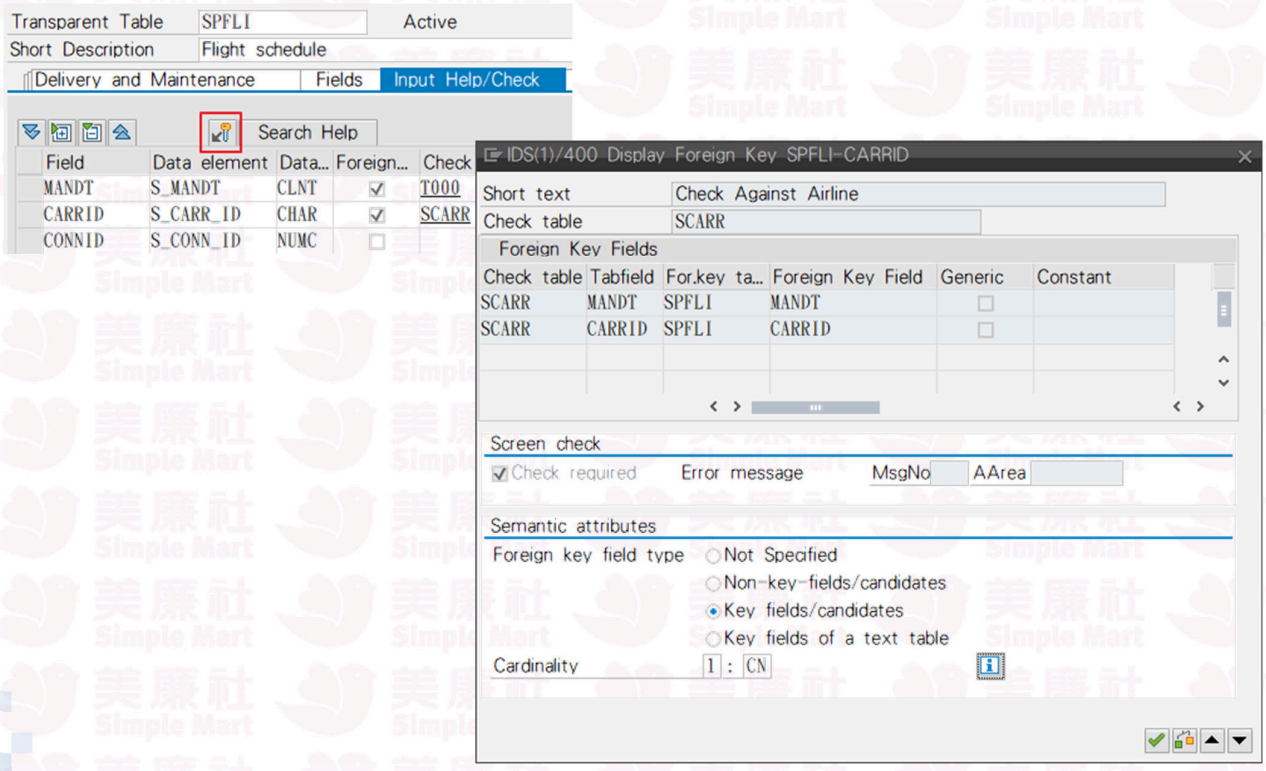
Data type in ABAP Dictionary



ABAP Dictionary – Check table & Foreign key



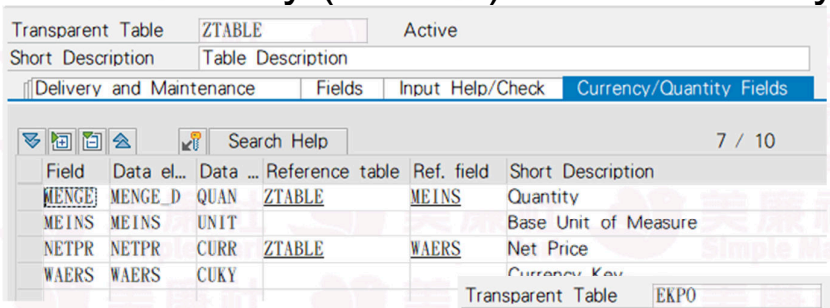
ABAP Dictionary – Check table & Foreign key



ABAP Dictionary – Ref field

Quantity (QUAN) ➔ Unit of measure (UNIT)

Currency (CURR) ➔ Currency key (CUKY)



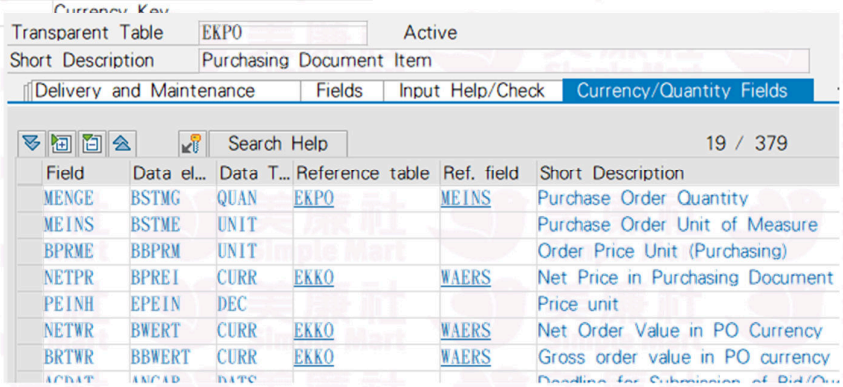
Transparent Table ZTABLE Active

Short Description Table Description

Delivery and Maintenance | Fields | Input Help/Check | Currency/Quantity Fields

Search Help 7 / 10

Field	Data el...	Data ...	Reference table	Ref. field	Short Description
MENGE	MENGE_D	QUAN	ZTABLE	MEINS	Quantity
MEINS	MEINS	UNIT			Base Unit of Measure
NETPR	NETPR	CURR	ZTABLE	WAERS	Net Price
WAERS	WAERS	CUKY			



Transparent Table EKPO Active

Short Description Purchasing Document Item

Delivery and Maintenance | Fields | Input Help/Check | Currency/Quantity Fields

Search Help 19 / 379

Field	Data el...	Data T...	Reference table	Ref. field	Short Description
MENGE	BSTMG	QUAN	EKPO	MEINS	Purchase Order Quantity
MEINS	BSTME	UNIT			Purchase Order Unit of Measure
BPRME	BBPRM	UNIT			Order Price Unit (Purchasing)
NETPR	BPREI	CURR	EKKO	WAERS	Net Price in Purchasing Document
PEINH	EPEIN	DEC			Price unit
NETWR	BWERT	CURR	EKKO	WAERS	Net Order Value in PO Currency
BRTWR	BBWERT	CURR	EKKO	WAERS	Gross order value in PO currency

© 2018 NTT DATA Taiwan Co., Ltd.

Currency Convert Function

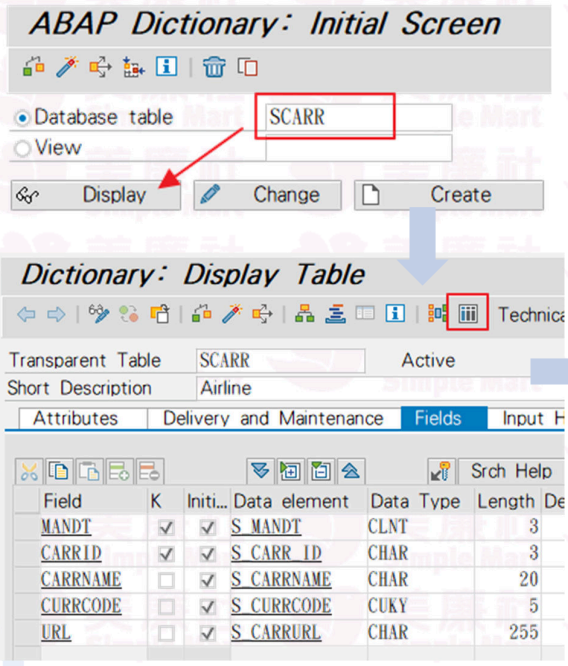
```
CALL FUNCTION 'CURRENCY_CONVERTING_FACTOR'
  EXPORTING
    currency =
  IMPORTING
    factor   =
```

```
CALL FUNCTION 'CURRENCY_AMOUNT_SAP_TO_DISPLAY'
  EXPORTING
    currency        =
    amount_internal =
  IMPORTING
    amount_display  =
```

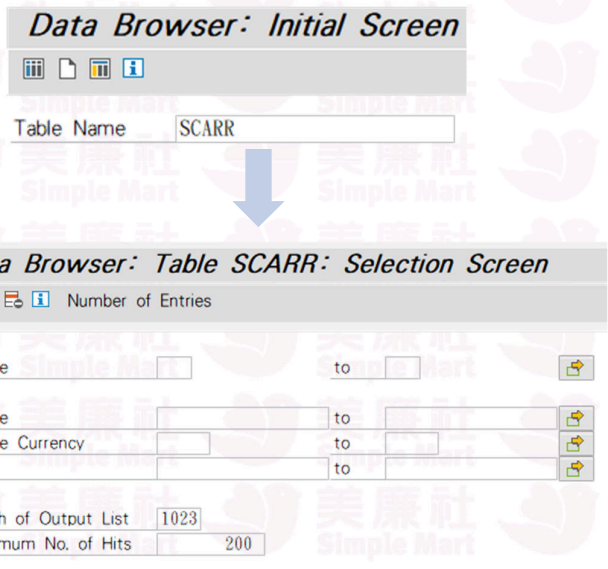
© 2018 NTT DATA Taiwan Co., Ltd.

ABAP Dictionary – Browser Data

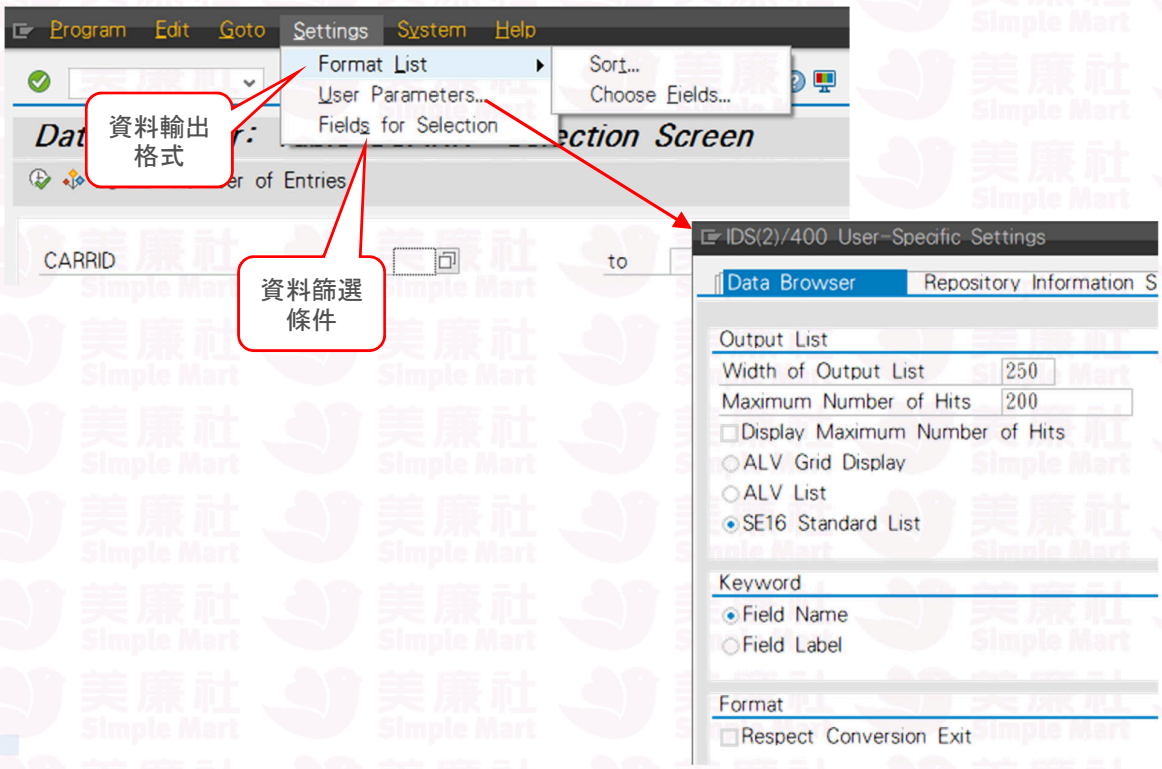
SE11



SE16



Data Browser Setting (SE16)



Using Dictionary Elements to Define Data Objects Within the Program

```
DATA mycarrid TYPE s_carr_id.          " Data Element
DATA myconnid TYPE sfl_info-connid.    " Structure Component
PARAMETERS p_carr TYPE spfli-carrid.   " Table Field
DATA wa_sfl_info TYPE sfl_info.        " Structure
DATA wa_spfli TYPE spfli.              " Transparent Table
```

Defined data objects

mycarrid

myconnid

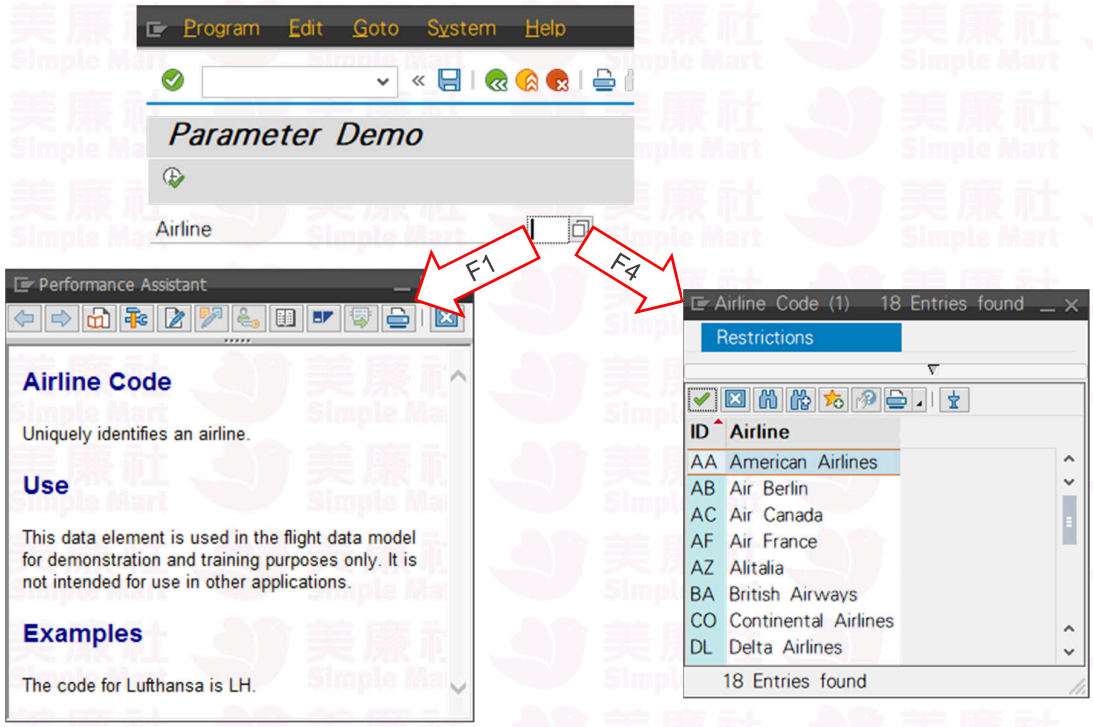
p_carr (at the same time Input field on the selection screen)

wa_sfl_info: pldate, deptime, arrtime, airfrom ...

wa_spfli: mandt, carrid, connid, countryfr ...

PARAMETERS Definition using a Dictionary Element

```
PARAMETERS p_carr TYPE spfli-carrid.
```

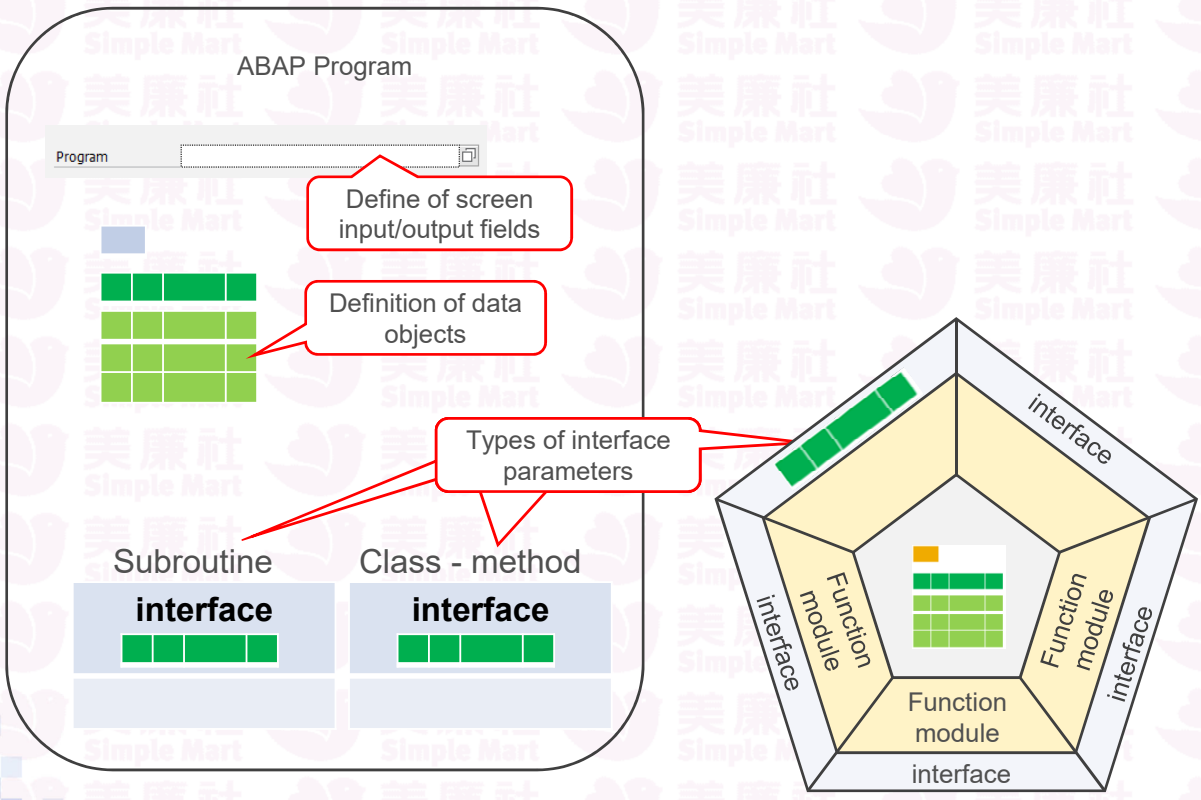


Basic ABAP Language Elements

- ✓Working with Elementary Data Objects
- ✓Working with Structure
- ✓Working with Internal Tables
- ✓Inline declaration

© 2018 NTT DATA Taiwan Co., Ltd.

Using Data Types



© 2018 NTT DATA Taiwan Co., Ltd.

Pre-define Data Type

Type	Length	Range	Initial value	Description
C	1	1 ~ n Byte	Space	Text
N	1	1 ~ n Byte	'000…0'	Numeric text
D	8	8 Bytes	'00000000'	Date
T	6	6 Bytes	'000000'	Time
I	4	$-2^{31} \sim 2^{31} - 1$	0	Integer
INT8	8	$-2^{63} \sim 2^{63} - 1$	0	Integer (NW740)
F	8	8 Bytes	0.00…	Floating point no.
P	8	1~16 Byte	0	Packed number
X	1	1 ~ n Byte	X'00'	Hexadecimal
String	Dynamic	Dynamic	Space	Text
XString	Dynamic	Dynamic		Hexadecimal
DECFLOAT16	8	8 Byte		Packed number (NW702)
DECFLOAT34	16	16 Byte		Packed number (NW702)

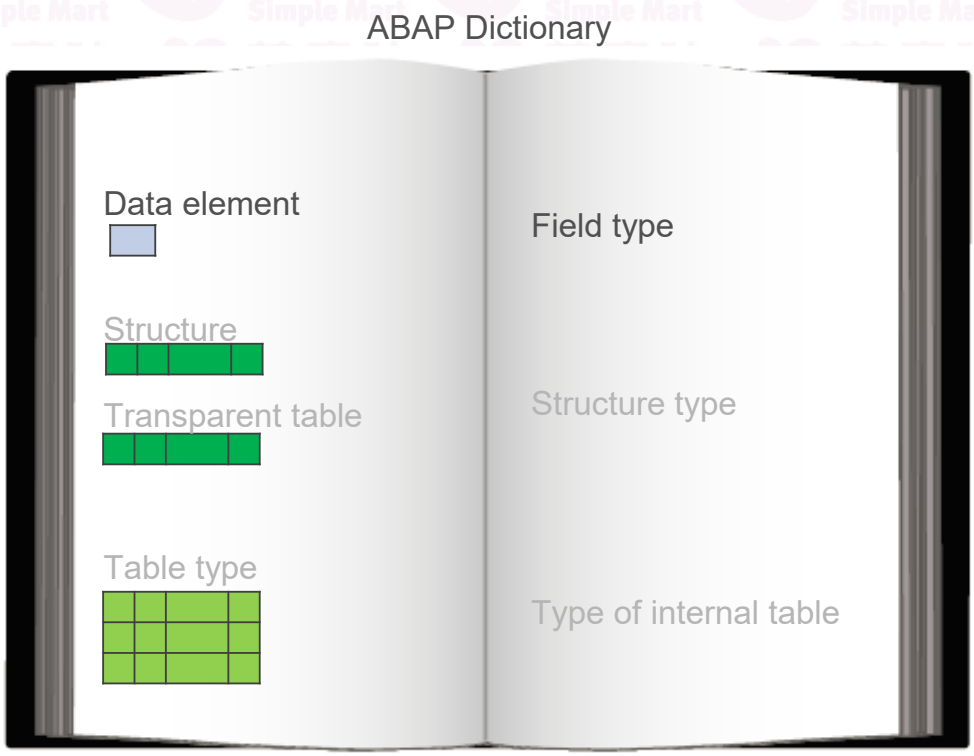
© 2018 NTT DATA Taiwan Co., Ltd.

Declaring Local Types

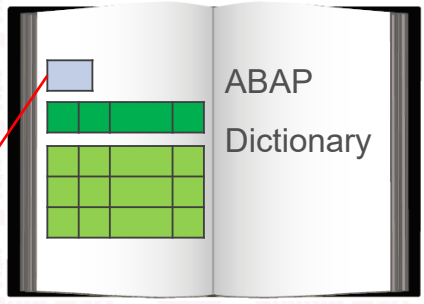
```
TYPES my_c_type TYPE c LENGTH 8.

TYPES: my_n_type(5) TYPE n,
       my_p_type(3) TYPE p DECIMALS 2.

TYPES my_c_type2. " TYPE c LENGTH 1.
```

© 2018 NTT DATA Taiwan Co., Ltd.

Global Data Types in the Dictionary



Defining Data Objects



ABAP Program

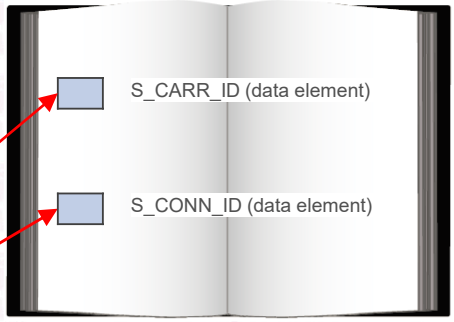
```
" Declaration of local types
TYPES type_name TYPE c.

DATA my_var1 TYPE type_name.

DATA my_var2 LIKE my_var1.
```

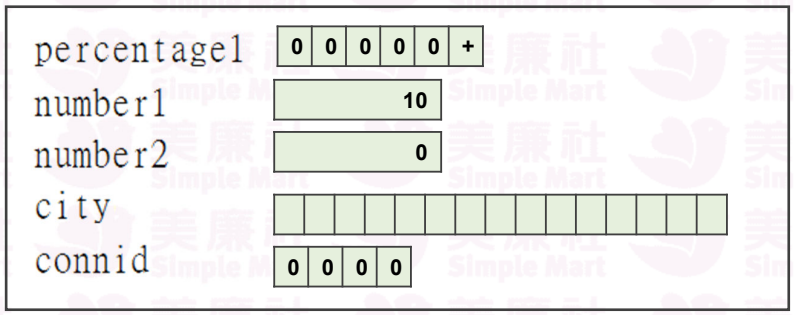
Predefined ABAP Types
(standard types in ABAP)

C / N / D / T
I / F / P / X
String / XString

Examples of the Definition of Elementary Data Objects

```
TYPES t_percentage(3) TYPE p DECIMALS 2.

DATA: percentage1 TYPE t_percentage,
      percentage2 TYPE t_percentage,
      number1 TYPE i VALUE 10,
      number2 LIKE number1,
      city TYPE c LENGTH 15,
      street(20), " TYPE c
      carrid TYPE s_carr_id,
      connid TYPE s_conn_id.
```





Literals and Constants (Fixed Data Objects)

- Fixed data objects without label

Literals

Numeric literals	
Positive integer:	123
Negative integer:	-123

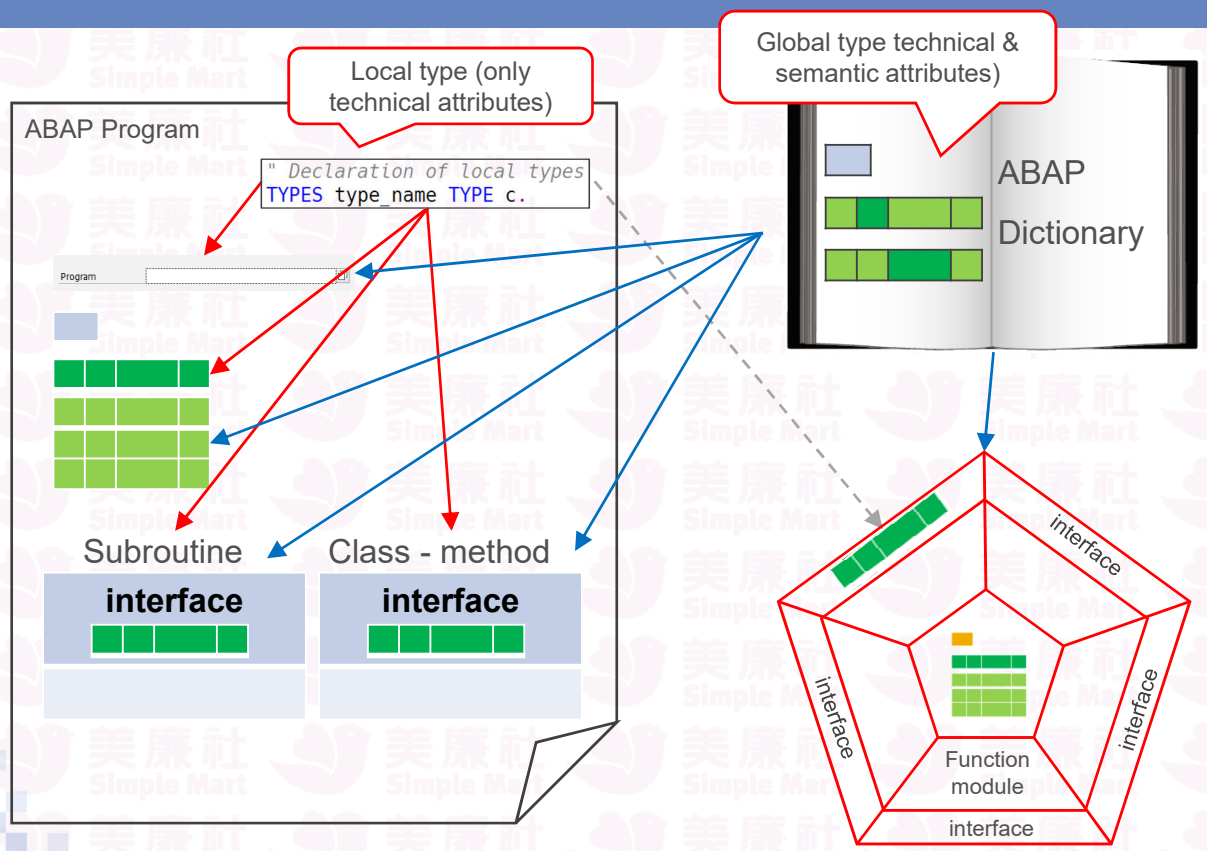
Text literals
Character string: ‘Hello’
Decimal numbers: ‘123.45’
Floating point numbers: ‘123.45E02’

Fixed data objects with label

Constants

```
CONSTANTS my_const TYPE type_name VALUE literal.
```

Local vs. Global Data Types



Value Assignments

ABAP Program

Data declarations

```
CONSTANTS c_qf TYPE s_carr_id VALUE 'QF'.

DATA: carrid1 TYPE s_carr_id,
      carrid2 TYPE s_carr_id VALUE 'LH',
      counter TYPE i.
```

Program start

c_qf | QF

```
MOVE c_qf TO carrid1.
carrid2 = carrid1.
ADD 1 TO counter.
CLEAR: carrid1,
       carrid2,
       counter.
```

carrid1	carrid2	counter
	LH	0
QF	LH	0
QF	QF	0
QF	QF	1
	QF	1
		1
		0

Calculation: Syntax

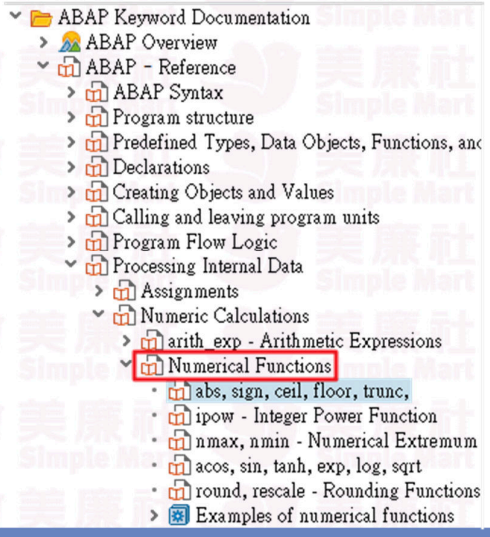
```
[COMPUTE] result = [+|-] operand1
                   [{+|-|*|/|DIV|MOD|**} [+|-] operand2
                   [{+|-|*|/|DIV|MOD|**} [+|-] operand3
                   ... ]].
```

Operator	Calculation	Priority	Order
+	Addition of operands	1	from left to right
–	Subtraction of the right from the left operand	1	from left to right
*	Multiplication of operands	2	from left to right
/	Division of the left by the right operand	2	from left to right
DIV	Integer portion of the division of the left by the right operand	2	from left to right
MOD	Integer remainder of the division of the left by the right operand	2	from left to right
**	Exponentiation of the left with the right operand	3	from right to left

© 2018 NTT DATA Taiwan Co., Ltd.

Build-in Function - Mathematical

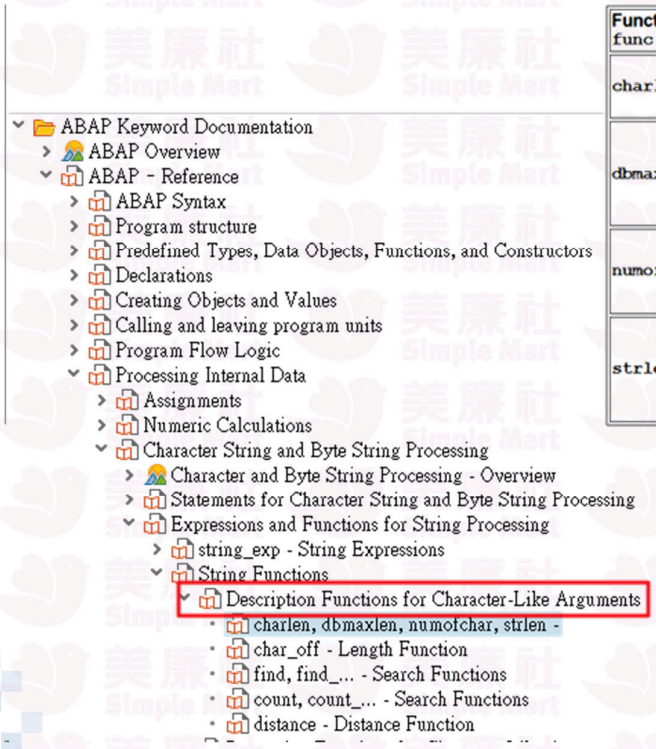
Function func	Return Value
abs	Absolute value of argument arg
sign	Sign of argument arg: -1, if the value of arg is negative; 0, if the value of arg is 0; 1, if the value of arg is positive
ceil	Smallest integer that is not less than the value of the argument arg is
floor	Largest integer that is not greater than the value of the argument arg is
trunc	Value of the integer part of the argument arg
frac	Value of the decimal places of the argument arg



Function func	Meaning	Definition Range
acos	arccosine	[-1,1], no decfloat34
asin	arcsine	[-1,1], no decfloat34
atan	arctangent	-, no decfloat34
cos	cosine	-, no decfloat34
sin	sine	-, no decfloat34
tan	tangent	-, no decfloat34
cosh	hyperbolic cosine	-, no decfloat34
sinh	hyperbolic sine	-, no decfloat34
tanh	hyperbolic tangent	-, no decfloat34
exp	Exponential function for base e	[-709, 709] for type f and [-14144, 14149] for type decfloat34
log	Natural logarithm	> 0
log10	Logarithm to base 10	> 0
sqrt	Square root	>= 0

© 2018 NTT DATA Taiwan Co., Ltd.

Build-in Function - Description



Function `func`	Return Value
`charlen`	Length of first character of `arg` in the code page used: 1 for non-Unicode single-byte code; 1 or 2 for non-Unicode double-byte code; 1 for Unicode with simple Unicode character; 2 for Unicode with surrogates;
`dbmaxlen`	Maximum length of a string defined in ABAP Dictionary (RAWSTRING, SSTRING, STRING). If the string is unrestricted, the constant `abap_max_db_string_ln` or `abap_max_db_rawstring_ln` from the type group ABAP is returned. The latter is also returned for the predefined ABAP types `string` and `xstring`.
`numofchar`	Number of characters in `arg`, where closing blanks are not counted in data objects with fixed lengths or in data objects with the type `string`. If a non-Unicode double-byte code is used, a character that uses two bytes is only counted once.
`strlen`	Number of characters in `arg`, where closing blanks in data objects with fixed lengths are not counted. They are counted though in data objects with the type `string`. If a non-Unicode double-byte code is used, a character that uses two bytes is counted twice. The argument can be byte-like outside Unicode programs.

Processing Character String

CONCATENATE	‘AB’ + ‘AP’ => ‘ABAP’
& , && (NW731)	‘AB’ & ‘AP’ => ‘ABAP’ ‘AB’ && ‘AP’ => ‘ABAP’
CONDENSE	‘ A P ‘ => ‘AP’
FIND	‘ABAP’
OVERLAY	‘AB P’ & ‘AAAA’ => ‘ABAP’
REPLACE	‘ABAP’ & ‘A’ to ‘B’ => ‘BBAP’
SHIFT	‘ABAP’ => ‘BAP ‘
SPLIT	‘ABAP’ with ‘B’ => ‘A’ + ‘AP’
TRANSLATE	‘ABAP’ => ‘abap’
<FIELD>+<off>(length)	

Concatenating Character Strings (CONCATENATE)

```
PARAMETERS: p_fname(30) TYPE c,
            p_lname(30) TYPE c.

DATA: g_name TYPE string,
      g_sep  TYPE c.

CONCATENATE p_fname p_lname
       INTO g_name
       SEPARATED BY g_sep.

WRITE:/ 'Name:', g_name.
WRITE:/ 'Name:', p_fname, p_lname.
```

```
ABAP Demo

Name: MICHAEL CHUANG
Name: MICHAEL                                  CHUANG
```


New String Expressions (NW731)

```
DATA: a1     TYPE p DECIMALS 2 VALUE '456.78',
      a2(50).
" & 僅能接字串
a2 = `Information ` & ` concatenate`.
a2 = 'Information ' & ' concatenate'.
a2 = |Information | & | concatenate|.
" && 能接字串與變數
a2 = 'Information ' && a1 && ' concatenate'.
a2 = `Information ` && a1 && ` concatenate`.
a2 = |Information | && a1 && | concatenate|.
a2 = |Information { a1 } concatenate|.
a2 = |Information { a1 WIDTH = 10 ALIGN = CENTER }|
  && | concatenate|.
"Char |, {, and } have to be escaped by \ in literal text.
a2 = |Char \|, \{, and \} have to be escaped by \\|
  && | in literal text.|.
```

Searching (FIND) and Replacing (PEPLACE) within Character Strings

```
DATA: g_text TYPE string
 VALUE 'Introduction to Programming with JAVA'.

WRITE:/ g_text.

FIND FIRST OCCURRENCE OF SUBSTRING 'JAVA'
  IN g_text IGNORING CASE.
IF sy-subrc = 0.
  REPLACE ALL OCCURRENCES OF
    SUBSTRING 'JAVA' IN g_text
    WITH 'ABAP' RESPECTING CASE.
ENDIF.
WRITE:/ g_text.
```

```
ABAP Demo
-----------------------------------------
Introduction to Programming with JAVA
Introduction to Programming with ABAP
```


Splitting Character String(SPLIT)

```
DATA: g_text  TYPE string
        VALUE 'Michael Jordan',
      g_fname TYPE string,
      g_lname TYPE string.

SPLIT g_text AT '' INTO g_fname g_lname.

WRITE:/ 'First name:', g_fname,
      / 'Last name:', g_lname.
```

```
ABAP Demo
-----------------------------------------
First name: Michael
Last name: Jordan
```

Substring Access

- **Example for <FIELD>+<off>(length)**

```
DATA: a1(10) TYPE c VALUE 'ABCDEFGHIJ'.
DATA: a2(5) TYPE c.
DATA: a3 TYPE i VALUE 3, a4 TYPE i VALUE 2.

a2 = a1+4(2).    " a2 = 'EF'.
a2 = a1+3.       " a2 = 'DEFGH'.
a2 = a1(3).      " a2 = 'ABC'.
a2 = a1+a3(a4). " a2 = 'DE'.
```

© 2018 NTT DATA Taiwan Co., Ltd.

Conditional Branches

```
IF var > 0.
  " Statement
ELSEIF var = 0.
  " Statement
ELSE.
  " Statement
ENDIF.
```

```
CASE var.
  WHEN 'AA'.
    " Statement
  WHEN 'BB' OR 'CC'.
    " Statement
  WHEN OTHERS.
    " Statement
ENDCASE.
```

© 2018 NTT DATA Taiwan Co., Ltd.

Examples: IF statement

```
" Negation
IF gv_carrid IS NOT INITIAL.
  " Statement
ELSE.
  " Statement
ENDIF.
" AND and OR links with parentheses
IF ( gv_carrid = 'AA' OR gv_carrid = 'LH' )
  AND gv_fldate = sy-datum.
  " Statement
ELSEIF ( gv_carrid = 'UA' OR gv_carrid = 'DL' )
  AND gv_fldate >= sy-datum.
  " Statement
ENDIF.
" Negation before logical conditions
IF NOT ( gv_carrid = 'AA' OR gv_carrid = 'DL' )
  AND gv_fldate < sy-datum.
  " Statement
ENDIF.
```

© 2018 NTT DATA Taiwan Co., Ltd.

Relational Operator

Operator	Meaning
= , EQ	Equal
<> , NE	Not Equal
< , LT	Lower Than
> , GT	Greater Than
<= , LE	Lower Equal
>= , GE	Greater Equal
~~><~~	~~Obsolete~~
~~=<~~	~~Obsolete~~
~~=>~~	~~Obsolete~~

Operator	Operator
AND	OR
NOT	( )

Operator	Meaning
CO	Contains Only
CN	Contains Not Only
CA	Contains Any
NA	Contains Not Any
CS	Contains String
NS	Contains Not String
CP	Covers Pattern
NP	Not Pattern

© 2018 NTT DATA Taiwan Co., Ltd.

Loops

```
SELECT * FROM db_table ...
  " Statement
ENDSELECT.

LOOP AT <itab>.
  " Statement
ENDLOOP.
```

```
DO.
  " Statement
  IF <abort_condition>.
    EXIT.
  ENDIF.
ENDDO.

DO n TIMES.
  " Statement
ENDDO.

WHILE <condition>.
  " Statement
ENDWHILE.
```

Loop counter

sy-index

© 2018 NTT DATA Taiwan Co., Ltd.

Dialog Messages

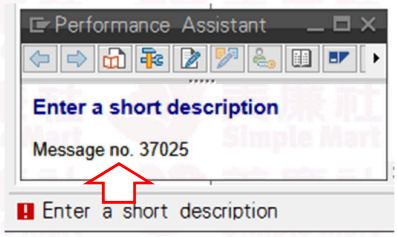
- MESSAGE tnnn(message_class) [WITH V1 [V2] [V3] [V4]].
  - t : message type
  - nnn : message number

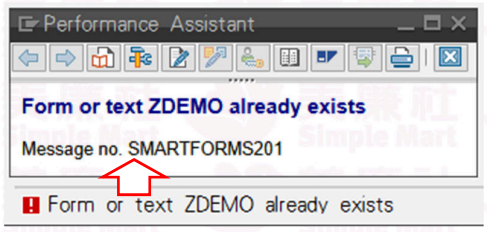
Type	Description	Dialog behavior	Message appears in
I	Information	Program continues after interruption	Model dialog box
S	Success / status	Program continues without interrupt	Status bar in next screen
W	Warning	Context-dependent	Status bar or modal dialog box
E	Error		
~~A~~	~~Termination~~	~~Program cancelled~~	~~Modal dialog box~~
~~X~~	~~Short dump~~	~~Runtime error MESSAGE_TYPE_X triggered~~	~~Short dump~~

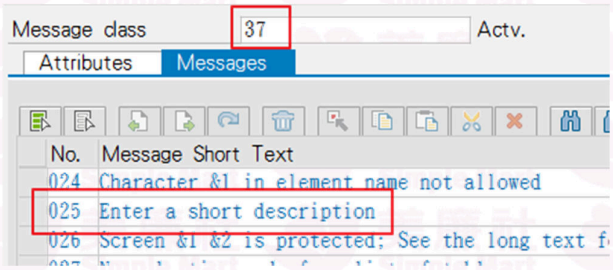
© 2018 NTT DATA Taiwan Co., Ltd.

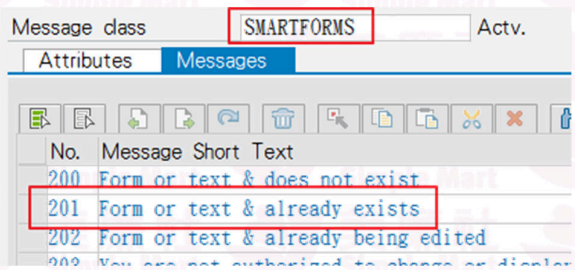
Find message class & message number

- Double-click in status bar





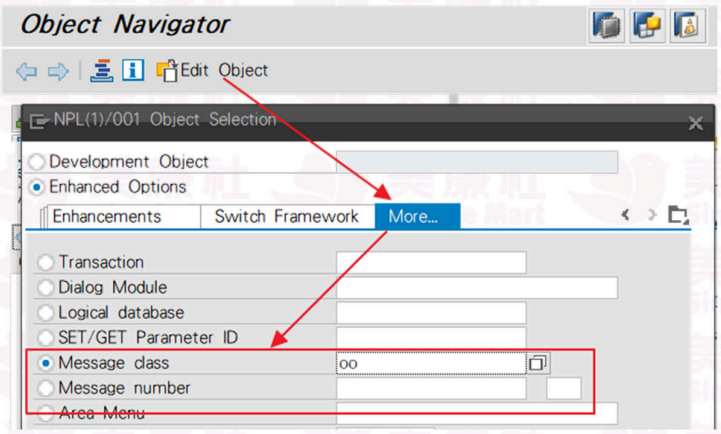


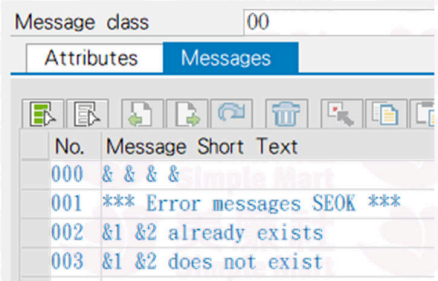


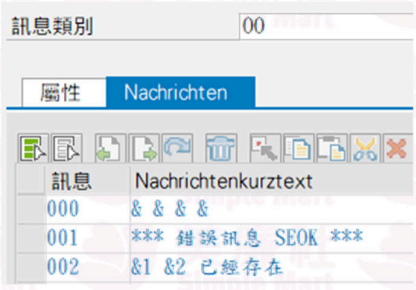
© 2018 NTT DATA Taiwan Co., Ltd.

NTT DATA

Display message class / number in SAP GUI



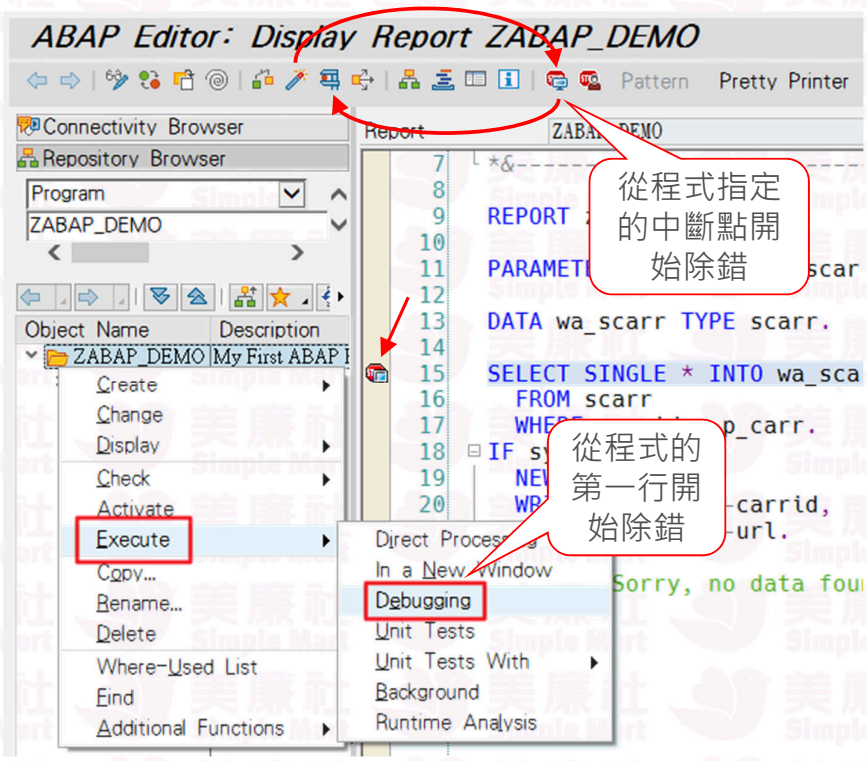




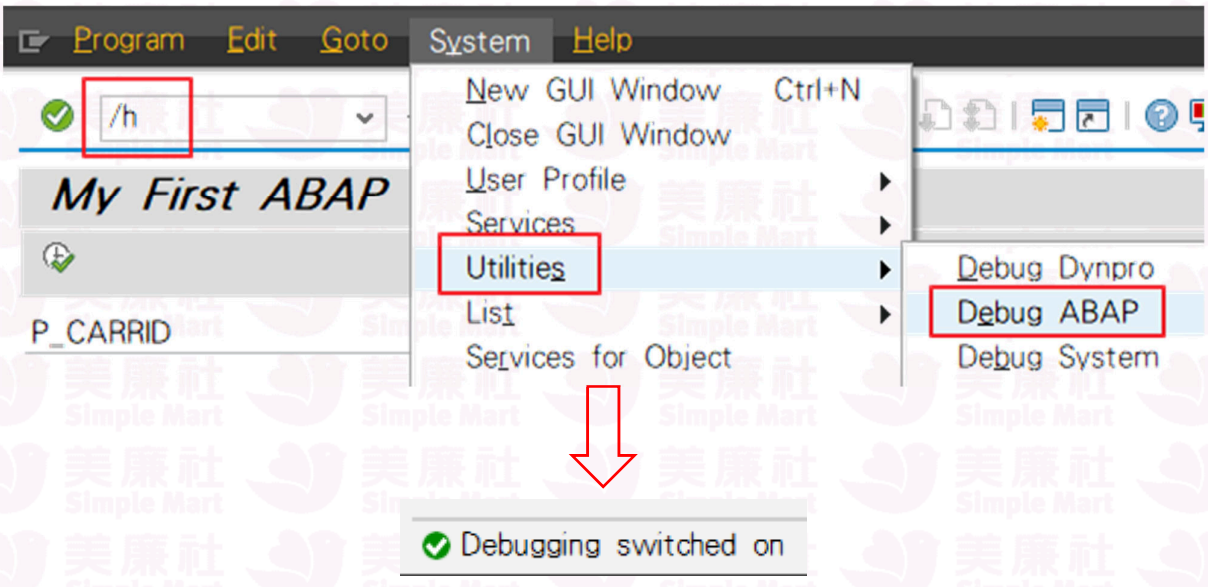
© 2018 NTT DATA Taiwan Co., Ltd.

NTT DATA

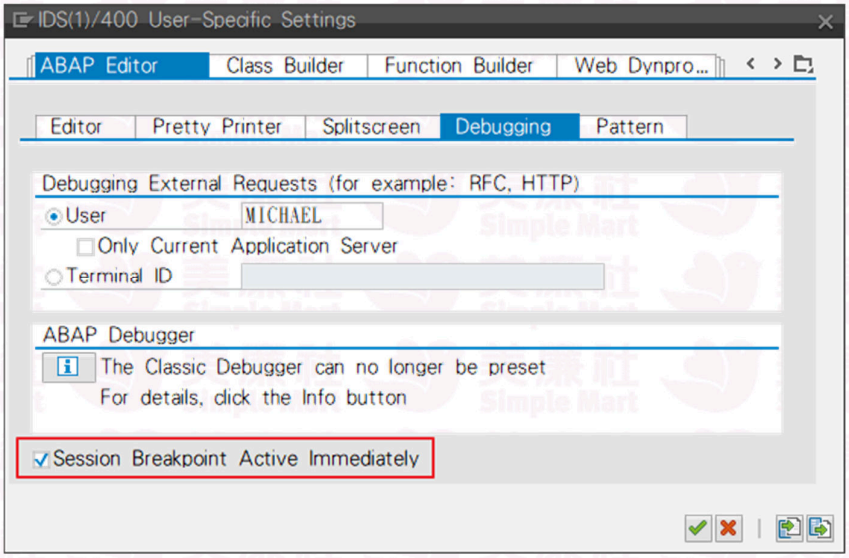
Executing a Program in Debugging Mode



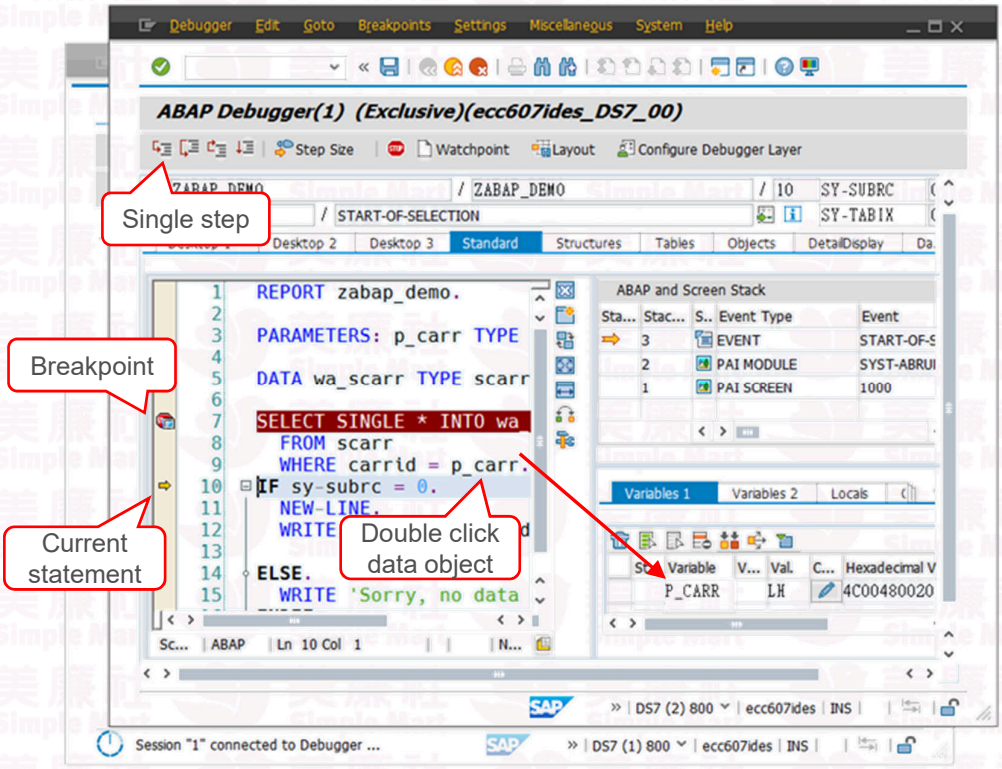
Switching to Debugging Mode at Runtime



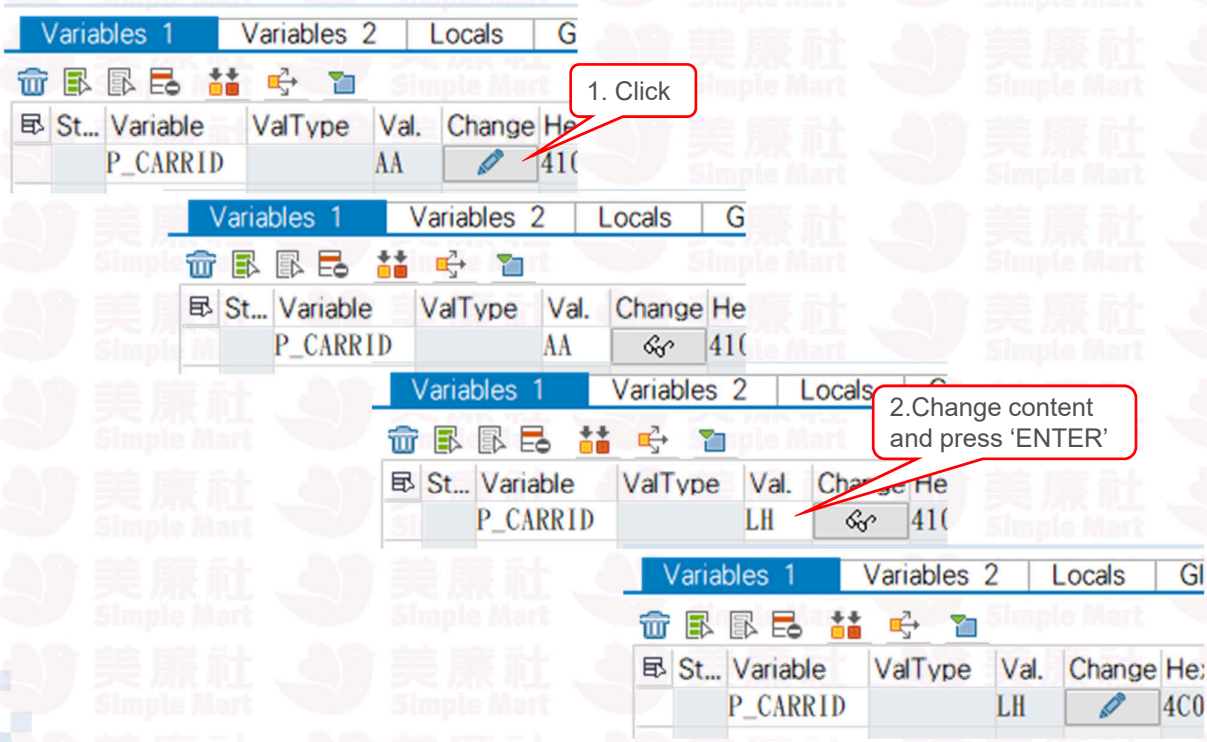
Debugger – in User-Specific Settings



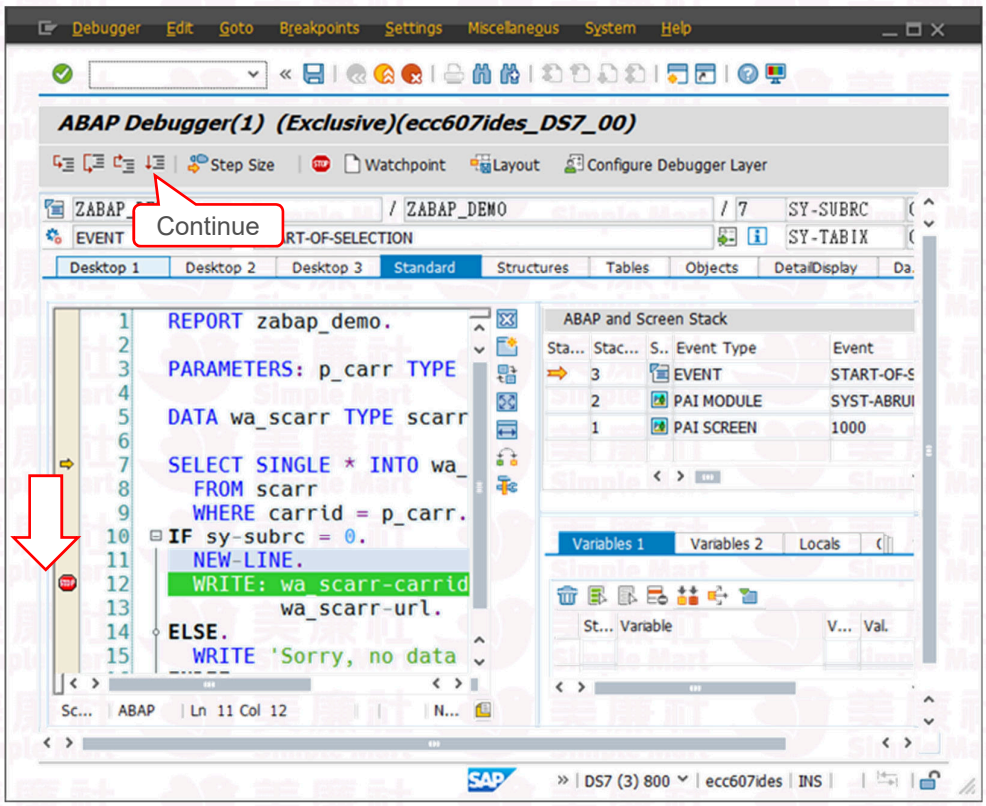
Debugger – Single Step & Field Contents



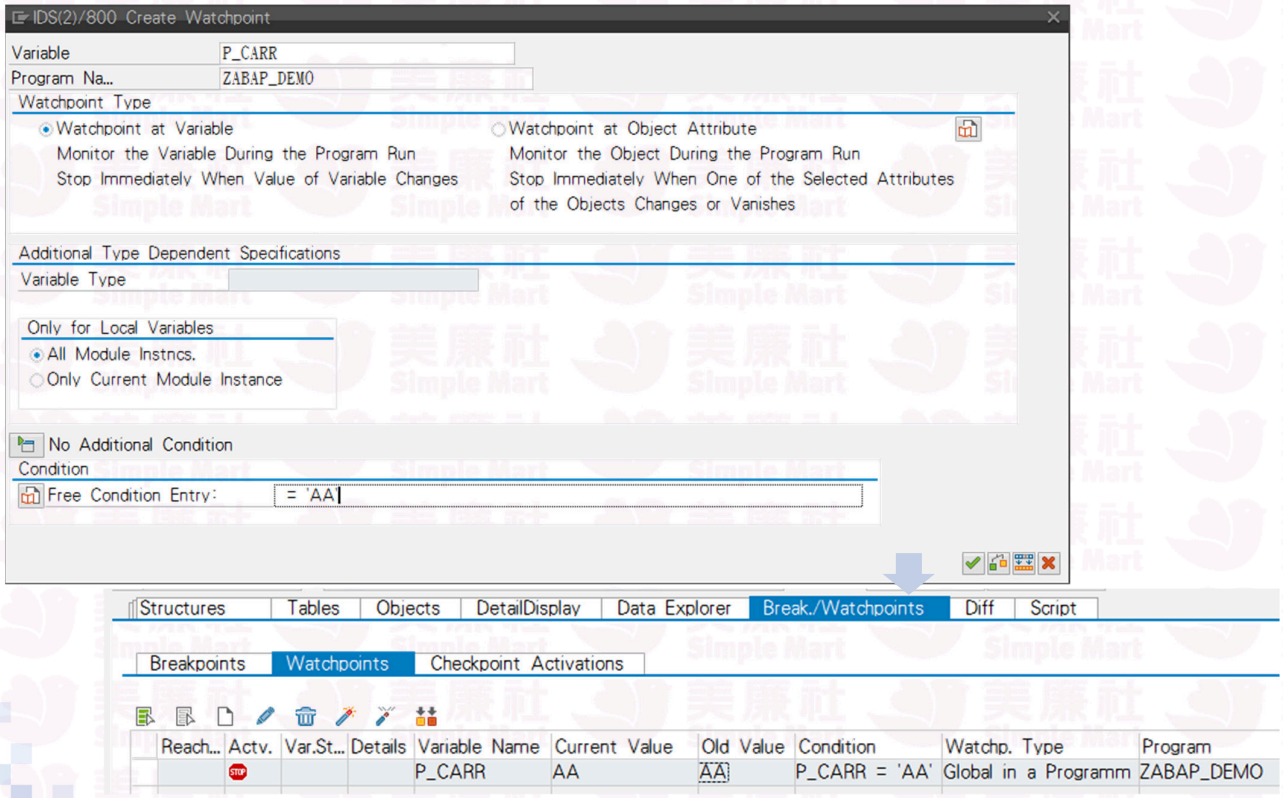
Debugger : Change Field Contents



Debugger : Breakpoint



Debugger : Tracing Data Changes



© 2018 NTT DATA Taiwan Co., Ltd.

NTT DATA

Triggering a Runtime Error

```
DATA: g_a      TYPE i VALUE 5,
      g_b      TYPE i,
      g_result TYPE p DECIMALS 2.

g_result = g_a / g_b.

WRITE:/ 'Result:', g_result.
```

Division by Zero !

© 2018 NTT DATA Taiwan Co., Ltd.

NTT DATA

Short Dumps Triggered by Runtime Errors

Runtime Error – Description of Exception

Long Text Debugger

Category	ABAP Programming Error
Runtime Errors	BCD_ZERODIVIDE
Except.	CX_SY_ZERODIVIDE
ABAP Program	ZABAP_DEMO
Application Component	Not assigned
Date and Time	2015/10/25 11:21:10

Short Text
Division by 0 (type P) in program "ZABAP_DEMO"

What happened?
Error in the ABAP Application Program

The current ABAP program "ZABAP_DEMO" had to be terminated because it has come across a statement that unfortunately cannot be executed.

Error analysis
An exception has occurred which is explained in more detail below. The exception, which is assigned to class 'CX_SY_ZERODIVIDE' was not caught and therefore caused a runtime error. The reason for the exception is:
In the current program "ZABAP_DEMO", an arithmetic operation ('DIVIDE', '/', 'DIV' or 'MOD') with a type P operand attempted to divide by 0.

Missing Handling of System Exception

Program	ZABAP_DEMO

Trigger Location of Exception

Program	ZABAP_DEMO
Include	ZABAP_DEMO
Row	7
Module Name	START-OF-SELECTION

Source Code Extract

Line	SourceCde
1	REPORT zabap_demo.
2	
3	DATA: g_a TYPE i VALUE 5,
4	g_b TYPE i,
5	g_result TYPE p DECIMALS 2.
6	
>>>>>	g_result = g_a / g_b.
8	
9	WRITE:/ 'Result:', g_result.

Active Calls/Events

No.	Ty.	Program	Include
	Name		
1	EVENT START-OF-SELECTION	ZABAP_DEMO	ZABAP_DEMO

© 2018 NTT DATA Taiwan Co., Ltd.

ABAP Runtime Error – ST22

ABAP Runtime Errors – All Clients

Parameters

Standard

Today	1	Runtime Errors
Yesterday	0	Runtime Errors

Own selection

Date	2015.10.25	to	
Time	00:00:00	to	00:00:00
Host		to	
Work Process Index		to	
User	MICHAEL	to	
Client		to	
To be stored		to	
Runtime Error		to	
Terminated Program		to	
Exception		to	
Transaction ID		to	
EPP Full Context ID		to	
EPP Connection ID		to	

Start

List of Selected Runtime Errors

These files were

With Short Text

Corresponding A

Use old dump

Runtime Errors

Date	Time	Application Server	User	Client	Keep	Runtime Error	Exception	Program	WP	Transaction ID
2015.10.25	11:37:09	NKIDESSRV1_IDS_00	MICHAEL	800	C	BCD_ZERODIVIDE	CX_SY_ZERODIVIDE	ZABAP_DEMO	4	AAC97AE52DA1F18180BF901B0E0F83CF

© 2018 NTT DATA Taiwan Co., Ltd.

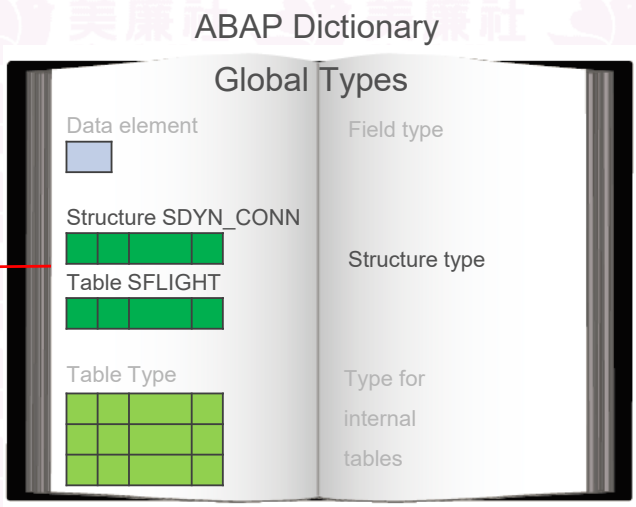
Basic ABAP Language Elements

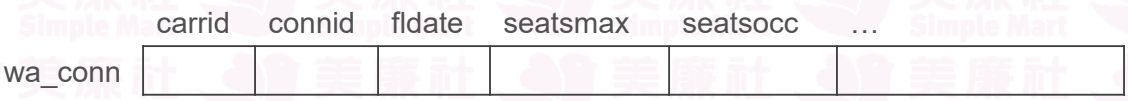
- ✓Working with Elementary Data Objects
- ✓Working with Structure
- ✓Working with Internal Tables
- ✓Inline declaration

© 2018 NTT DATA Taiwan Co., Ltd.

Definition of Structures with Global Types

```
DATA
  wa_conn TYPE sdyn_conn.
```





© 2018 NTT DATA Taiwan Co., Ltd.

Defining Structures with Local Types

```
" Declaration of a local structure type
TYPES: BEGIN OF st_flight,
         carrid TYPE s_carr_id,
         connid TYPE s_conn_id,
         fldate TYPE s_date,
         seatsmax TYPE sflight-seatsmax,
         seatsocc TYPE sflight-seatsocc,
         percentage(3) TYPE p DECIMALS 2,
       END OF st_flight.

" Definition of a structure variable
DATA wa_flightinfo TYPE st_flight.
```

```
DATA: BEGIN OF wa_flightinfo,
        carrid TYPE s_carr_id,
        connid TYPE s_conn_id,
        fldate TYPE s_date,
        seatsmax TYPE sflight-seatsmax,
        seatsocc TYPE sflight-seatsocc,
        percentage(3) TYPE p DECIMALS 2,
      END OF wa_flightinfo.
```

	carrid	connid	fldate	seatsmax	seatsocc	percentage
wa_flightinfo						

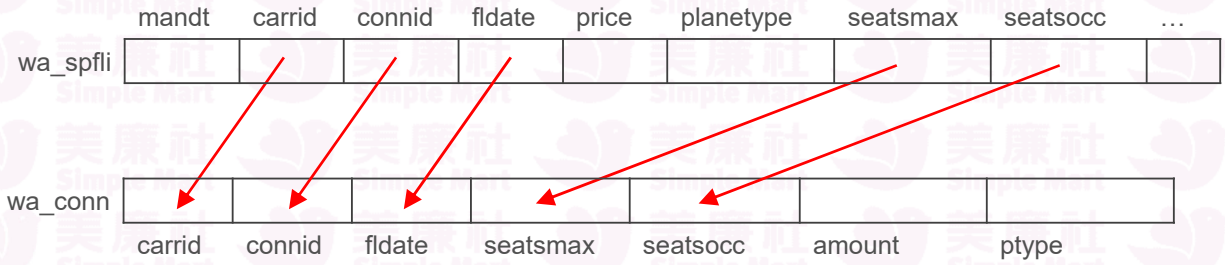
Access to Structure Components

```
wa_scarr-carrid = 'DL'.

SELECT SINGLE * INTO @wa_scarr
  FROM scarr
  WHERE carrid = @wa_scarr-carrid.
IF sy-subrc = 0.
  WRITE: wa_scarr-carrid,
         wa_scarr-carrname,
         wa_scarr-url.
ENDIF.
```

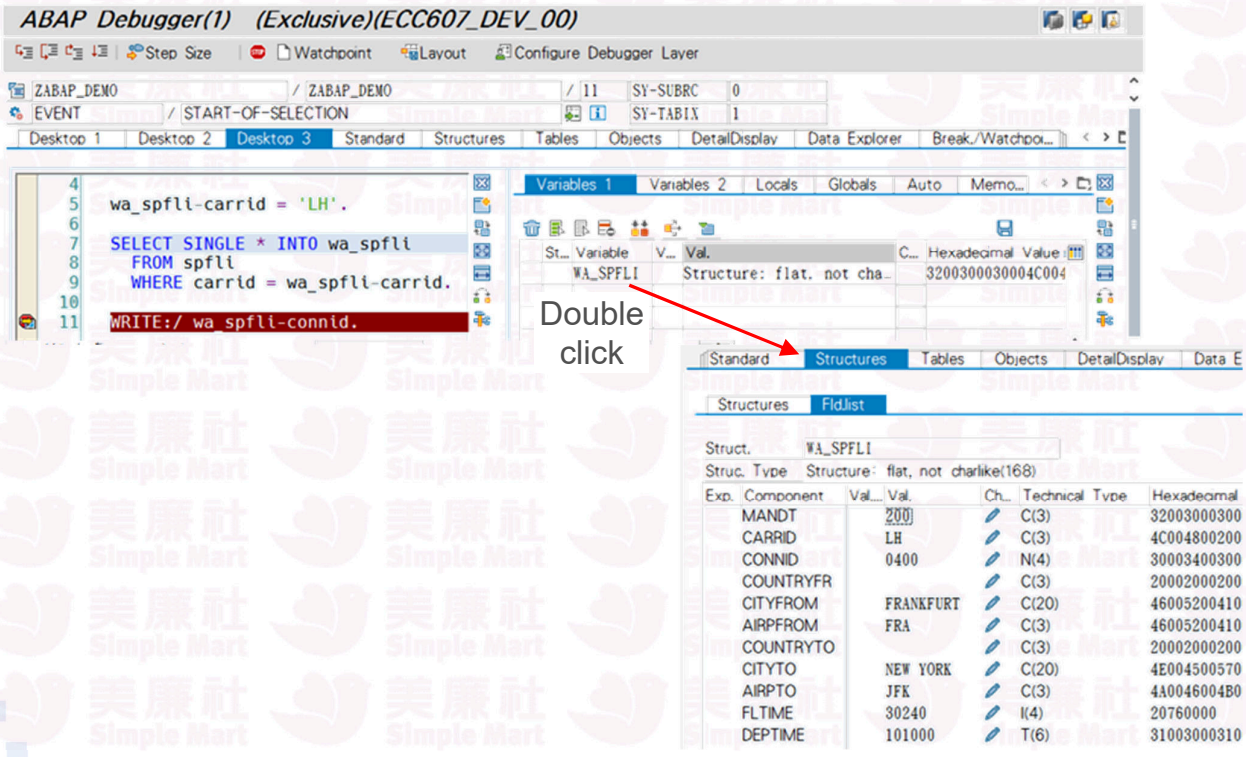
	mandt	carrid	carrname	currcode	url
wa_scarr	400	LH	Lufthansa	EUR	www.lufthansa.com

Copying Structure Components with the **Same Name**

```
DATA:
  wa_conn  TYPE sdyn_conn,
  wa_spfli TYPE spfli.

MOVE-CORRESPONDING wa_spfli TO wa_conn.

"new from NW740
wa_conn = CORRESPONDING #( wa_spfli ).
```



Structures in Debugging Mode

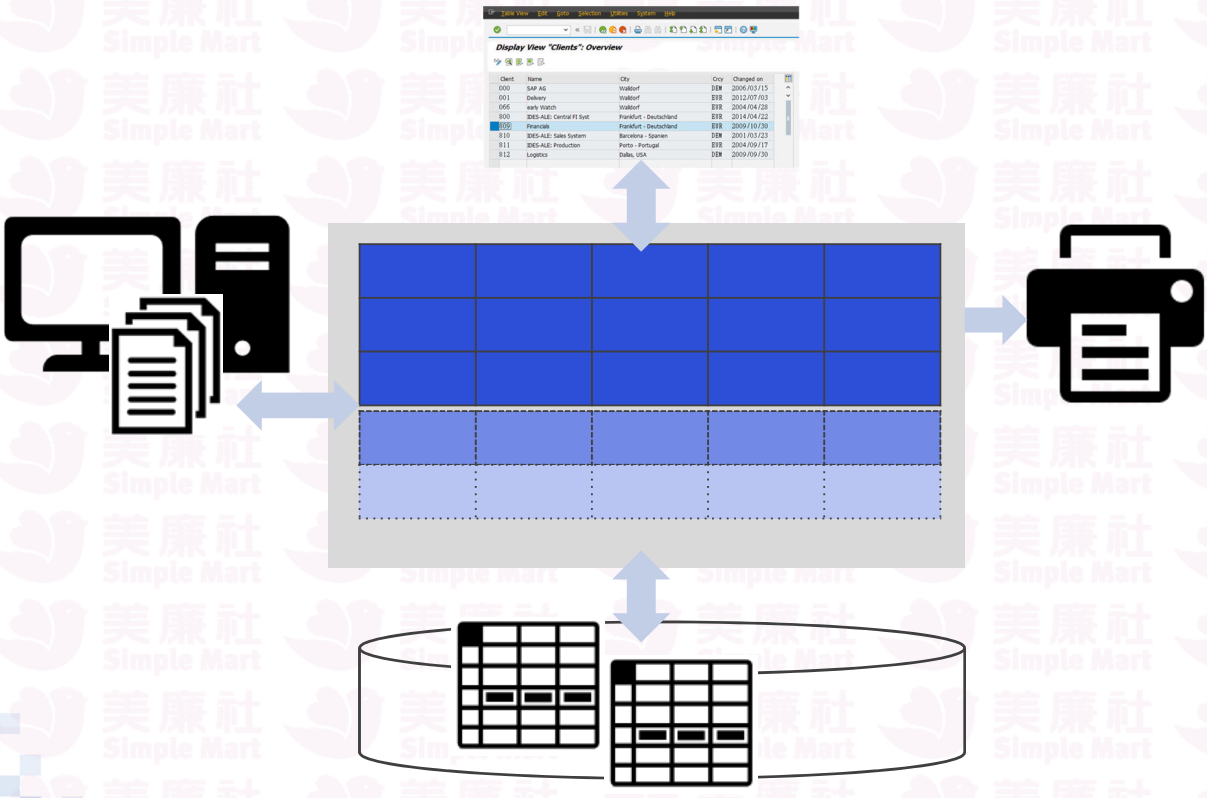


Basic ABAP Language Elements

- ✓Working with Elementary Data Objects
- ✓Working with Structure
- ✓Working with Internal Tables
- ✓Inline declaration

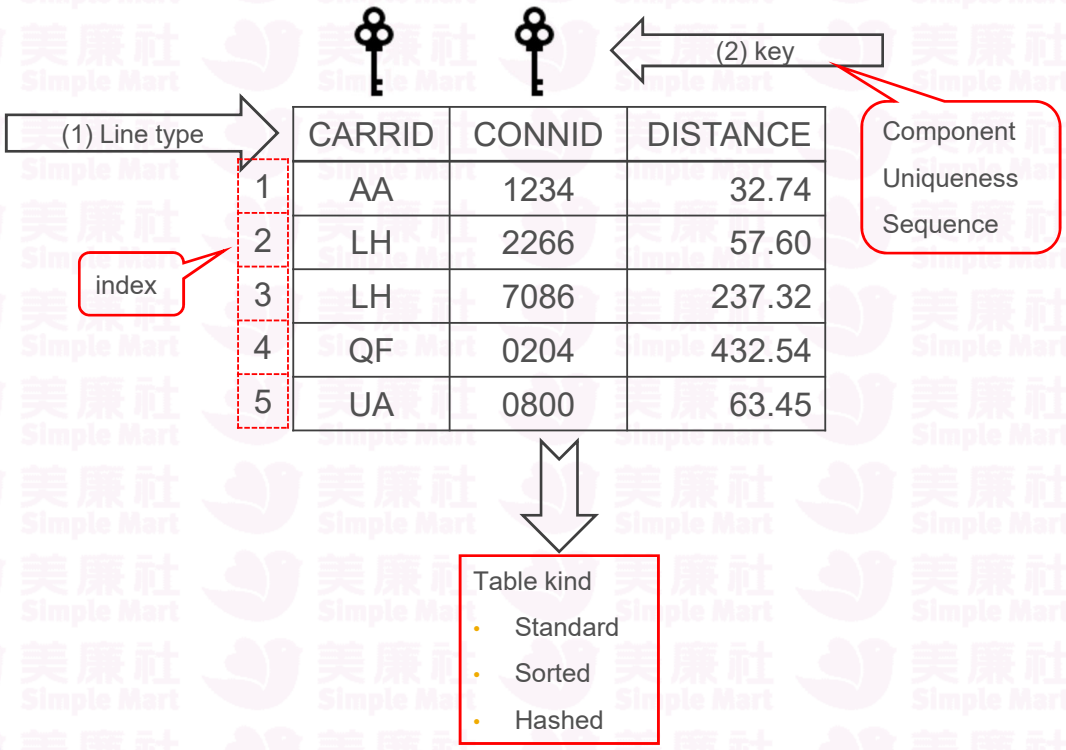
© 2018 NTT DATA Taiwan Co., Ltd.

Internal Tables: Options



© 2018 NTT DATA Taiwan Co., Ltd.

Attributes of Internal Tables



© 2018 NTT DATA Taiwan Co., Ltd.

NTT DATA

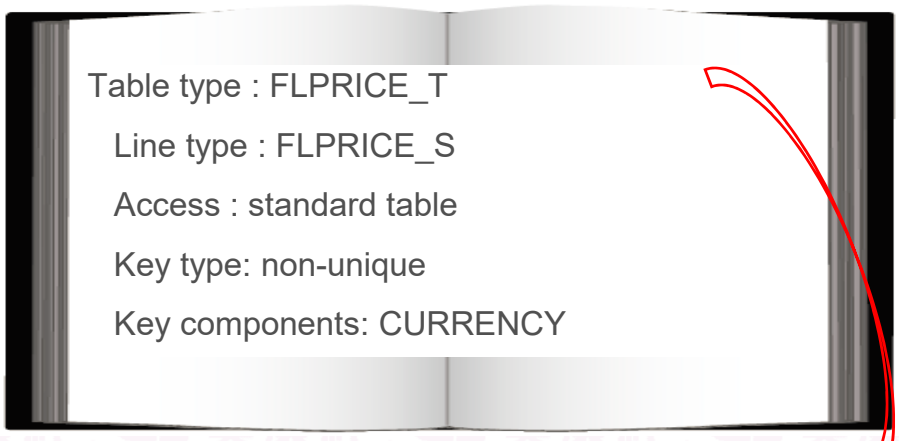
Attributes and Use of the Table Types

	Index table		Hashed table
Table kind	Standard table	Sorted table	Hashed table
Index access	V	V	
Key access	V	V	V
Uniqueness of key	Non-unique	Unique Non-unique	Unique
Use in	Primarily index access	Primarily key access	Only key access
Change key field	V	X	X

© 2018 NTT DATA Taiwan Co., Ltd.

NTT DATA

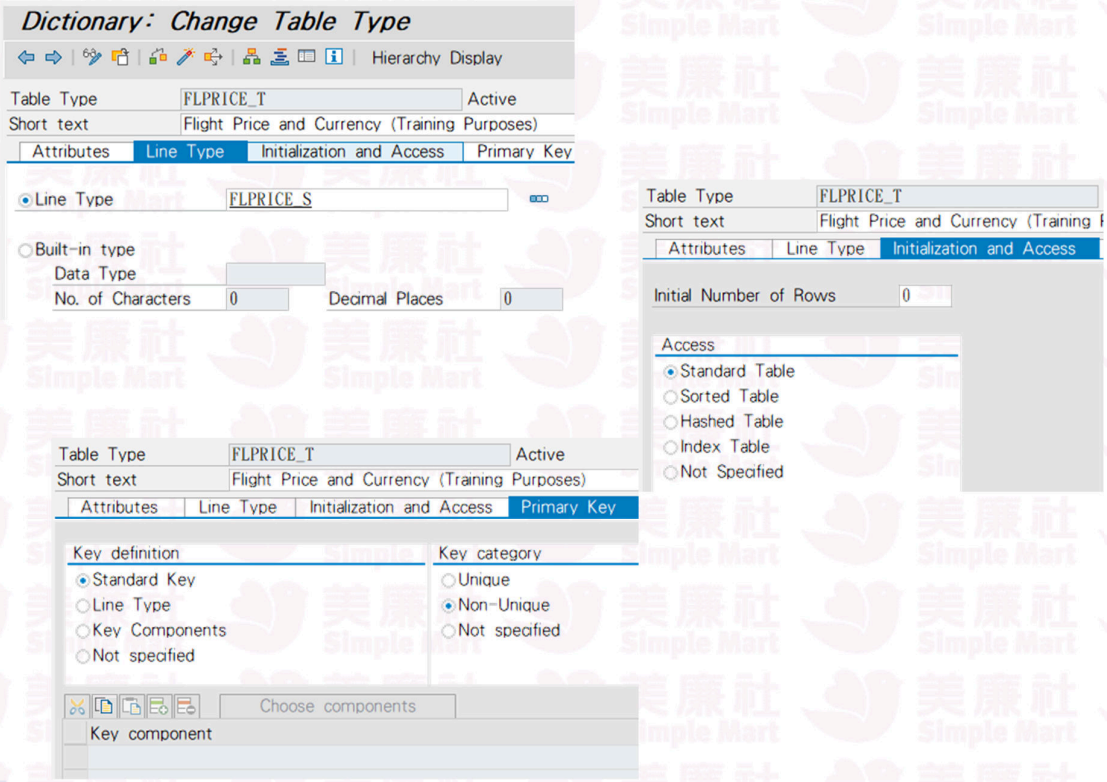
Defining Internal Tables with Global Types




```
DATA: itab_flprice TYPE flprice_t.
```

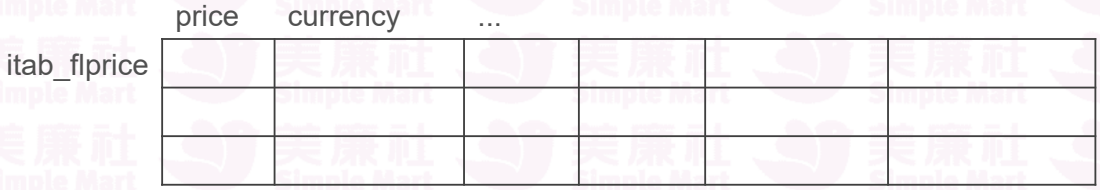
	price	currency	...			
itab_flprice						

Global table type in dictionary (SE11)



Defining Internal Tables with Local Types

```
TYPES "local table type
 tt_flprice TYPE STANDARD TABLE OF flprice_s
            WITH NON-UNIQUE KEY currency.

" internal table (itab)
DATA itab_flprice TYPE tt_flprice.
```



Independent Definition of Internal Tables

```
" Local structure type
TYPES: BEGIN OF st_type,
         carrid TYPE s_carr_id,
         connid TYPE s_conn_id,
       END OF st_type.

" itab
                SORTED
DATA itab TYPE STANDARD TABLE OF st_type
                HASHED
          WITH  UNIQUE       KEY ...
                NON-UNIQUE
```

```
DATA: BEGIN OF itab OCCURS 0,
        carrid TYPE s_carr_id,
        connid TYPE s_conn_id,
      END OF itab .
```

Possible Definitions of Internal Tables

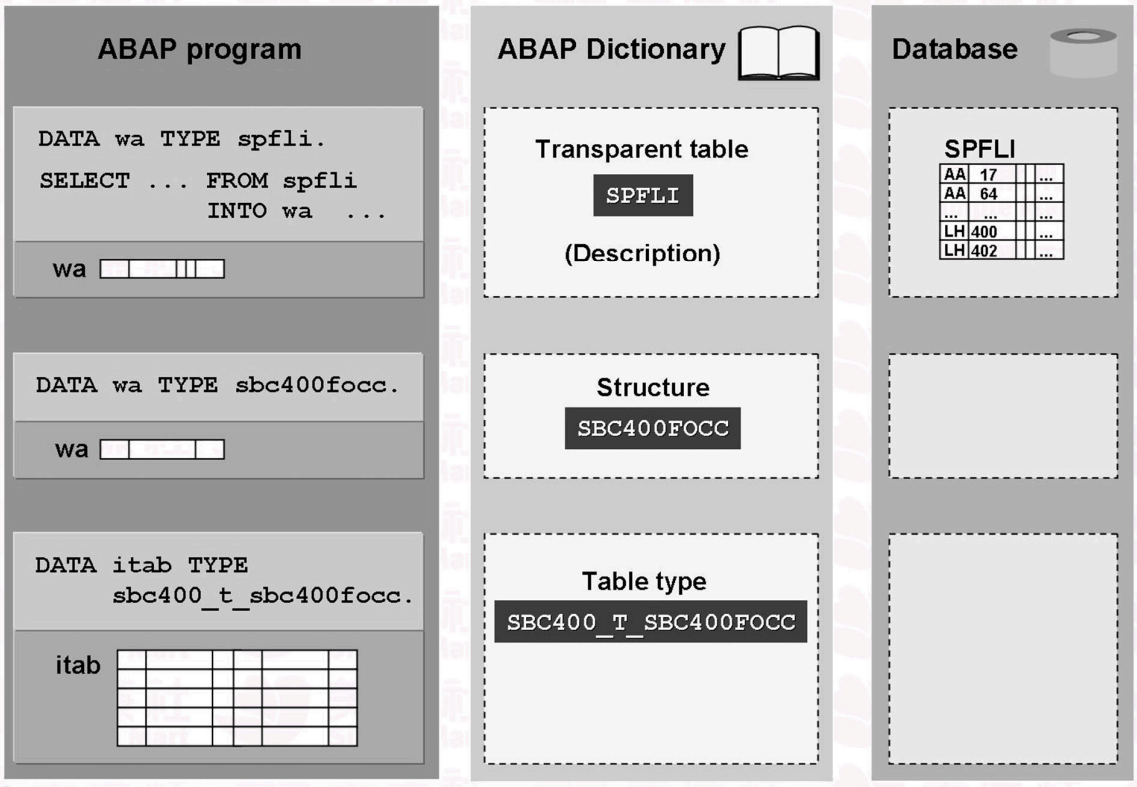
1. DATA itab TYPE <Table_type>.

2. DATA itab TYPE STANDARD / SORTED / HASHED TABLE OF <Structure_type> WITH … KEY. …

3. DATA itab TYPE TABLE OF <Structure_type>.
   - (short form for definition of a standard table with non-unique default key)

© 2018 NTT DATA Taiwan Co., Ltd.

Overview: Definition of Structured Data Objects



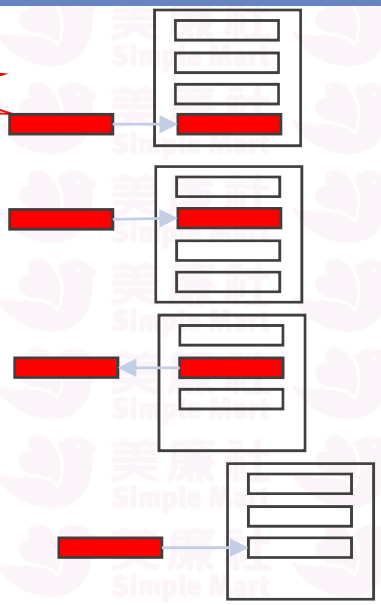
© 2018 NTT DATA Taiwan Co., Ltd.

Accessing Single Records (Overview)

Append	APPEND wa TO itab.
Insert	INSERT wa INTO TABLE itab <condition>.
Read	READ TABLE itab INTO wa <condition>.
Change	MODIFY TABLE itab FROM wa [<condition>].
Delete	DELETE itab <condition>.
Collect	COLLECT wa INTO itab.

Standard itab only

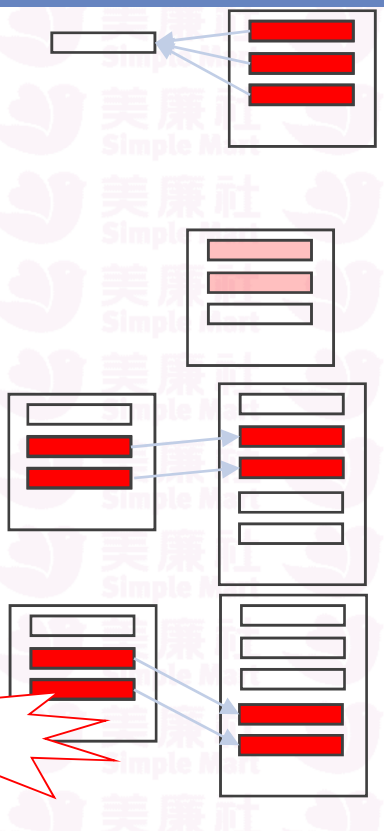
- Non-number type for key for check exist
- Number type for sum



Processing Sets of Records (Overview)

Record by record processing across the entries (or part) of the itab	LOOP AT itab INTO wa <condition>. … ENDLOOP.
Deleting several rows	DELETE itab <condition>.
Insert several rows from another itab	INSERT LINES OF itab1 <condition> INTO itab2 <condition>.
Appending several rows from another itab	APPEND LINES OF itab1 <condition> TO itab2.

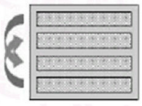
Standard itab only



Syntax Example: Sort, and Delete Content

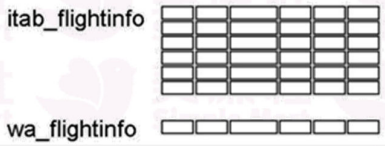
```
SORT itab. " by all fields

SORT itab BY carrid.  " ASCENDING

SORT itab BY percentage DESCENDING
             carrid ASCENDING.


REFRESH itab_flightinfo.

FREE itab_flightinfo.
```

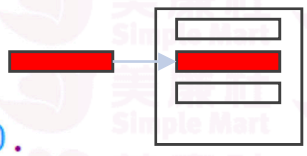
Standard itab only



Syntax Example: Insert Row

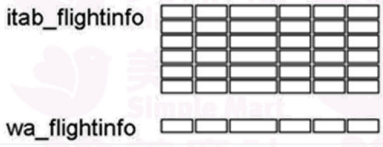
```
" Define itab and structure
DATA: itab_flights TYPE ty_flights,
      wa_flight    LIKE LINE OF itab_flights.


" fill structure with values
wa_flight-carrid = 'LH'.
wa_flight-connid = '0400'.
wa_flight-fldate = '20150101'.
wa_flight-seatsmax = 200.
wa_flight-seatsocc = 130.
wa_flight-percentage = 130 * 100 / 200.

" insert structure into itab
INSERT wa_flight INTO TABLE itab_flights.
```





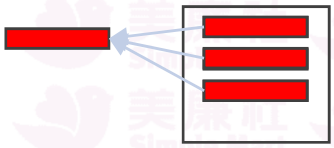
Syntax Example: Output of Internal Table Row-by-Row

```
" Define itab and structure
DATA: itab_flights TYPE ty_flights,
      wa_flight    LIKE LINE OF itab_flights.
```



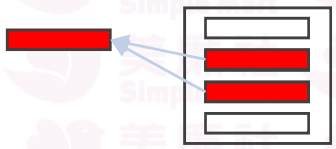
sy-tabix

```
LOOP AT itab_flights INTO wa_flight.
  WRITE:/ wa_flight-carrid,
          wa_flight-connid,
          wa_flight-fldate,
          wa_flight-seatsmax,
          wa_flight-seatsocc,
          wa_flight-percentage,
          '%'.
ENDLOOP.
```



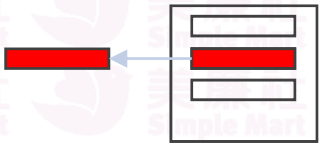
Syntax Example: Reading Using the Index

```
LOOP AT itab_flights INTO wa_flight
                     FROM 3 TO 8.
  WRITE:/ wa_flight-carrid,
          wa_flight-connid,
          wa_flight-fldate,
          wa_flight-seatsmax,
          wa_flight-seatsocc,
          wa_flight-percentage,
          '%'.
ENDLOOP.
```



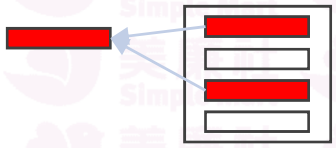
sy-tabix

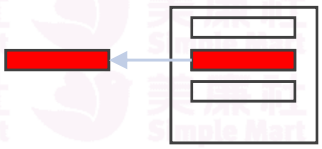
```
READ TABLE itab_flights INTO wa_flight
                        INDEX 7.
IF sy-subrc = 0.
  WRITE:/ wa_flight-carrid,
          wa_flight-connid,
          wa_flight-fldate,
          wa_flight-seatsmax,
          wa_flight-seatsocc,
          wa_flight-percentage,
          '%'.
ENDIF.
```



Syntax Example: Reading Using the Key

```
LOOP AT itab_flights INTO wa_flight
                     WHERE carrid = 'LH'
                       AND connid = '0400'.
  WRITE:/ wa_flight-carrid,
          wa_flight-connid,
          wa_flight-fldate,
          wa_flight-seatsmax,
          wa_flight-seatsocc,
          wa_flight-percentage,
          '%'.
ENDLOOP.

READ TABLE itab_flights INTO wa_flight
   WITH TABLE KEY " WITH KEY
    carrid = 'LH'      connid = '0400'
    fldate = sy-datum.
IF sy-subrc = 0.
  WRITE:/ wa_flight-carrid,
          wa_flight-connid,
          wa_flight-fldate,
          wa_flight-seatsmax,
          wa_flight-seatsocc,
          wa_flight-percentage,
          '%'.
ENDIF.
```

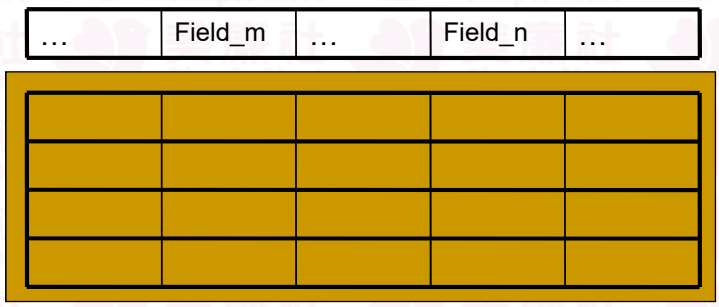




© 2018 NTT DATA Taiwan Co., Ltd.

Itab – loop event : AT

```
LOOP AT itab INTO wa.
  " Statement
  AT FIRST.
    " Statement
  ENDAT.
  " Statement
    AT NEW field_m.
      " Statement
    ENDAT.
  " Statement
      AT NEW field_n.
        " Statement
      ENDAT.
  " Statement
      AT END OF field_n.
        " Statement
      ENDAT.
  " Statement
    AT END OF field_m.
      " Statement
    ENDAT.
  " Statement
  AT LAST.
    " Statement
  ENDAT.
  " Statement
ENDLOOP.
```



© 2018 NTT DATA Taiwan Co., Ltd.

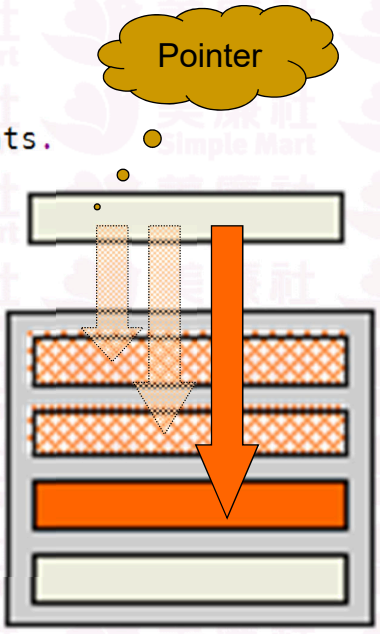
Loop event: AT (Continue)

Field X	Field Y	Field Z	FIRST	NEW			END			LAST
				X	Y	Z	X	Y	Z	
A	A	A								
A	A	B								
A	B	B								
A	B	C								
B	C	E								
B	C	F								
B	D	F								

© 2018 NTT DATA Taiwan Co., Ltd.

Itab – access (loop...endloop) with Field-Symbols

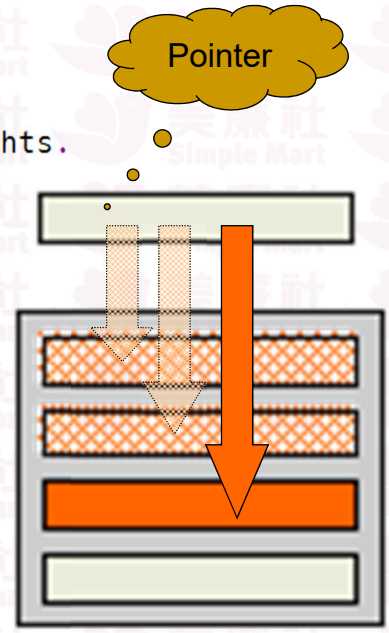
```
DATA itab_flights TYPE ty_flights.
FIELD-SYMBOLS <flight> LIKE LINE OF itab_flights.

LOOP AT itab_flights ASSIGNING <flight>
                     WHERE carrid = 'LH'
                       AND connid = '0400'.
  WRITE:/ <flight>-carrid,
          <flight>-connid,
          <flight>-fldate,
          <flight>-seatsmax,
          <flight>-seatsocc,
          <flight>-percentage,
          '%'.
ENDLOOP.
```



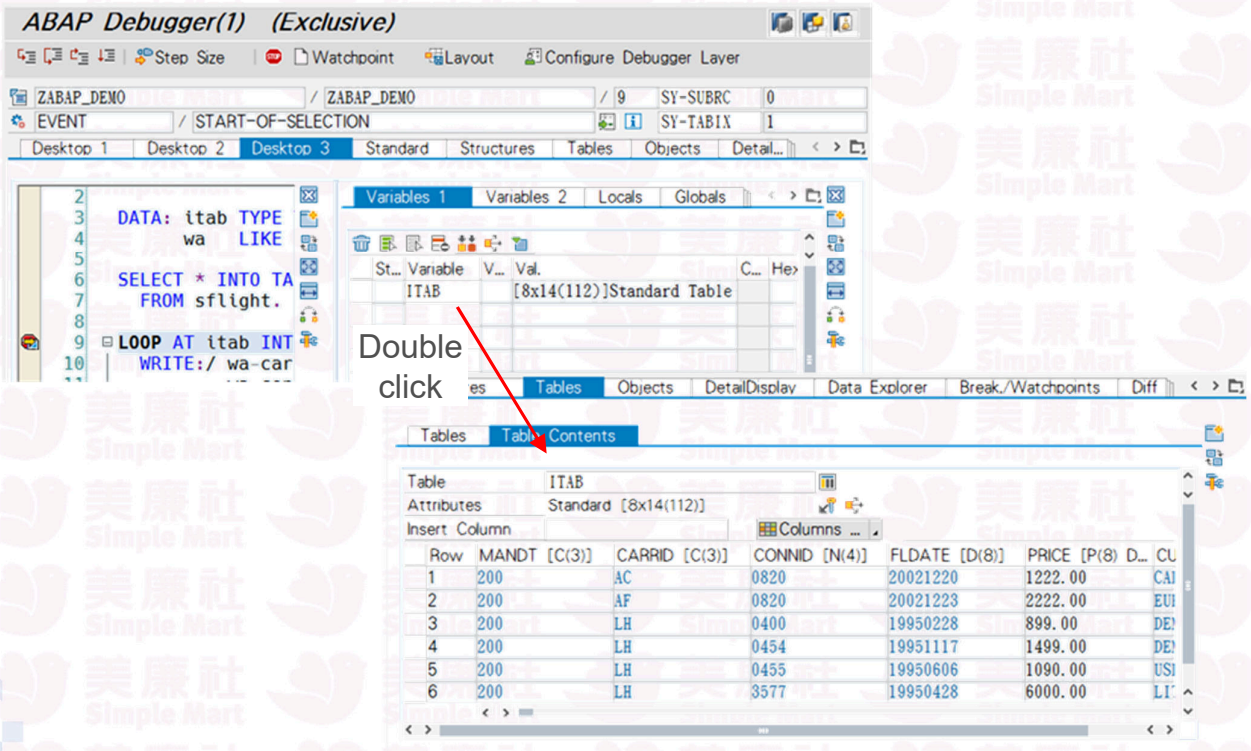
© 2018 NTT DATA Taiwan Co., Ltd.

Itab – access (read table ...) with Field-Symbols

```
DATA itab_flights TYPE ty_flights.
FIELD-SYMBOLS <flight> LIKE LINE OF itab_flights.

READ TABLE itab_flights ASSIGNING <flight>
  WITH TABLE KEY carrid = 'LH'
                 connid = '0400'
                 fldate = sy-datum.
IF sy-subrc = 0.
  WRITE:/ <flight>-carrid,
          <flight>-connid,
          <flight>-fldate,
          <flight>-seatsmax,
          <flight>-seatsocc,
          <flight>-percentage,
          '%'.
ENDIF.
```



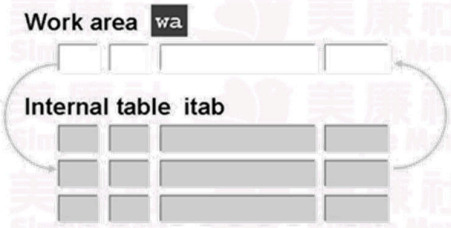
Internal Tables in Debugging Mode (SAP GUI)



Comparison: itab With and Without Header Lines

Explicit syntax

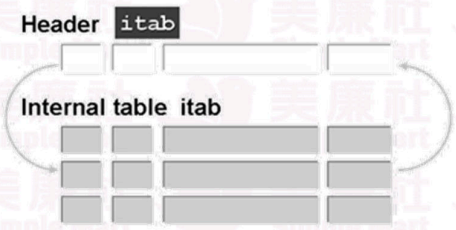
```
DATA: itab TYPE TABLE OF sflight,
      wa   LIKE LINE OF itab.
```




```
INSERT wa INTO itab INDEX n.
DELETE TABLE itab FROM wa.
READ TABLE itab INTO wa INDEX n.
WRITE wa-carrid.
```

Implicit syntax

```
DATA itab TYPE TABLE OF sflight
     WITH HEADER LINE.
```

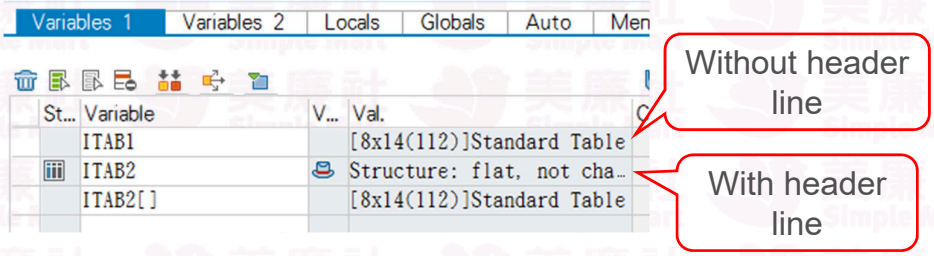



```
INSERT itab INDEX n.
DELETE TABLE itab.
READ TABLE itab INDEX n.
WRITE itab-carrid.
```

```
DATA: BEGIN OF itab OCCURS 0,
        carrid TYPE s_carr_id,
        connid TYPE s_conn_id,
      END OF itab .
```

© 2018 NTT DATA Taiwan Co., Ltd.

itab With and Without Header Lines in Debug Mode



© 2018 NTT DATA Taiwan Co., Ltd.

New Expressions for itab - insert

Before 740	740

`wa-field1 = 'A'.`
`wa-field2 = 'B'.`
`APPEND wa TO itab.`
	`APPEND VALUE #( field1 = 'C'`
	`field2 = 'D' )`
	`TO itab.`

`REFRESH itab.`
`wa-field1 = 'M'.`
`wa-field2 = 'B'.`
`APPEND wa TO itab.`
`wa-field1 = 'M'.`
`wa-field2 = 'C'.`
`APPEND wa TO itab.`
	`itab`
	`= VALUE #( field1 = 'M'`
	`( field2 = 'B' )`
	`( field2 = 'C' ) ).`

New Expressions for itab - read

Before 740	740

`READ TABLE itab INDEX 1`
`INTO wa.`
`WRITE wa-field.`
	`WRITE itab[ 1 ]-field.`

`READ TABLE itab`
`WITH KEY field1 = 'A'`
`INTO wa.`
`IF sy-subrc = 0.`
`WRITE wa-field2.`
`ENDIF.`
	`IF line_exists( itab[`
	`field1 = 'A' ] ).`
	`WRITE itab[ field1`
	`= 'A' ]-field2.`
	`ENDIF.`

`READ TABLE itab`
`WITH KEY field1 = 'A'`
`field2 = 'B'`
`ASSIGNING <fs>.`
	`ASSIGN itab[ field1 = 'A'`
	`field2 = 'B' ]`
	`TO <fs>.`

Test - 1

變數宣告	Err	Single	Stru	Itab
Data a1 type domain				
Data a2 like domain				
Data a3 type data-element				
Data a4 like data-element				
Data a5 like a3				
Data a6 type a3				
Data a7 type dictionary-structure				
Data a8 like dictionary-structure				
Data a9 like dictionary-table				
Data a10 like a7				

© 2018 NTT DATA Taiwan Co., Ltd.

Test - 2

變數宣告	Err	Single	Stru	Itab
Data a11 type table-type				
Data a12 like table-type				
Data a13 type table of dictionary-stru				
Data a14 like table of dictionary-stru				
Data a15 type table of data-element				
Data a16 type table of table-type				
Data a17 like line of a11				
Data a18 like line of a13				
Data a19 like line of a17				
TABLES dictionary-table				
TABLES dictionary-stru				

© 2018 NTT DATA Taiwan Co., Ltd.

Basic ABAP Language Elements

- ✓Working with Elementary Data Objects
- ✓Working with Structure
- ✓Working with Internal Tables
- ✓Inline declaration

© 2018 NTT DATA Taiwan Co., Ltd.

Inline Declaration (NW740)

```
DATA(a1) = 123.    " DATA a1 TYPE i.
DATA(a2) = '123'.  " DATA a2 TYPE c LENGTH 3.

  TYPES t_itab TYPE TABLE OF i
    WITH NON-UNIQUE KEY table_line.

  DATA(itab)
    = VALUE t_itab( ( 1 ) ( 2 ) ( 3 ) ).

  LOOP AT itab INTO DATA(wa).
    ...
  ENDLOOP.

  READ TABLE itab INDEX 1
    ASSIGNING FIELD-SYMBOL(<fs>).

  ASSIGN itab[ 1 ] TO FIELD-SYMBOL(<fs>).
```

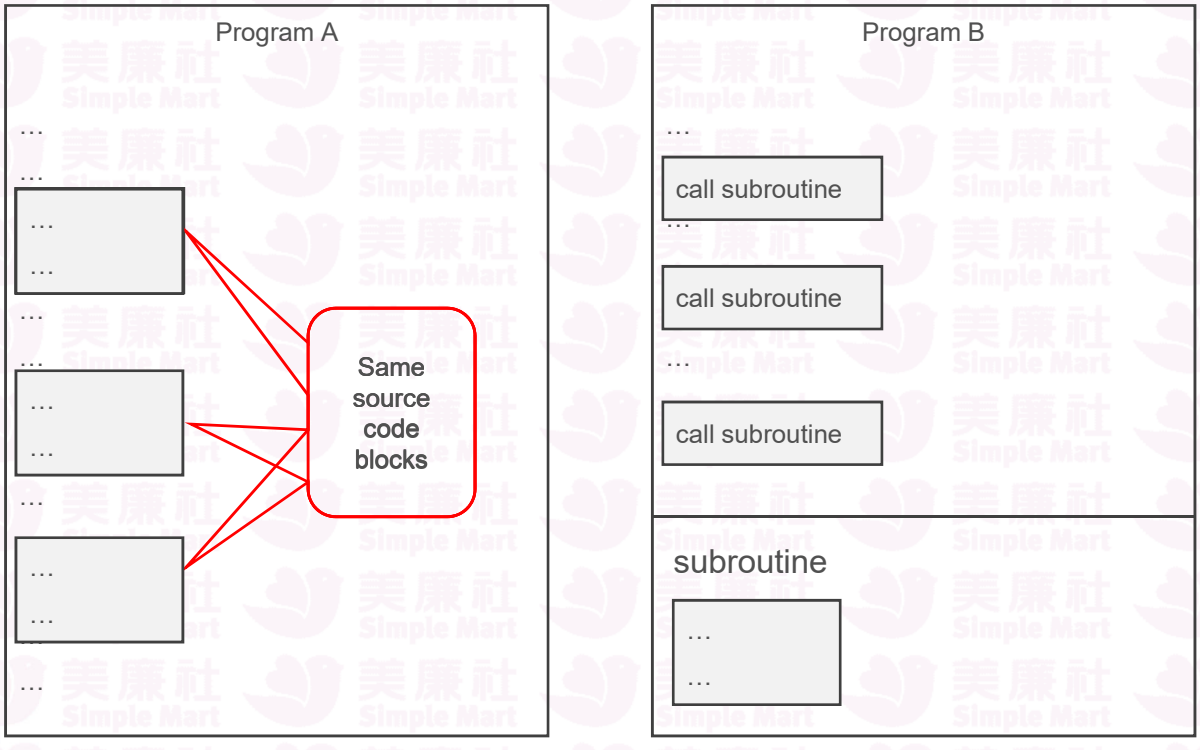
© 2018 NTT DATA Taiwan Co., Ltd.

Modularization

- ✓Basic and Overview
- ✓Subroutine
- ✓Function Group / Function Module and BAPI

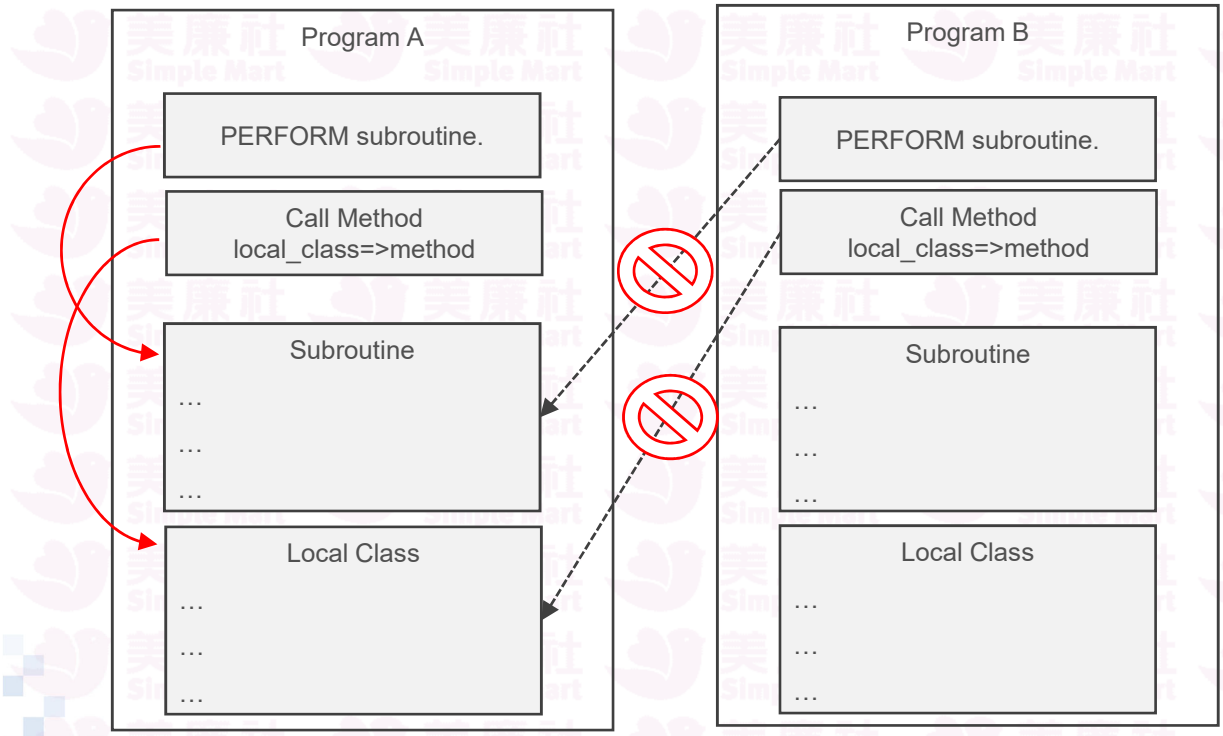
© 2018 NTT DATA Taiwan Co., Ltd.

Using Subroutines (Motivation)



© 2018 NTT DATA Taiwan Co., Ltd.

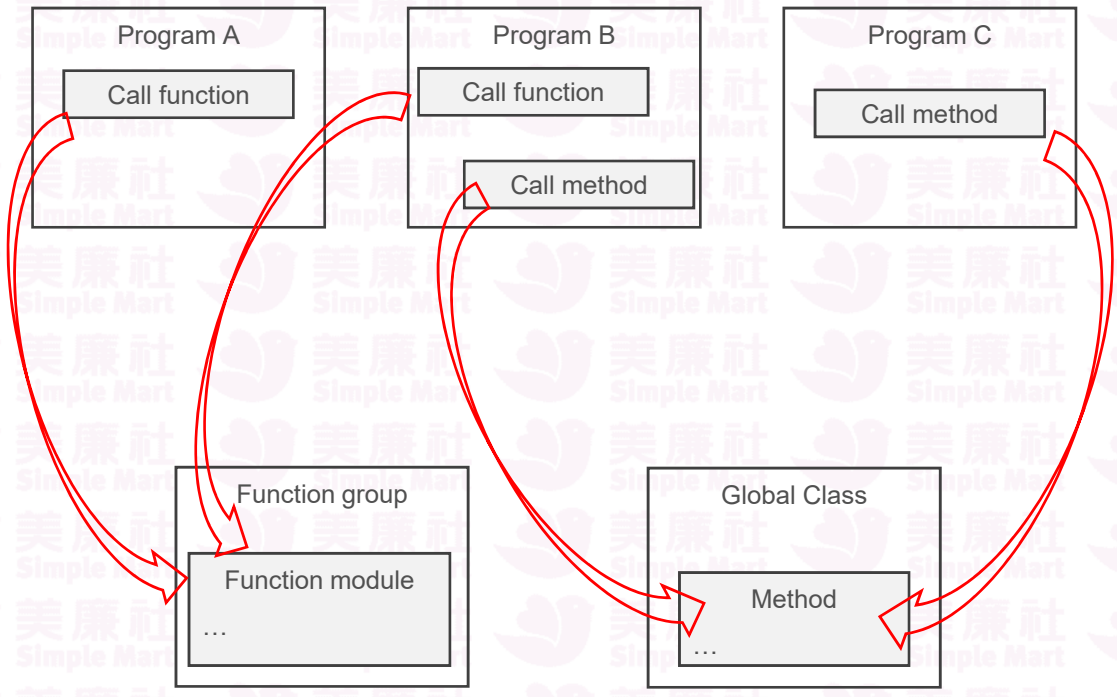
Local Program Modularization



© 2018 NTT DATA Taiwan Co., Ltd.

NTT DATA

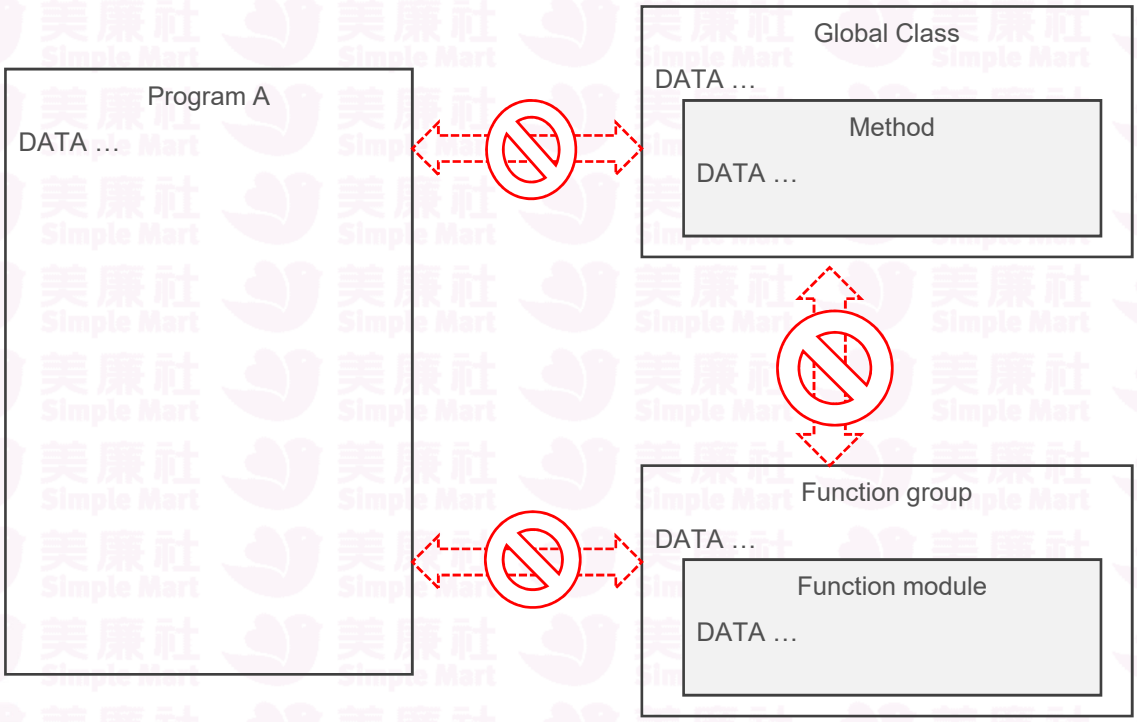
Global Modularization



© 2018 NTT DATA Taiwan Co., Ltd.

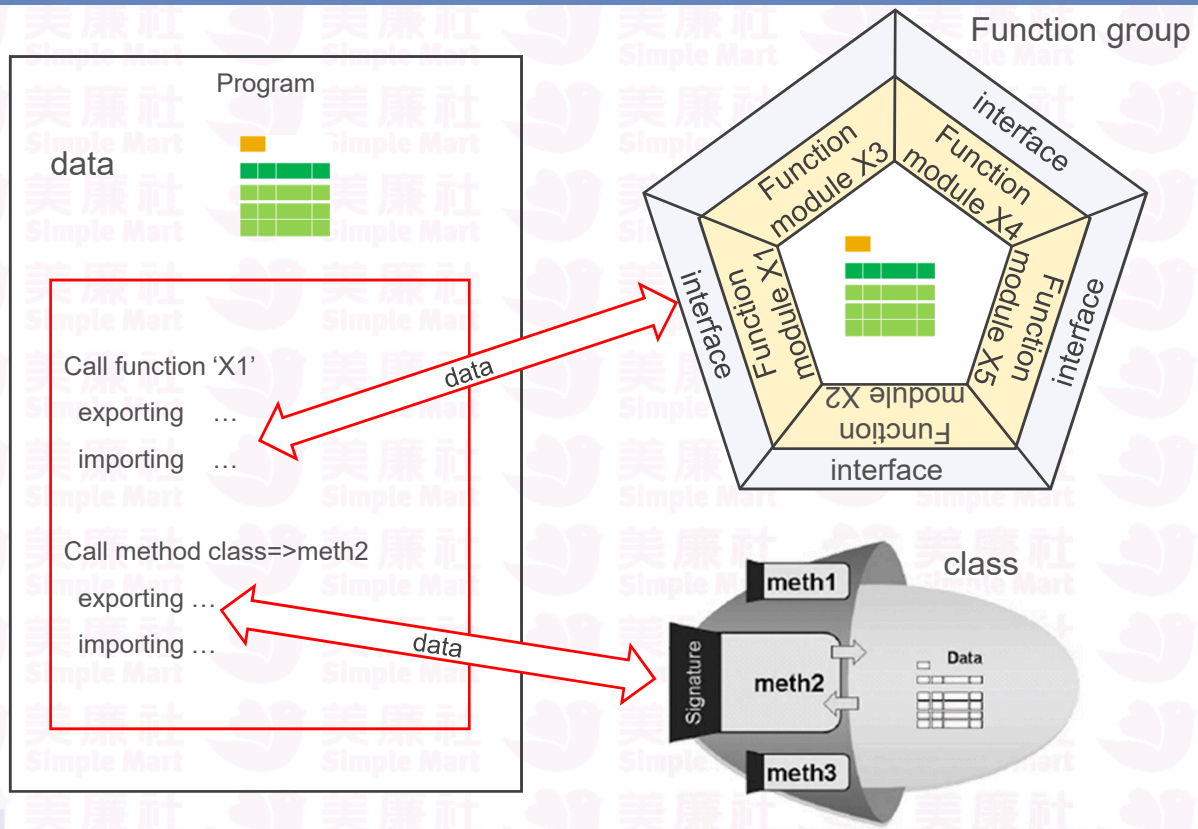
NTT DATA

Separating Data / Data Encapsulation



© 2018 NTT DATA Taiwan Co., Ltd.

Data Transports Between the Program & Modularization Unit



© 2018 NTT DATA Taiwan Co., Ltd.

Modularization

- ✓Basic and Overview
- ✓Subroutine
- ✓Function Group / Function Module and BAPI

© 2018 NTT DATA Taiwan Co., Ltd.

Defining and Calling Subroutines

```
DATA: v1 TYPE c, v2 TYPE c, v3 TYPE c.

PERFORM my_form USING v1
                      CHANGING v2
                               v3.

FORM my_form
     USING VALUE(f1)   " Call by Value
  CHANGING VALUE(f2)   " Call by value and result
           f3.         " Call by reference
  ...
ENDFORM.
```

Call

Definition

```
FORM my_form
    TABLES itab       " Call by Reference
     USING VALUE(f1)  " Call by Value
           f2         " Call by Reference
  CHANGING VALUE(f3)  " Call by Value and Result
           f4.        " Call by Reference

ENDFORM.
```

© 2018 NTT DATA Taiwan Co., Ltd.

Typing the Interface Parameters

```
TYPES t_perc TYPE p DECIMALS 2.

DATA: v1 TYPE i, v2 TYPE i, v3 TYPE t_perc.

PERFORM calc_perc USING v1 v2 CHANGING v3.
```

Generic type assignment

Inherited type + danger of type conflict

```
FORM calc_perc
  USING
    VALUE(f_part) " TYPE any
    VALUE(f_all)  " TYPE a
  CHANGING
    VALUE(f_perc). " TYPE any
  CHECK f_all <> 0.
  f_perc = f_part * 100 / f_all.
ENDFORM.
```

Exact type assignment

Default type for actual parameter

```
FORM calc_perc
  USING
    VALUE(f_part) TYPE i
    VALUE(f_all)  TYPE i
  CHANGING
    VALUE(f_perc) TYPE t_perc.
  CHECK f_all <> 0.
  f_perc = f_part * 100 / f_all.
ENDFORM.
```


Typing the Interface Parameter for Structures and Internal Tables

```
DATA wa_flight TYPE sbc400focc.
...
PERFORM fill_wa
  CHANGING wa_flight.
...
FORM fill_wa
  CHANGING f_wa TYPE sbc400focc.

  f_wa-carrid = 'AA'.
  f_wa-connid = '0400'.

ENDFORM.
```

Structure type

```
DATA itab_flight
  TYPE sbc400_t_sbc400focc.
...
PERFORM fill_itab
  CHANGING itab_flight.
...

FORM fill_itab
  CHANGING f_itab
    TYPE sbc400_t_sbc400focc.

  LOOP AT f_itab ... .
    ...
  ENDLOOP.

ENDFORM.
```

Table type

Visibility of Global and Local Data Objects

```
TYPES t_perc TYPE p DECIMALS 2.

DATA: g_int1 TYPE i,
      g_int2 TYPE i,
      g_result TYPE t_perc.

...
PERFORM calc_perc USING g_int1 g_int2
                  CHANGING g_result.
...

FORM calc_perc USING p_part TYPE i
                     p_all  TYPE i
               CHANGING c_pc   TYPE t_perc.
  DATA: l_pc TYPE p LENGTH 16 DECIMALS 2.
  ...
ENDFORM.
```

Global variables

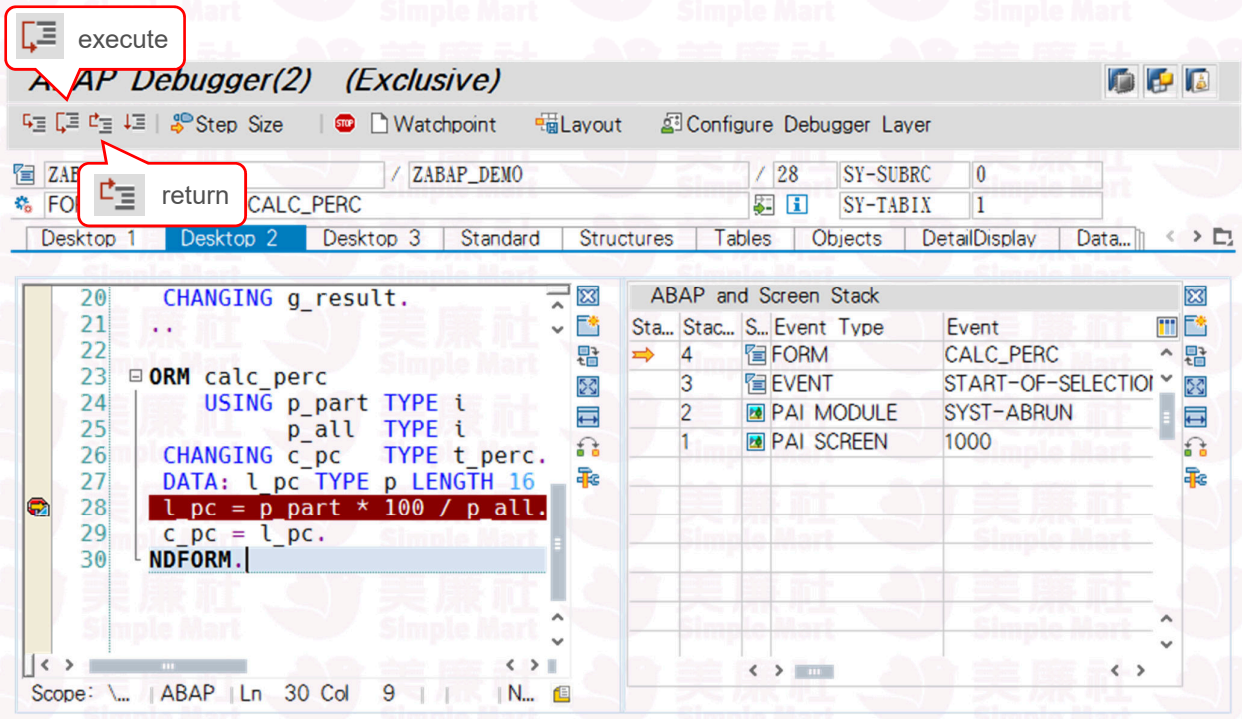
Formal variables

Local variables

Syntax Example: Passing an Internal Table

```
DATA: itab_flight TYPE sbc400_t_sbc400focc,
      carrid TYPE s_carr_id,
      lines TYPE i.
...
PERFORM count_lines USING carrid itab_flight
                    CHANGING lines.
...

FORM count_lines
  USING f_carr TYPE s_carr_id
        f_itab TYPE sbc400_t_sbc400focc
  CHANGING value(f_lines) TYPE i.

  DATA l_wa LIKE LINE OF f_itab.

  CLEAR f_lines.
  LOOP AT f_itab INTO l_wa WHERE carrid = f_carr.
    f_lines = f_lines + 1.
  ENDLOOP.
ENDFORM.
```

Subroutines in Debugging Mode



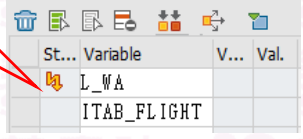
Subroutines in the Debugging Mode

Single step		The program will be executed one step at a time. After PERFORM, the first line of the subroutine will be displayed.
Execute		The subroutine will executed as a whole. After PERFORM, the next line of the calling program will be displayed.
Return		The subroutine will executed until the ENDFORM statement. The next line of the calling program, after PERFORM statement, will be displayed

Calling program

Subroutine

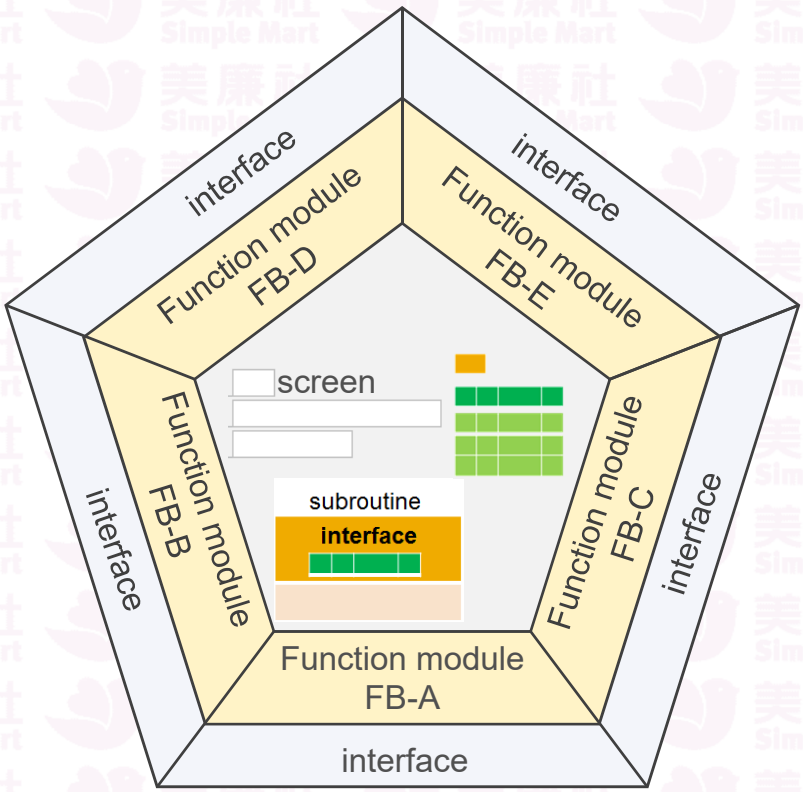
Display in debugger in the main program: local data object or interface parameters not in the subroutine



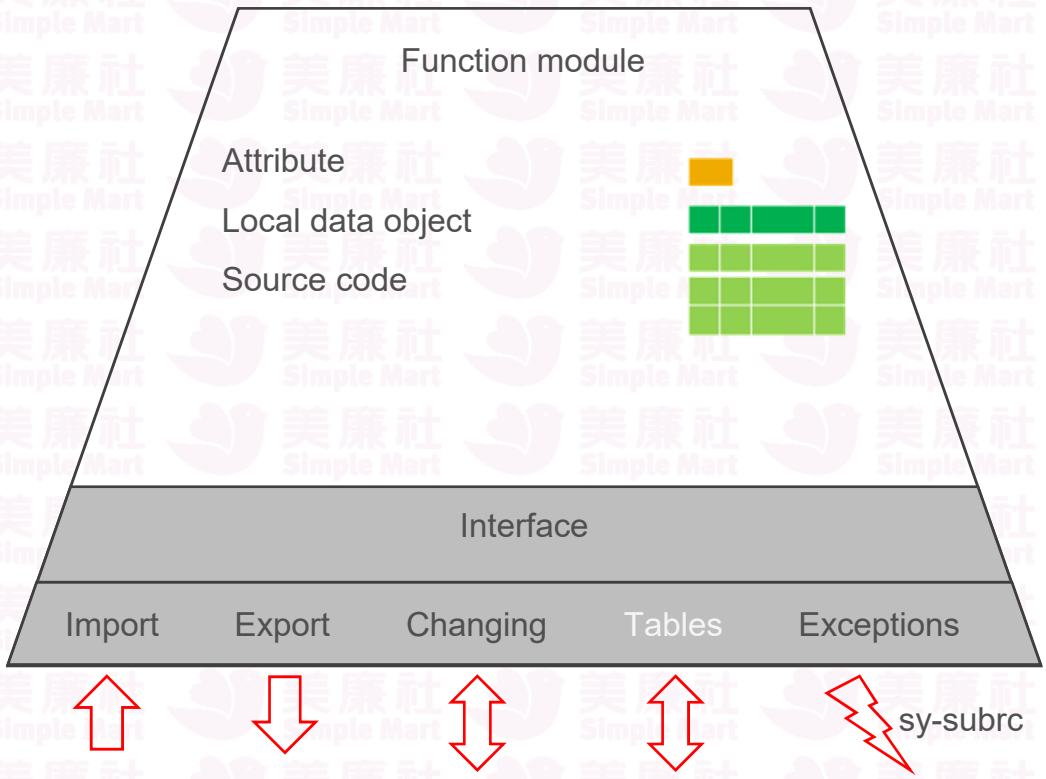
Modularization

- ✓Basic and Overview
- ✓Subroutine
- ✓Function Group / Function Module and BAPI

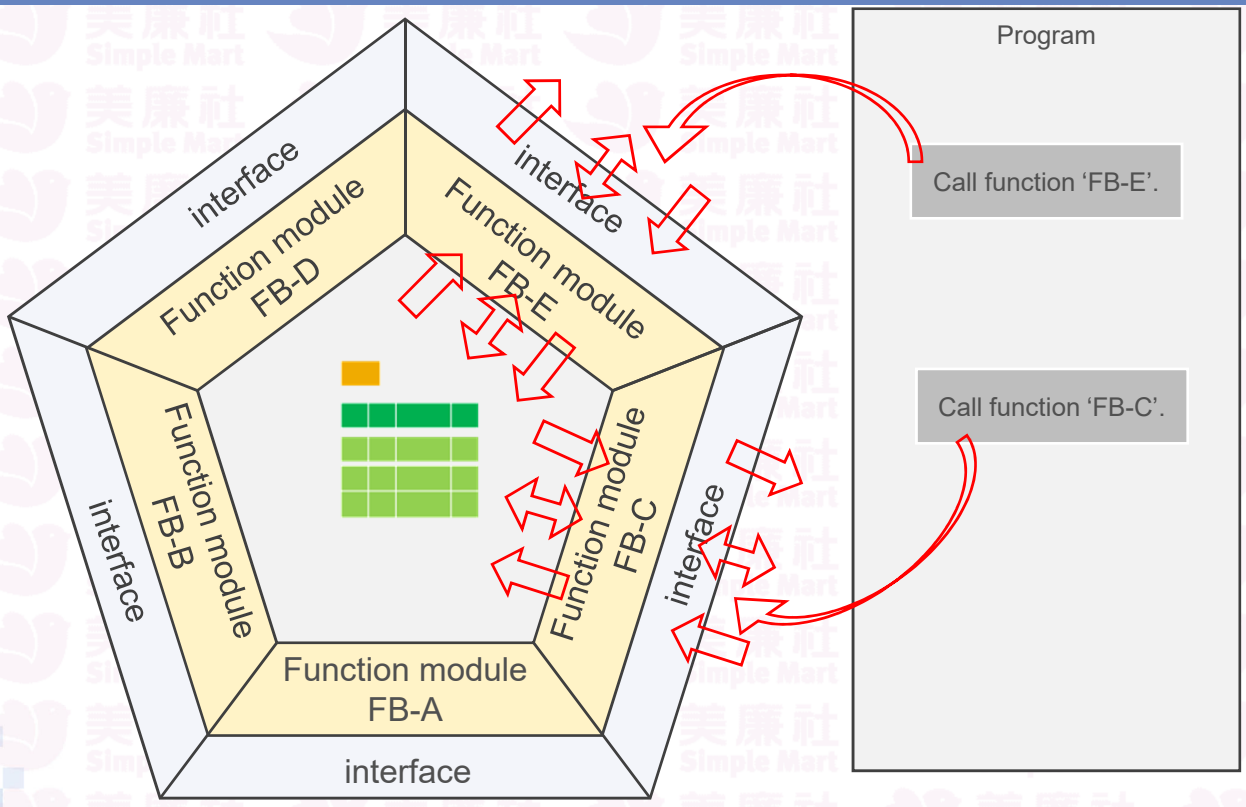
Function group



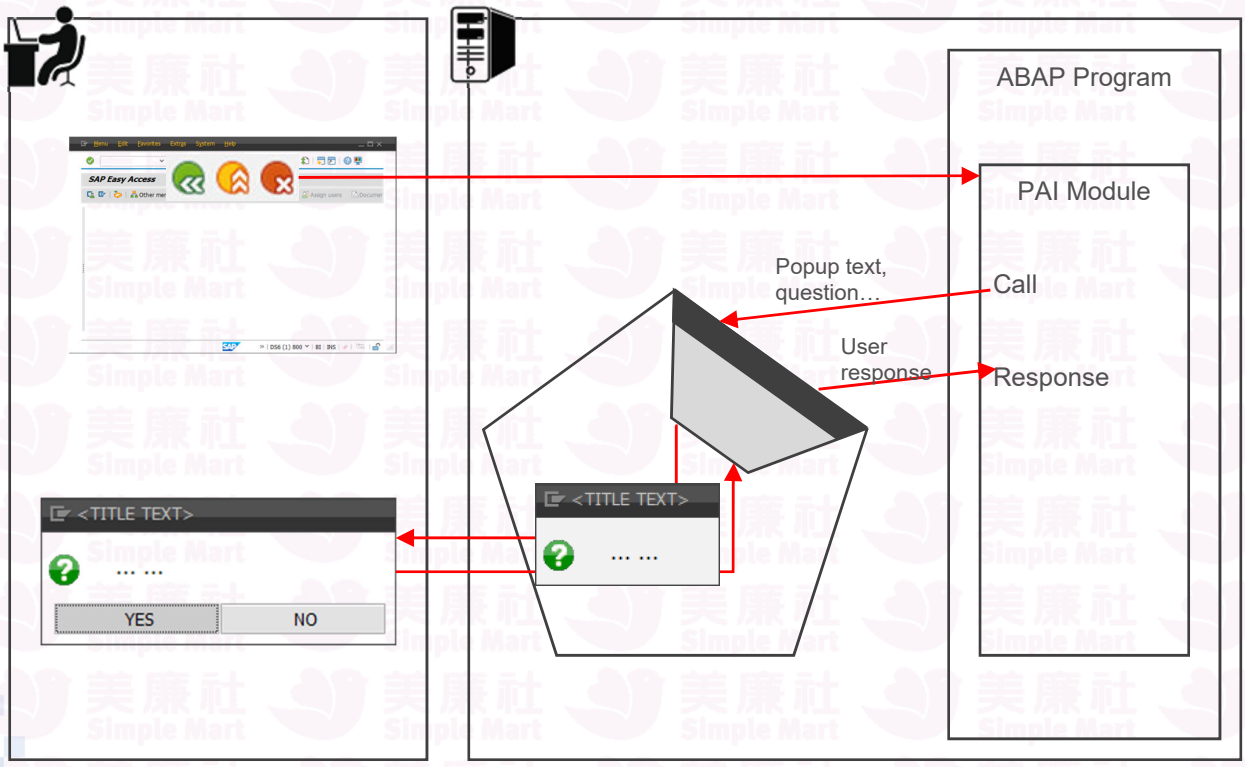
Function module



Data Objects Within a Function Group



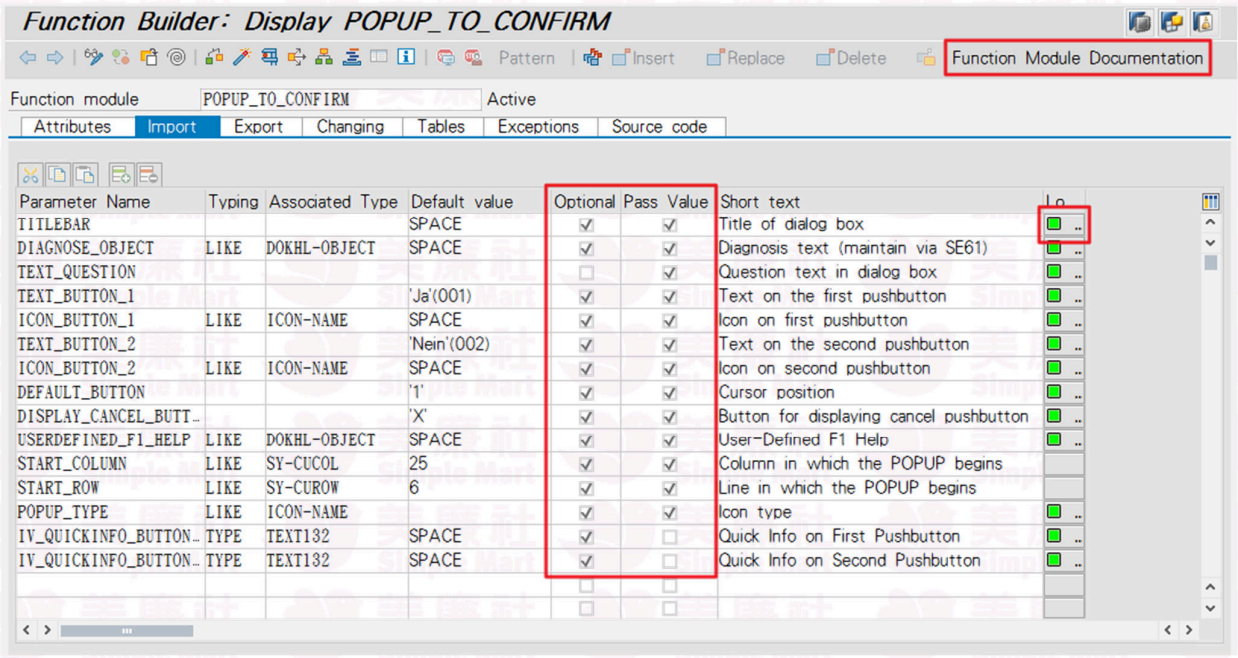
Example of a Function Module with User Dialog



© 2018 NTT DATA Taiwan Co., Ltd.

NTT DATA

T-Code : SE37
Interface of a Function Module



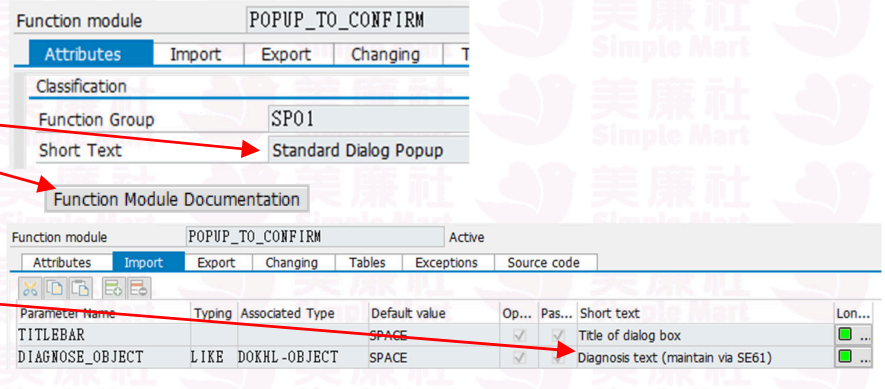
Parameter Name	Typing	Associated Type	Default value	Optional	Pass Value	Short text
TITLEBAR			SPACE	☑	☑	Title of dialog box
DIAGNOSE_OBJECT	LIKE	DOKHL-OBJECT	SPACE	☑	☑	Diagnosis text (maintain via SE61)
TEXT_QUESTION				☐	☑	Question text in dialog box
TEXT_BUTTON_1			'Ja'(001)	☑	☑	Text on the first pushbutton
ICON_BUTTON_1	LIKE	ICON-NAME	SPACE	☑	☑	Icon on first pushbutton
TEXT_BUTTON_2			'Nein'(002)	☑	☑	Text on the second pushbutton
ICON_BUTTON_2	LIKE	ICON-NAME	SPACE	☑	☑	Icon on second pushbutton
DEFAULT_BUTTON			'1'	☑	☑	Cursor position
DISPLAY_CANCEL_BUTT…			'X'	☑	☑	Button for displaying cancel pushbutton
USERDEFINED_F1_HELP	LIKE	DOKHL-OBJECT	SPACE	☑	☑	User-Defined F1 Help
START_COLUMN	LIKE	SY-CUCOL	25	☑	☑	Column in which the POPUP begins
START_ROW	LIKE	SY-CUROW	6	☑	☑	Line in which the POPUP begins
POPUP_TYPE	LIKE	ICON-NAME		☑	☑	Icon type
IV_QUICKINFO_BUTTON…	TYPE	TEXT132	SPACE	☑	☐	Quick Info on First Pushbutton
IV_QUICKINFO_BUTTON…	TYPE	TEXT132	SPACE	☑	☐	Quick Info on Second Pushbutton

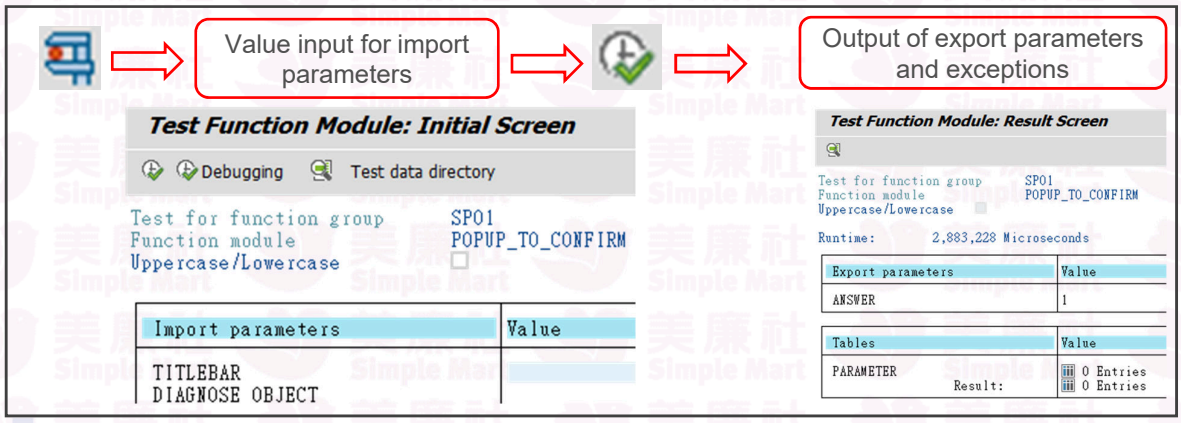
© 2018 NTT DATA Taiwan Co., Ltd.

NTT DATA

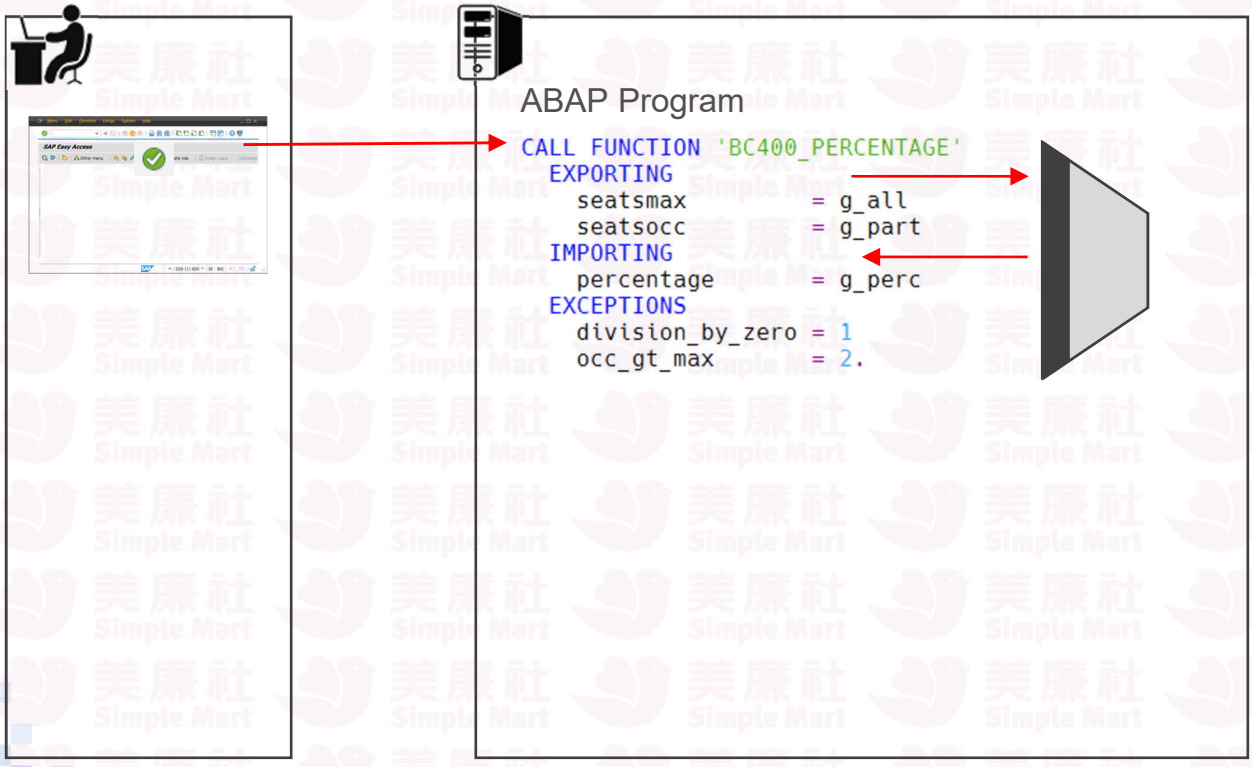
Documentation and Test Environment

Document
- Short text
- Function description
- Example of use
- Notes
- Parameter description
- Exception description

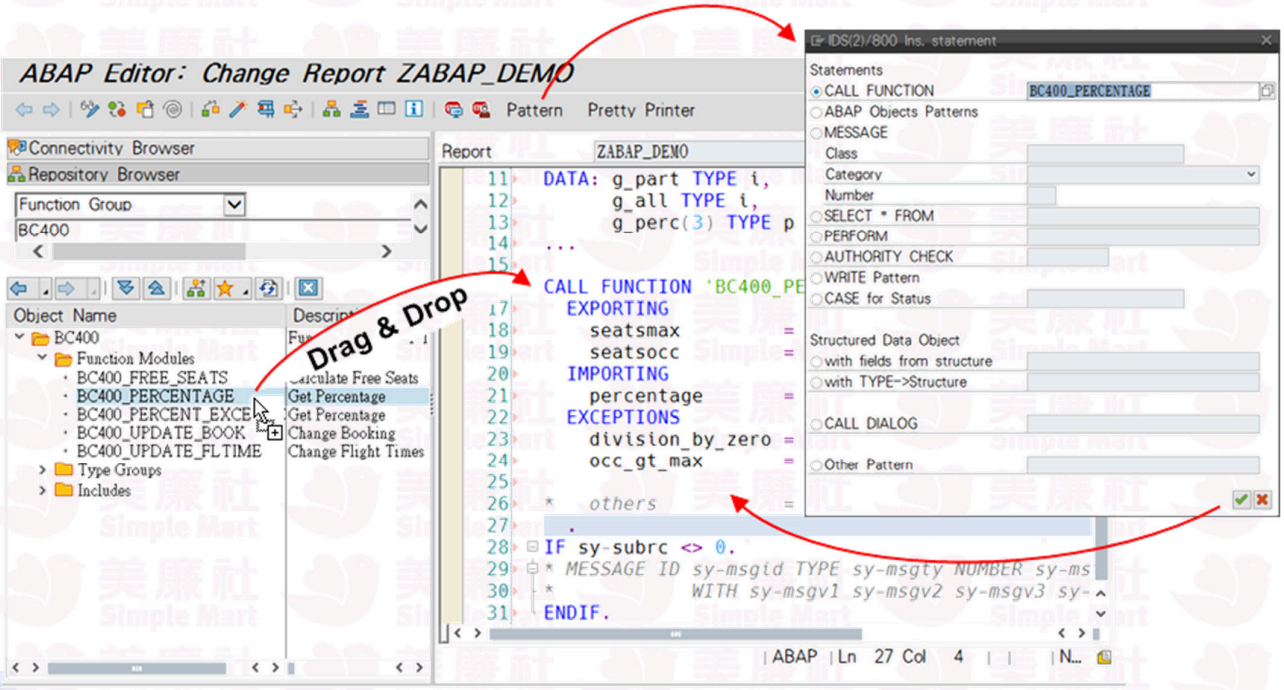




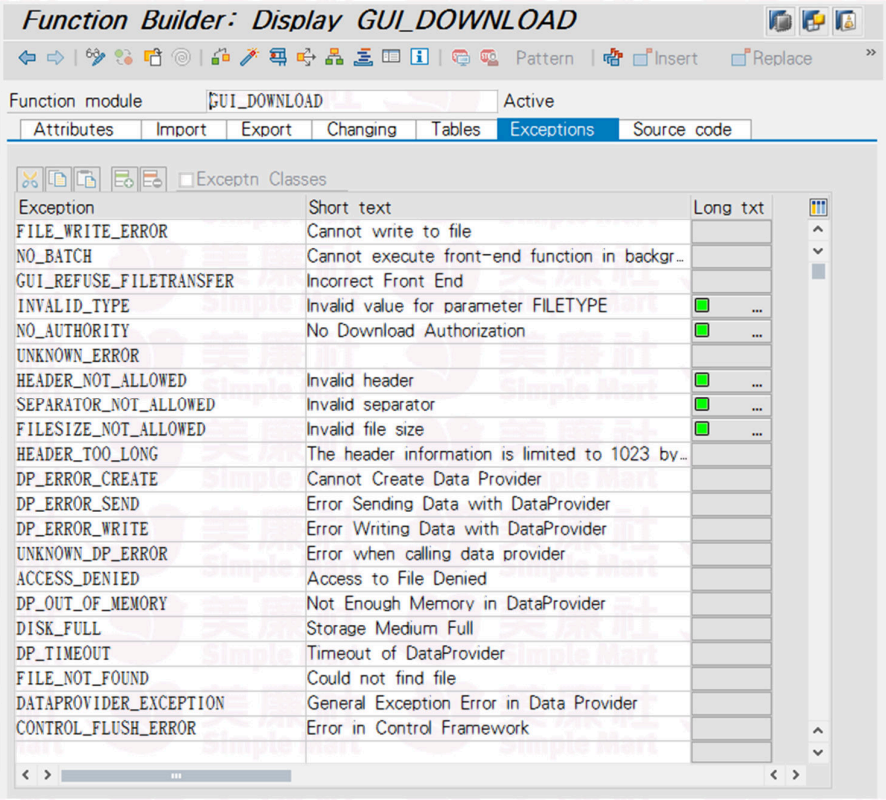
Calling a Function Module



Source Code Generation for Function Module Call



Exceptions

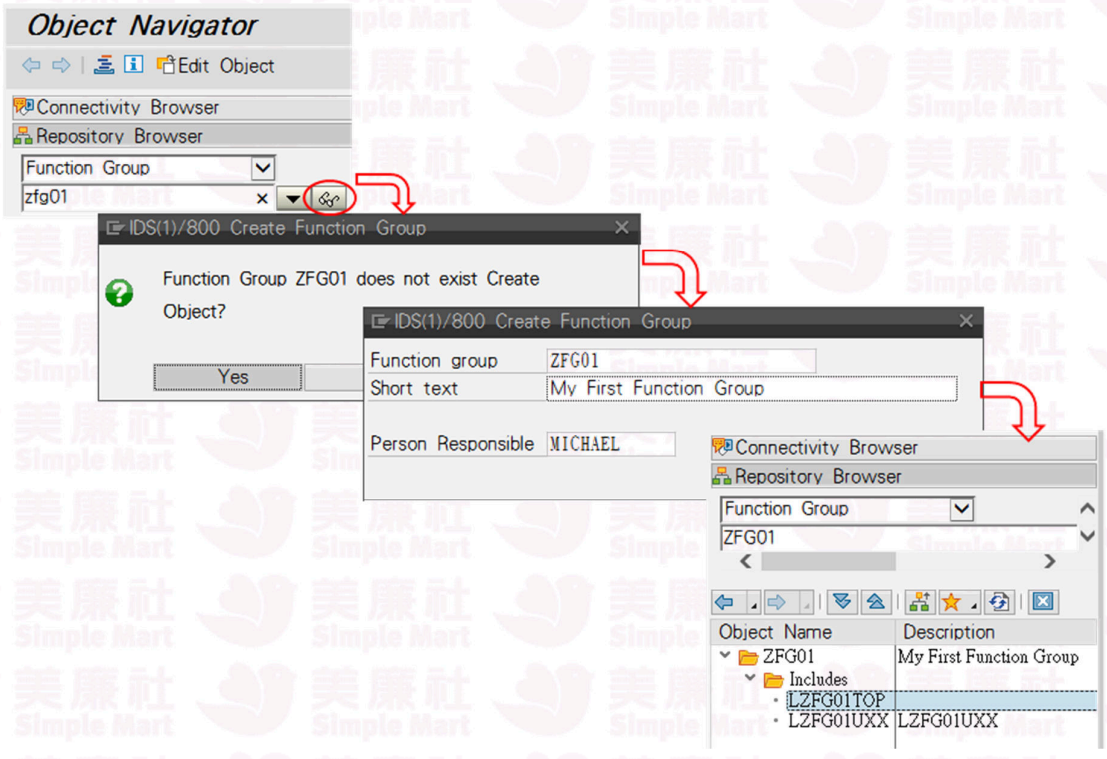


Handling Exceptions

```
CALL FUNCTION 'GUI_DOWNLOAD'
  EXPORTING
    filename                = 
  TABLES
    data_tab                = 
  EXCEPTIONS
    file_write_error        = 1
    no_batch                = 2
    gui_refuse_filetransfer = 3
    invalid_type            = 4
    no_authority            = 5
    unknown_error           = 6
    header_not_allowed      = 7
    separator_not_allowed   = 8
    filesize_not_allowed    = 9
    header_too_long         = 10
    dp_error_create         = 11
    dp_error_send           = 12
    dp_error_write          = 13
    unknown_dp_error        = 14
    access_denied           = 15
    dp_out_of_memory        = 16
    disk_full               = 17
    dp_timeout              = 18
    file_not_found          = 19
    dataprovider_exception  = 20
    control_flush_error     = 21
    OTHERS                  = 22.

CASE sy-subrc.
  WHEN 1.
  WHEN 2.
  WHEN OTHERS.
ENDCASE.
```

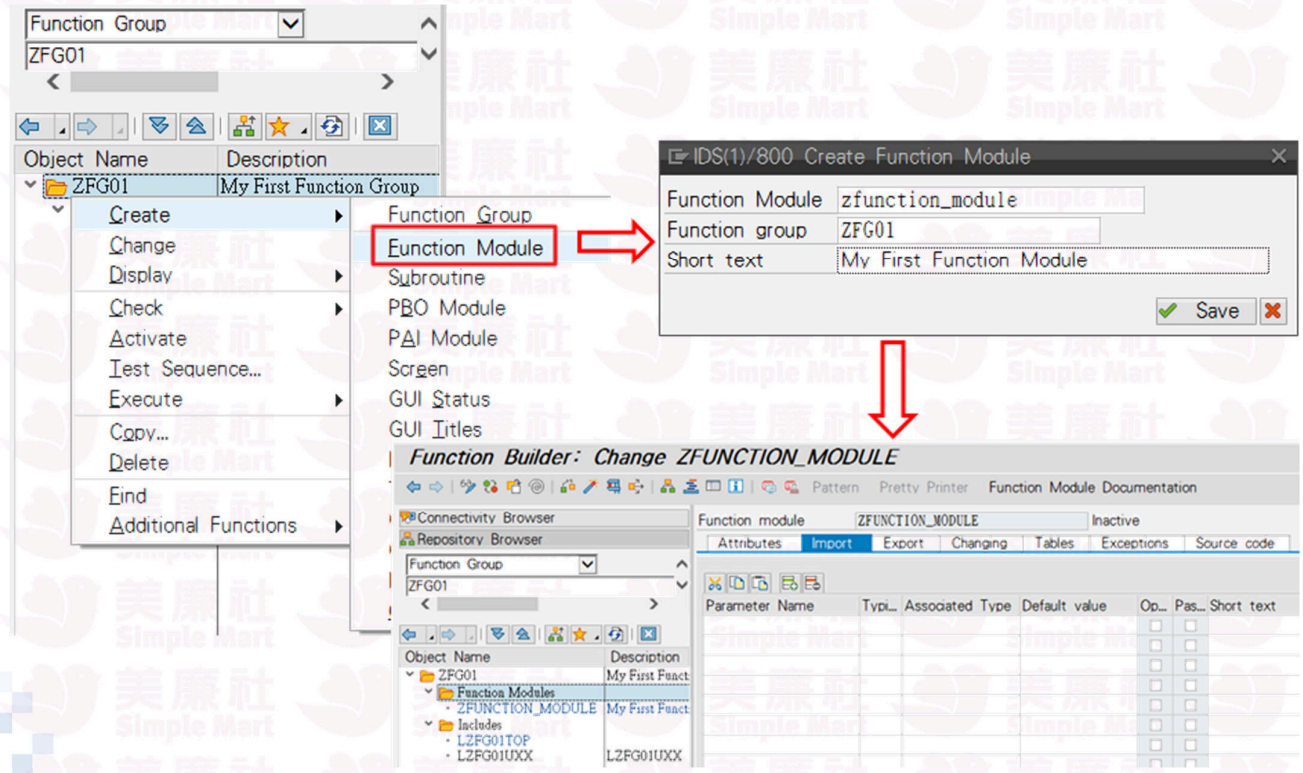
© 2018 NTT DATA Taiwan Co., Ltd.

T-Code: SE80
Creating a Function Group

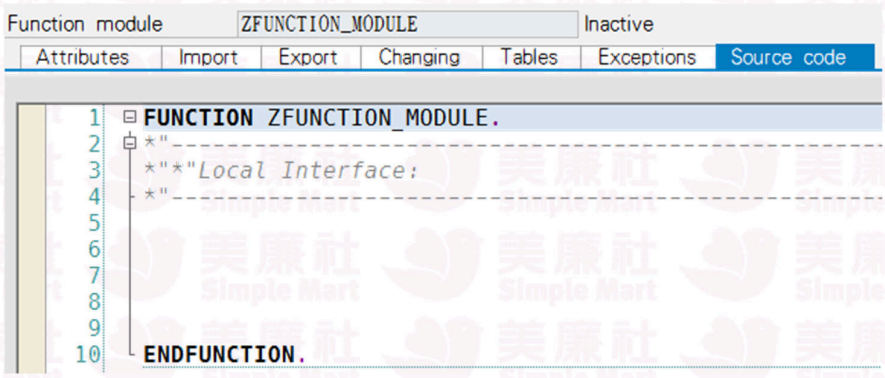


© 2018 NTT DATA Taiwan Co., Ltd.

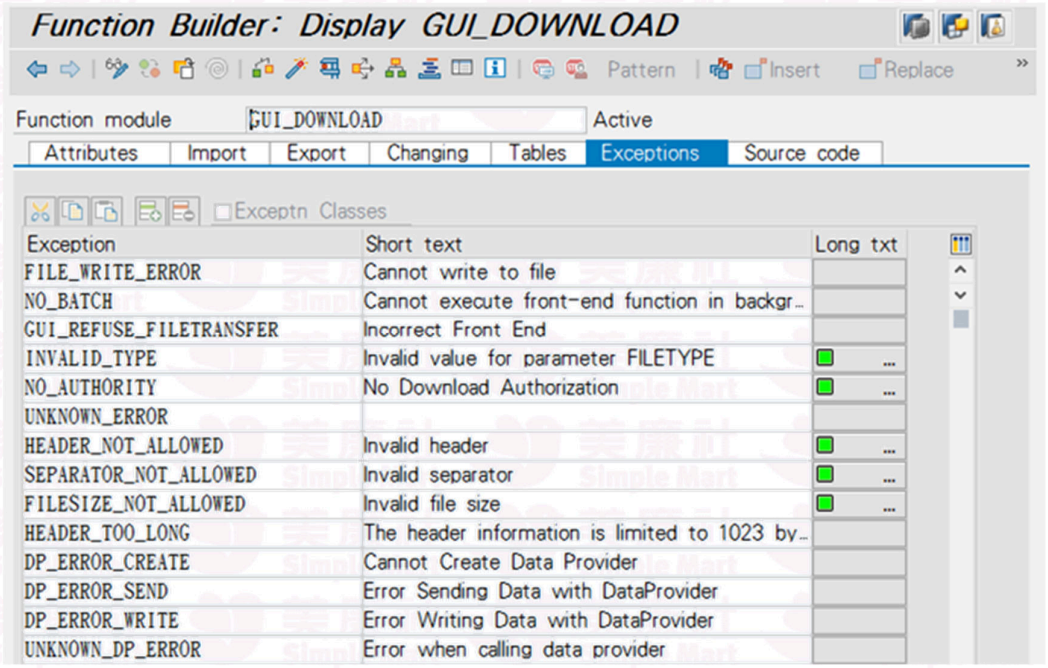
Creating a Function Module under a Function Group



Processing Logic

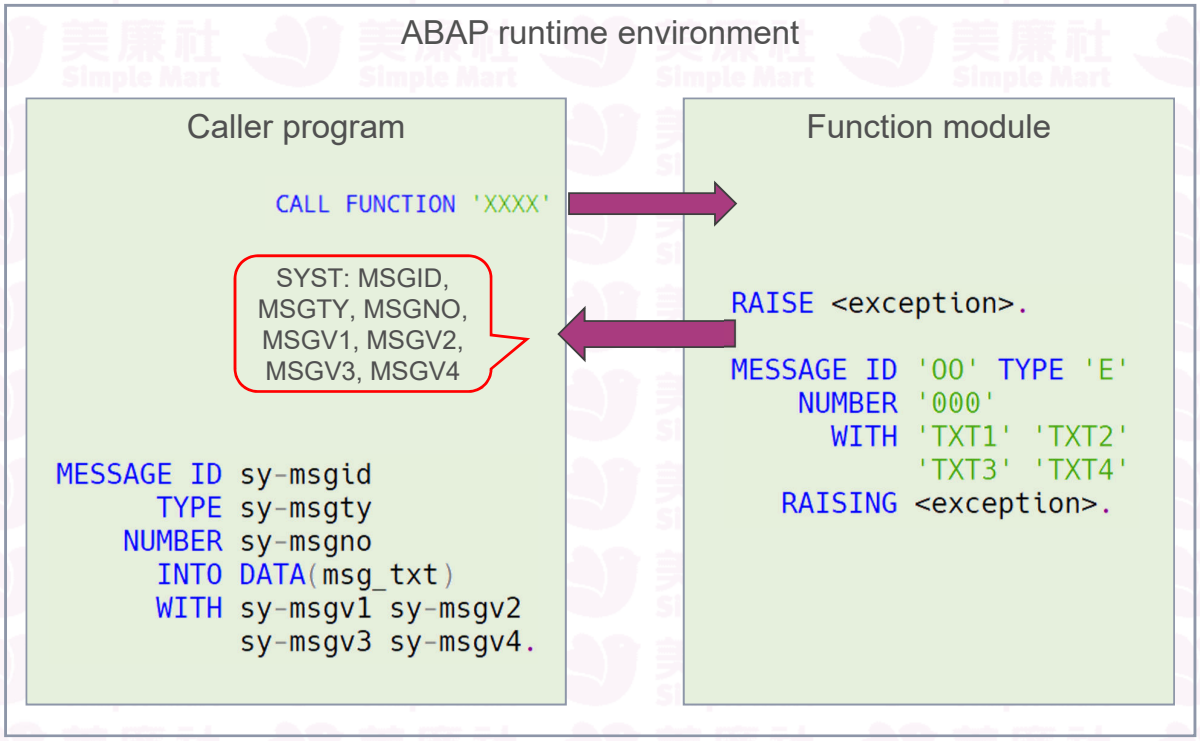


Define Exceptions



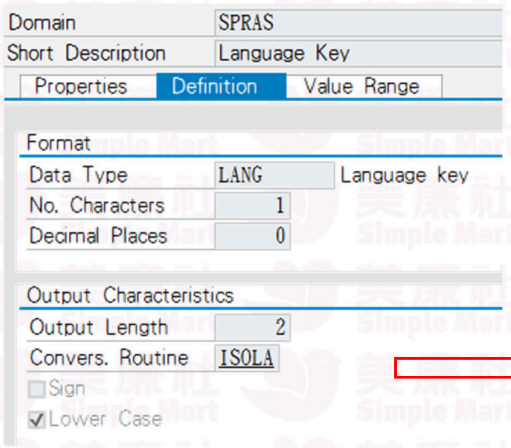
© 2018 NTT DATA Taiwan Co., Ltd.

Raise Exception



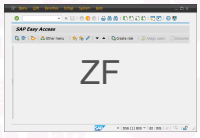
© 2018 NTT DATA Taiwan Co., Ltd.

Domain – Convert Routine



CONVERT_EXIT_xxxxx_INPUT

CONVERT_EXIT_xxxxx_OUTPUT

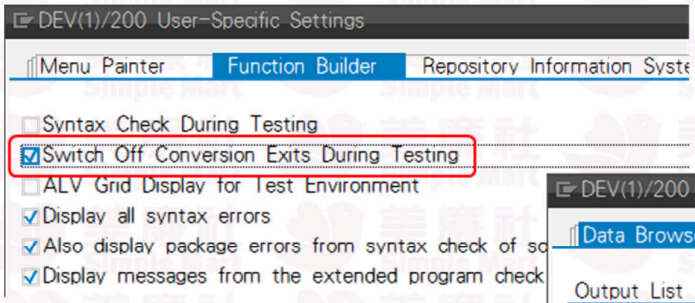


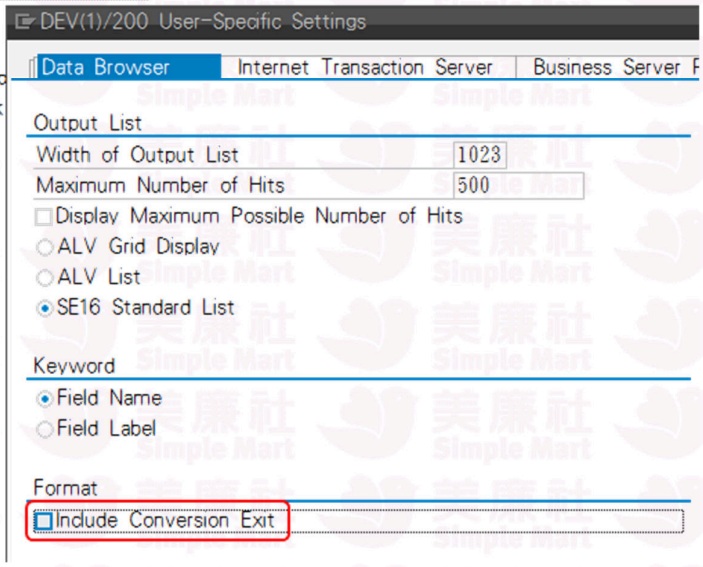
CONVERT_EXIT_**ISOLA**_INPUT

CONVERT_EXIT_**ISOLA**_OUTPUT

© 2018 NTT DATA Taiwan Co., Ltd.

Convert routine – environment setting

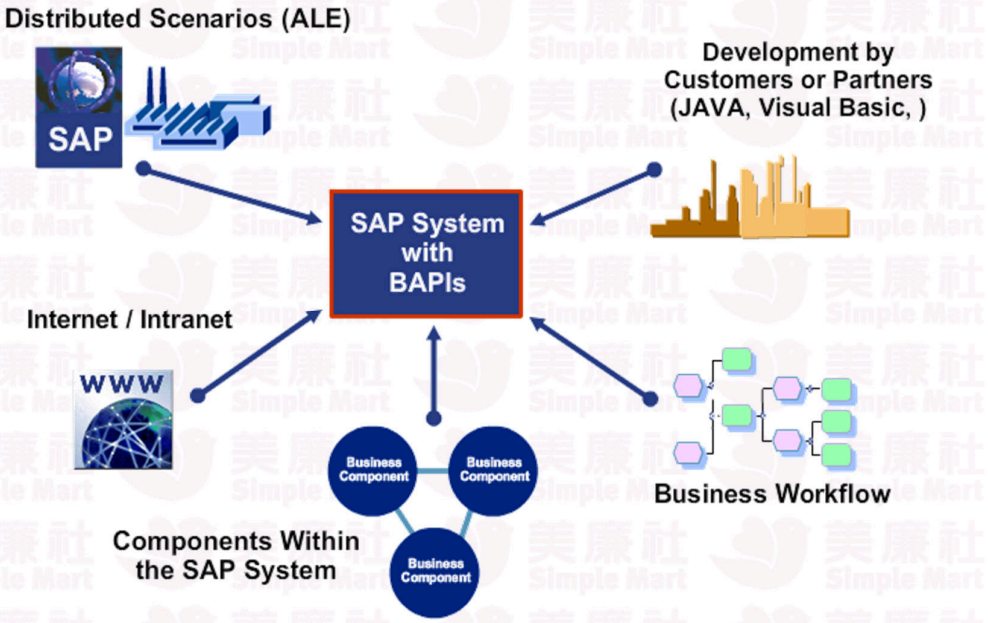




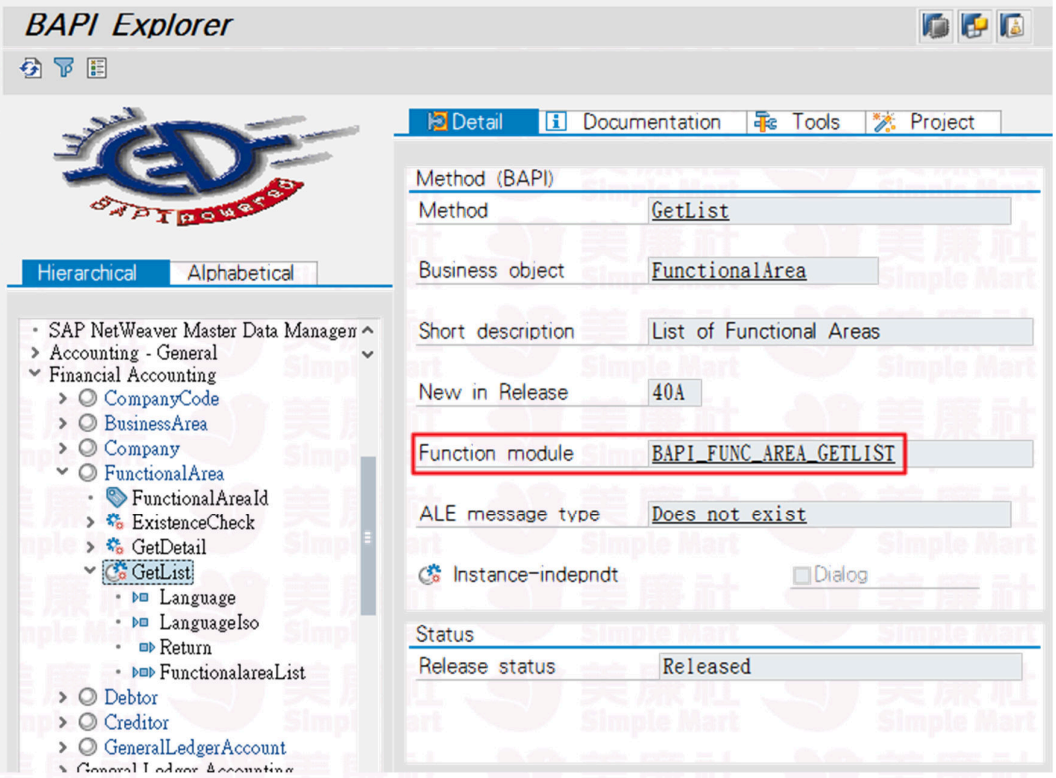
© 2018 NTT DATA Taiwan Co., Ltd.

BAPI Use

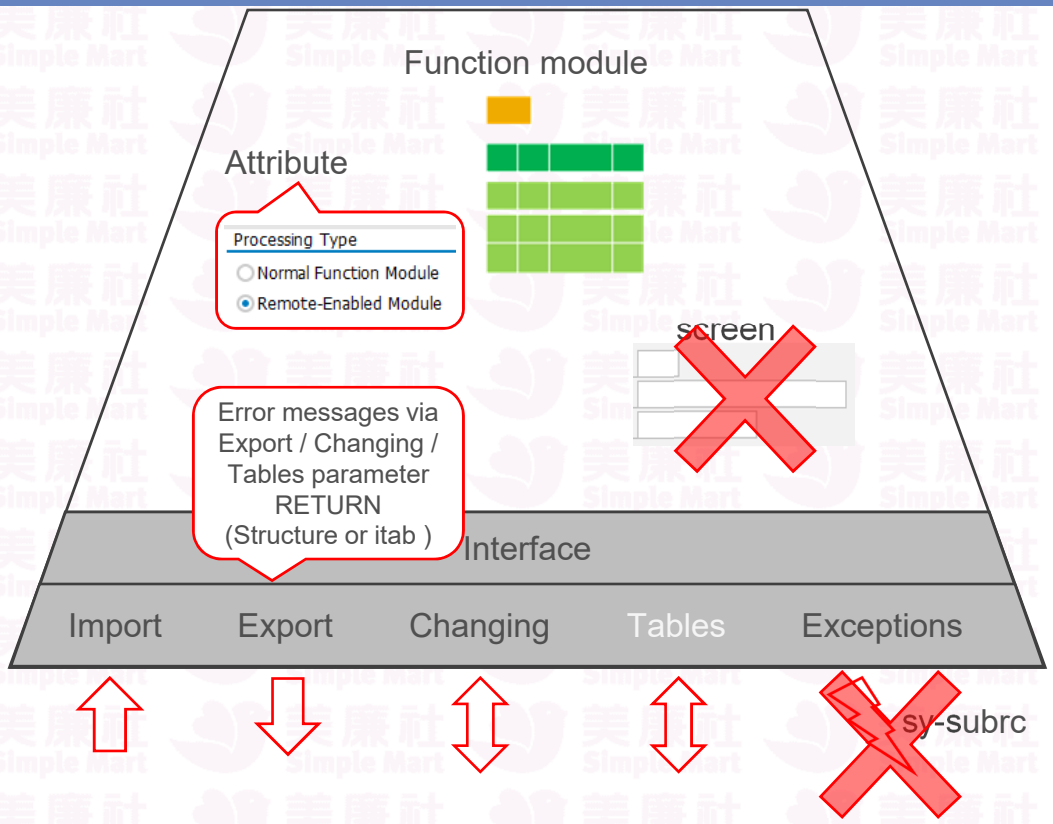
- BAPI = Business API



T-Code : BAPI
BAPI Explorer



Attributes of a BAPI Function Module



© 2018 NTT DATA Taiwan Co., Ltd.

其他必要的 BAPI

- 針對資料交易類的 BAPI (Create/Change/Delete/Release…)
- BAPI_TRANSACTION_COMMIT
- BAPI_TRANSACTION_ROLLBACK

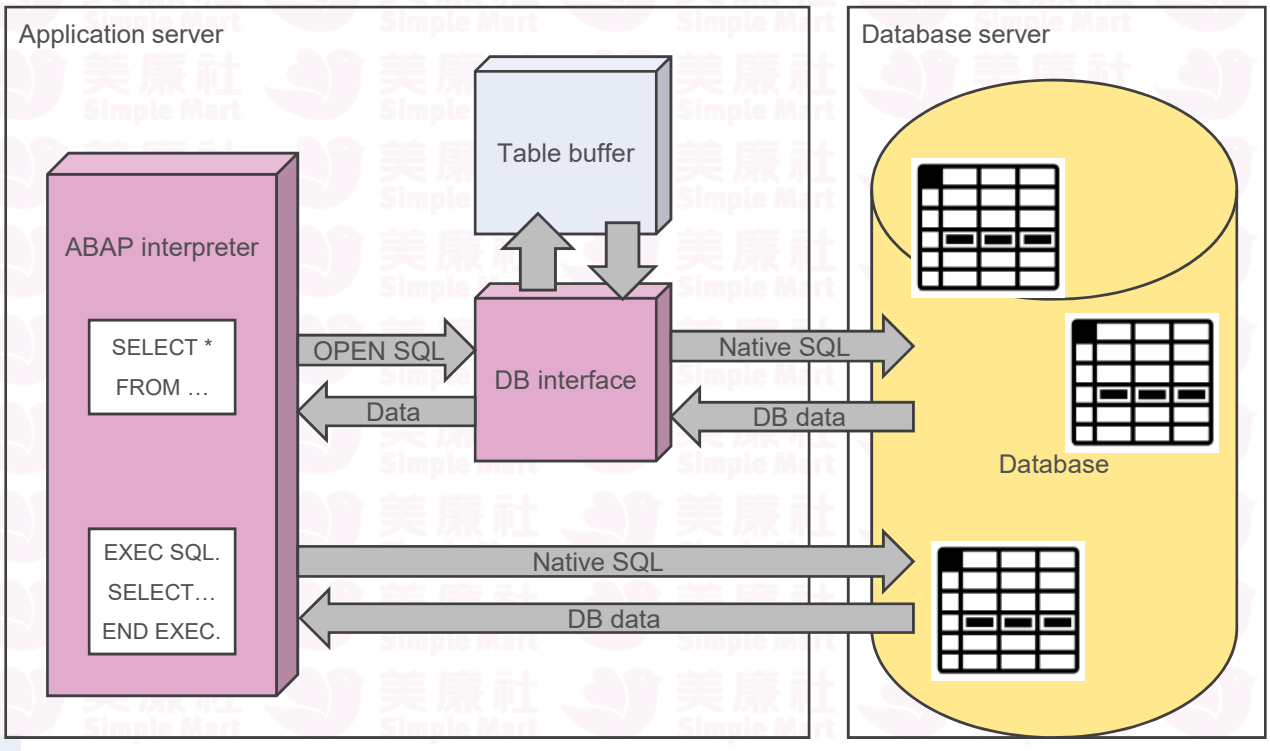
© 2018 NTT DATA Taiwan Co., Ltd.

Data Retrieval

- Read Database Tables

© 2018 NTT DATA Taiwan Co., Ltd.

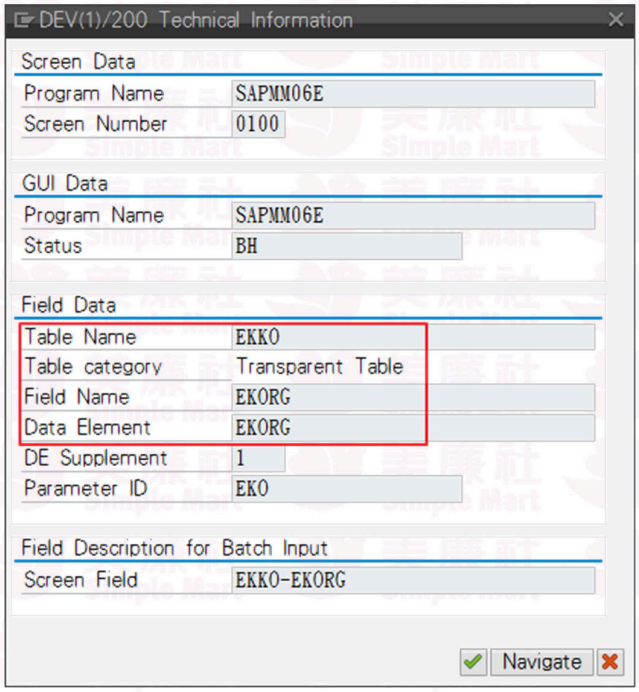
Database Access(Architecture)



© 2018 NTT DATA Taiwan Co., Ltd.

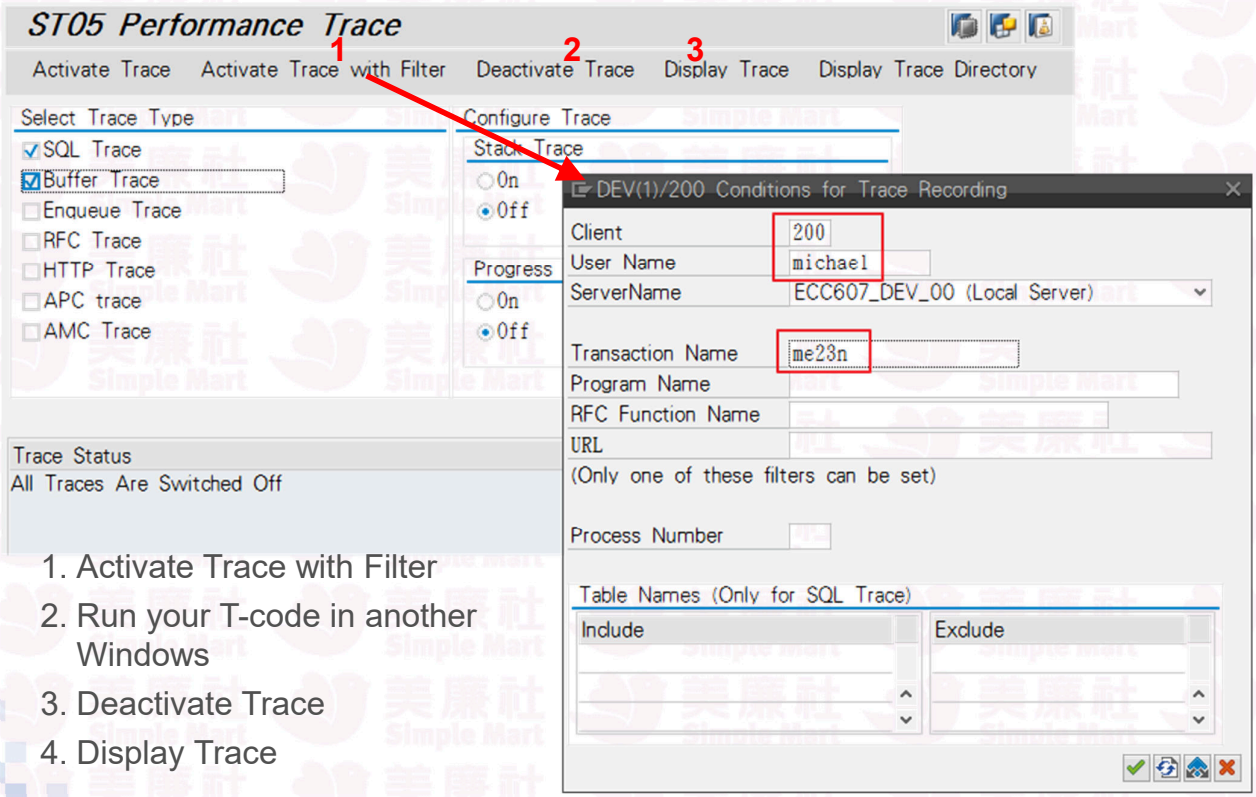
How to Get Correct Table/Field

- F1 => technical information
  - ABAP Dictionary：SE11 / SE16



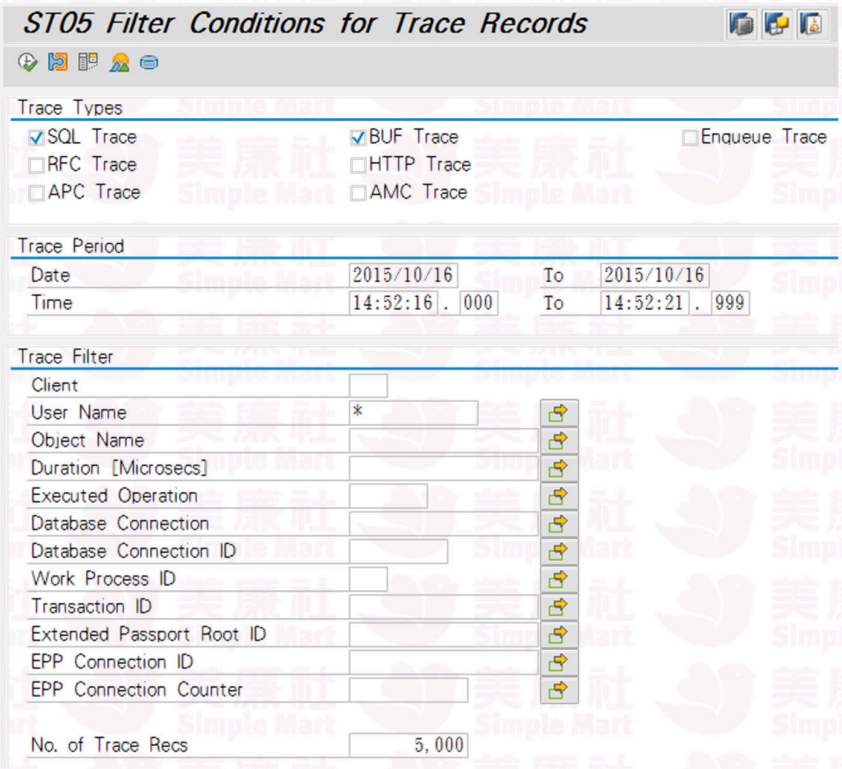
- SQL Trace(ST05)
  - ABAP Dictionary：SE11 / SE16

How to Get Correct Table/Field - step



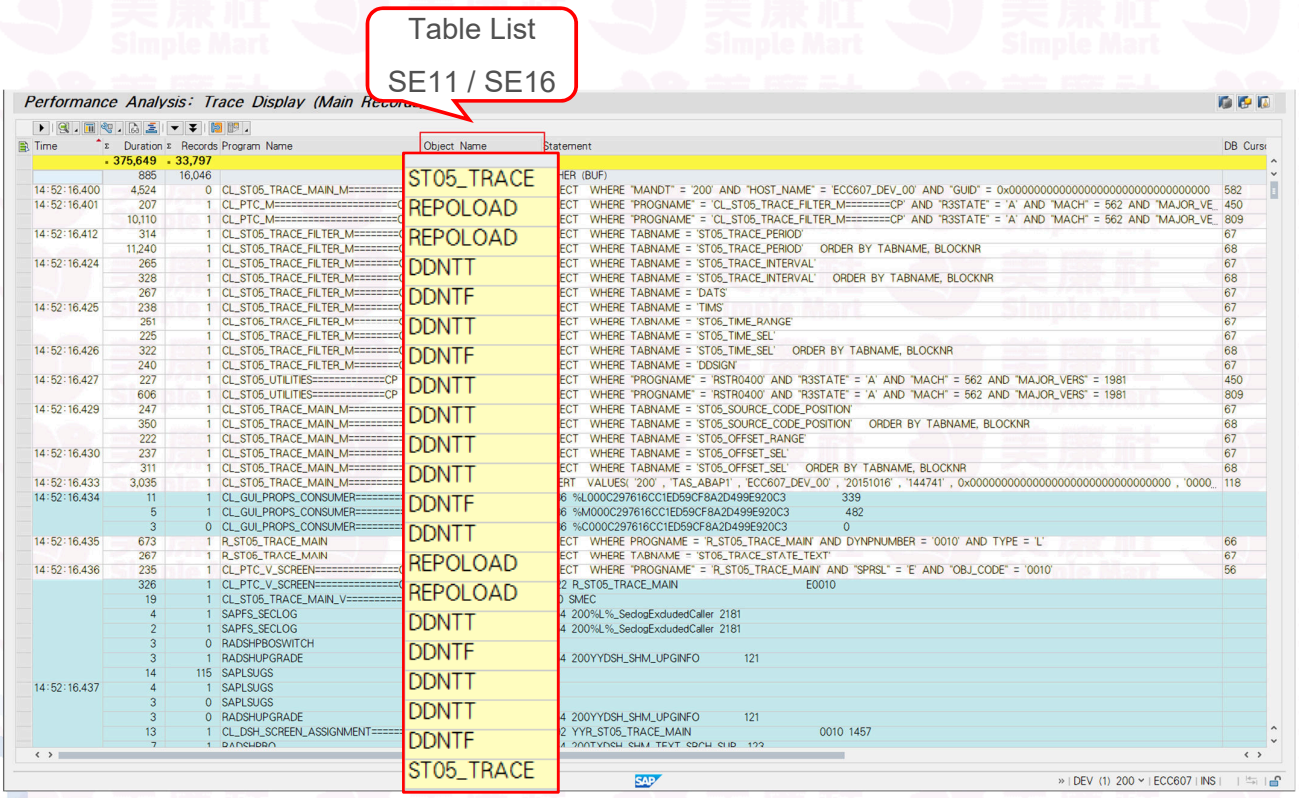
1. Activate Trace with Filter
2. Run your T-code in another Windows
3. Deactivate Trace
4. Display Trace

How to Get Correct Table/Field – Display trace



© 2018 NTT DATA Taiwan Co., Ltd.

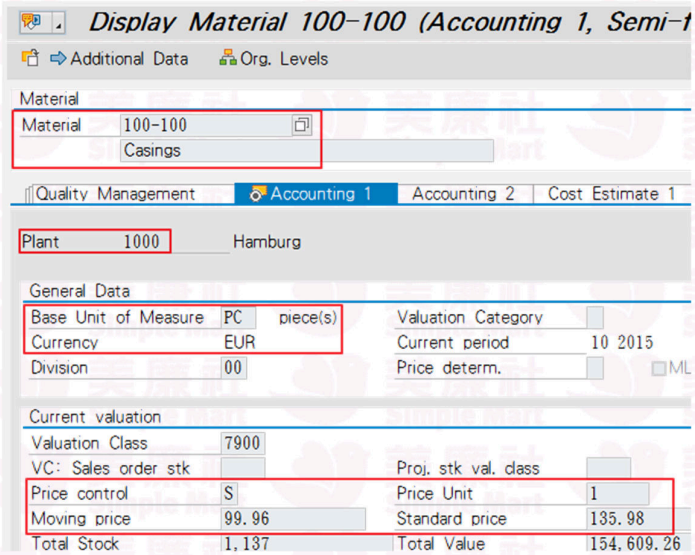
How to Get Correct Table/Field – Display trace result



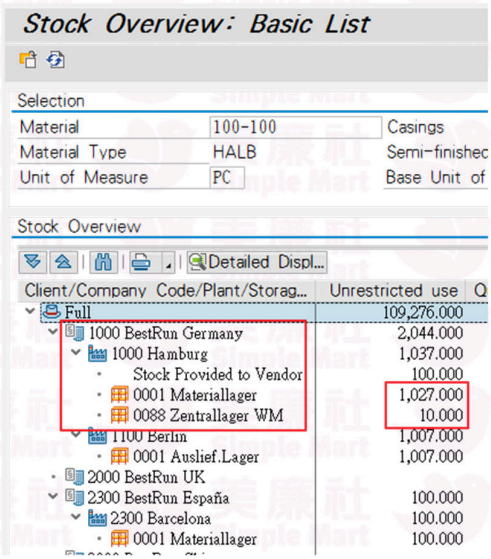
© 2018 NTT DATA Taiwan Co., Ltd.

Home Exercise : 庫存成本表

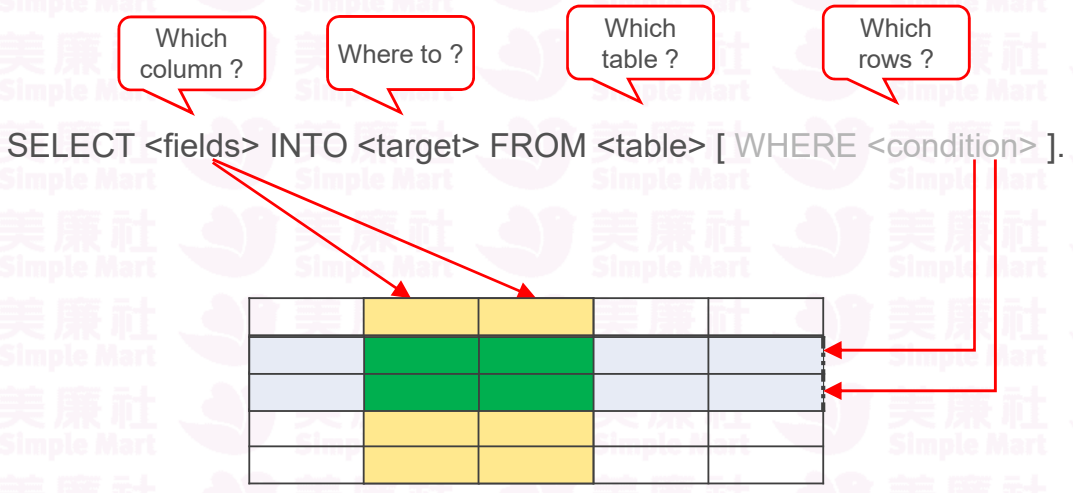
T-Code : MM03



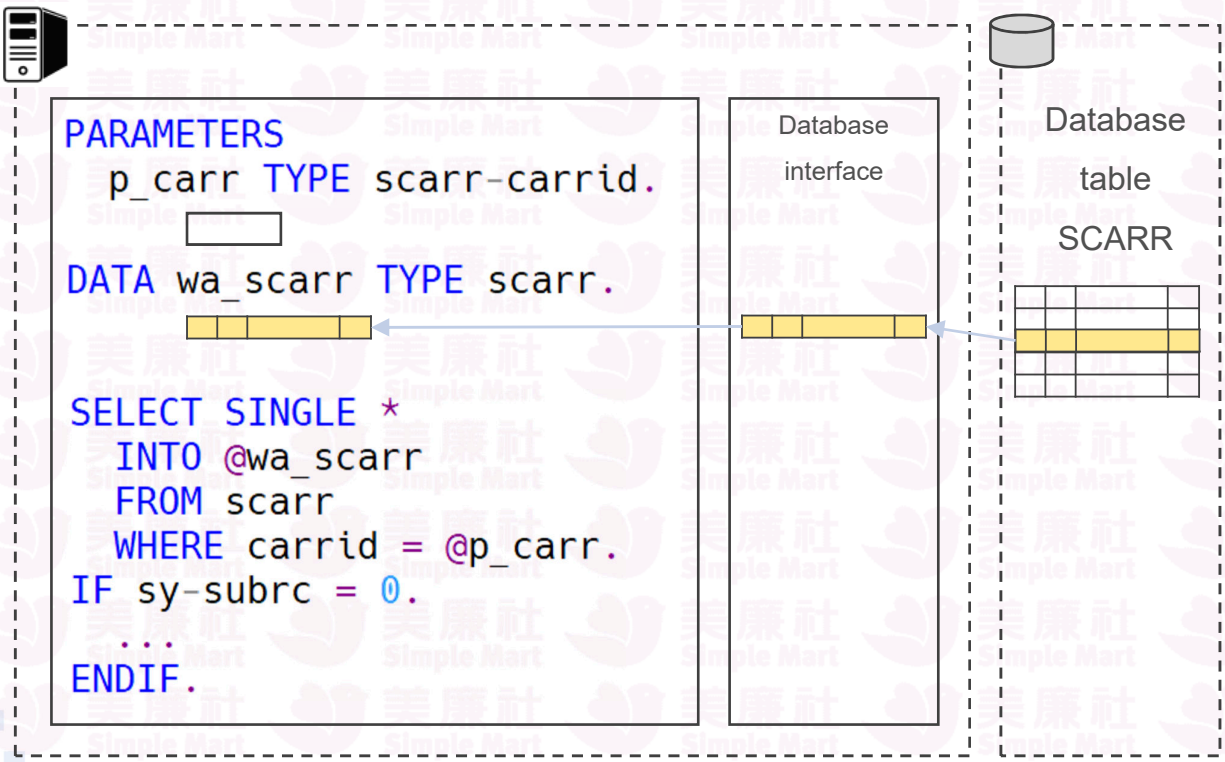
T-Code : MMBE



Database Read Access (Overview)

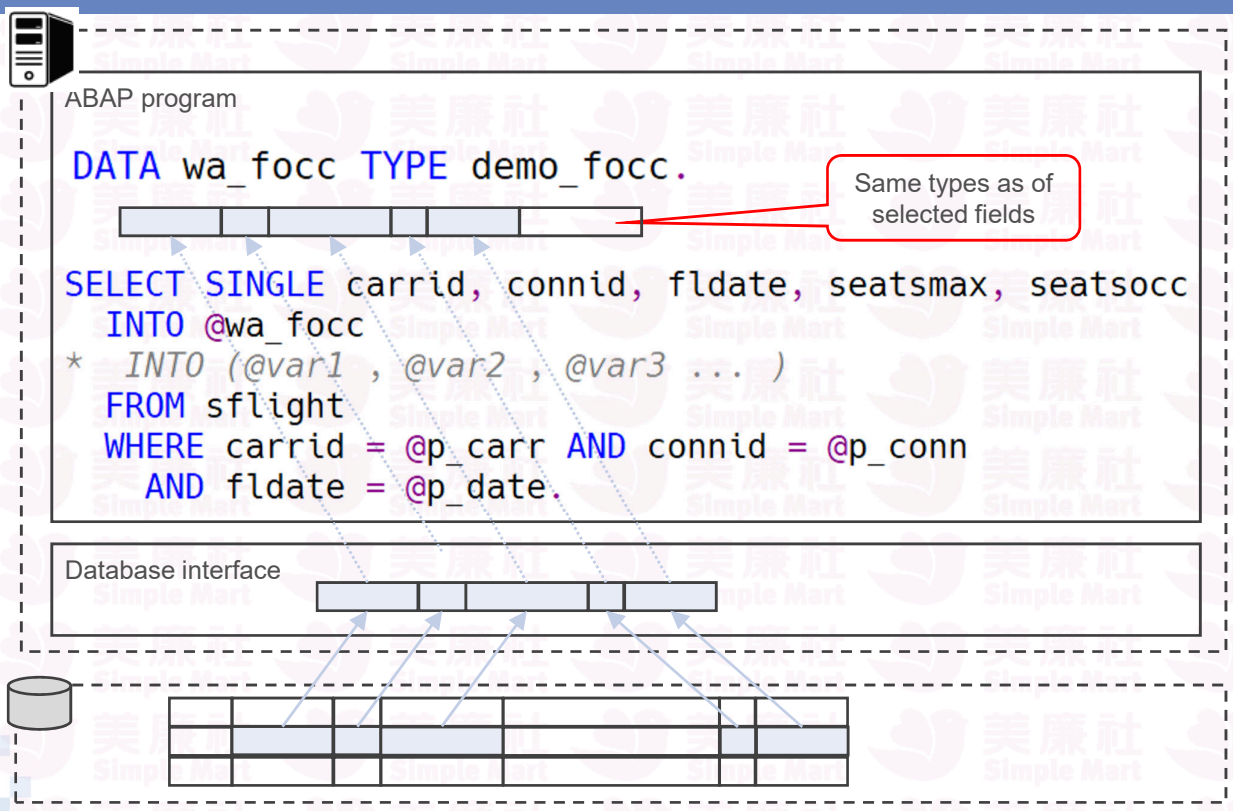


Reading Single Records



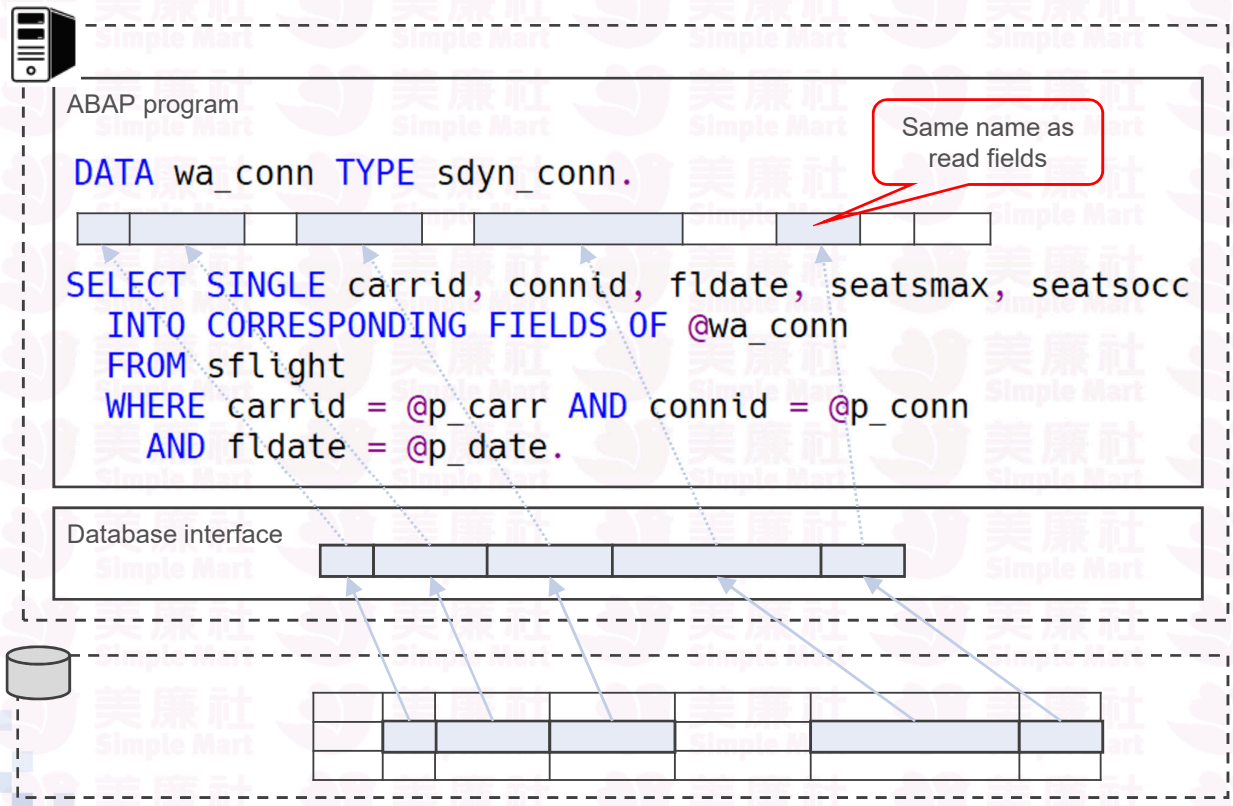
© 2018 NTT DATA Taiwan Co., Ltd.

Corresponding Target Structure for the Field List

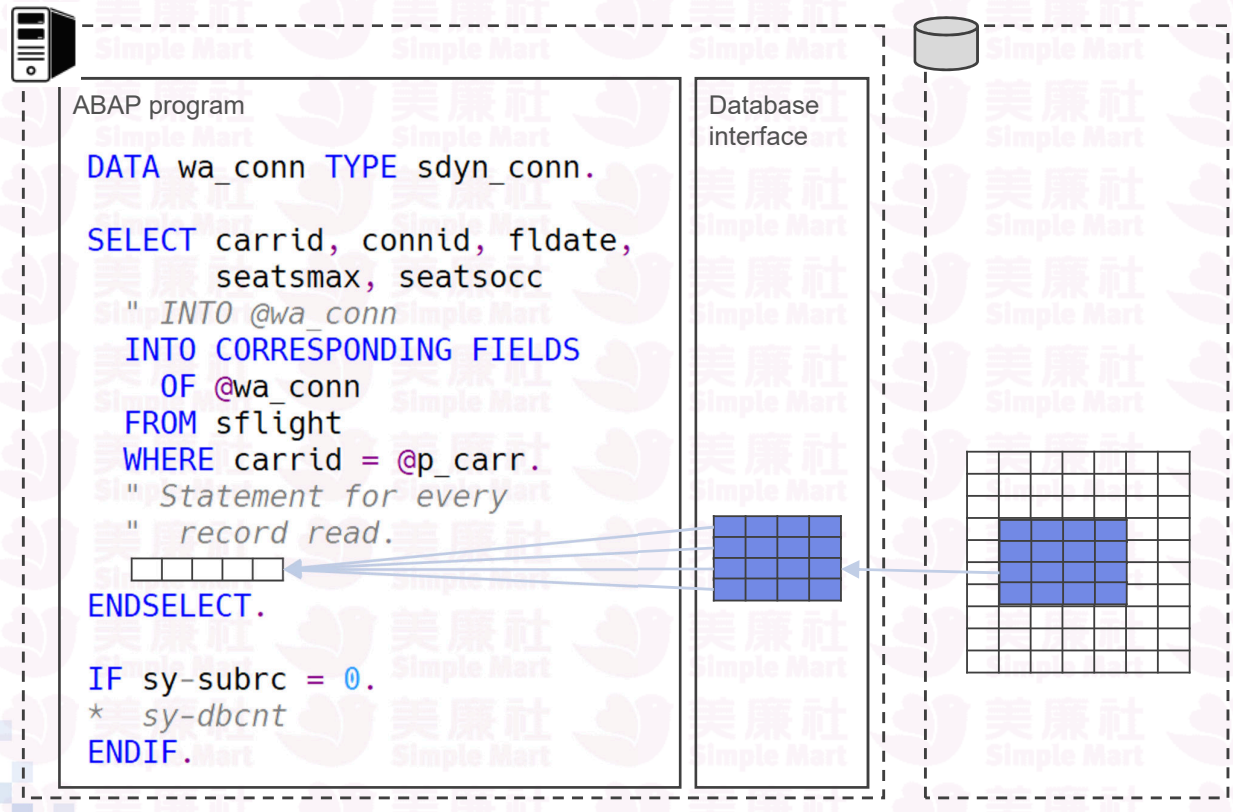


© 2018 NTT DATA Taiwan Co., Ltd.

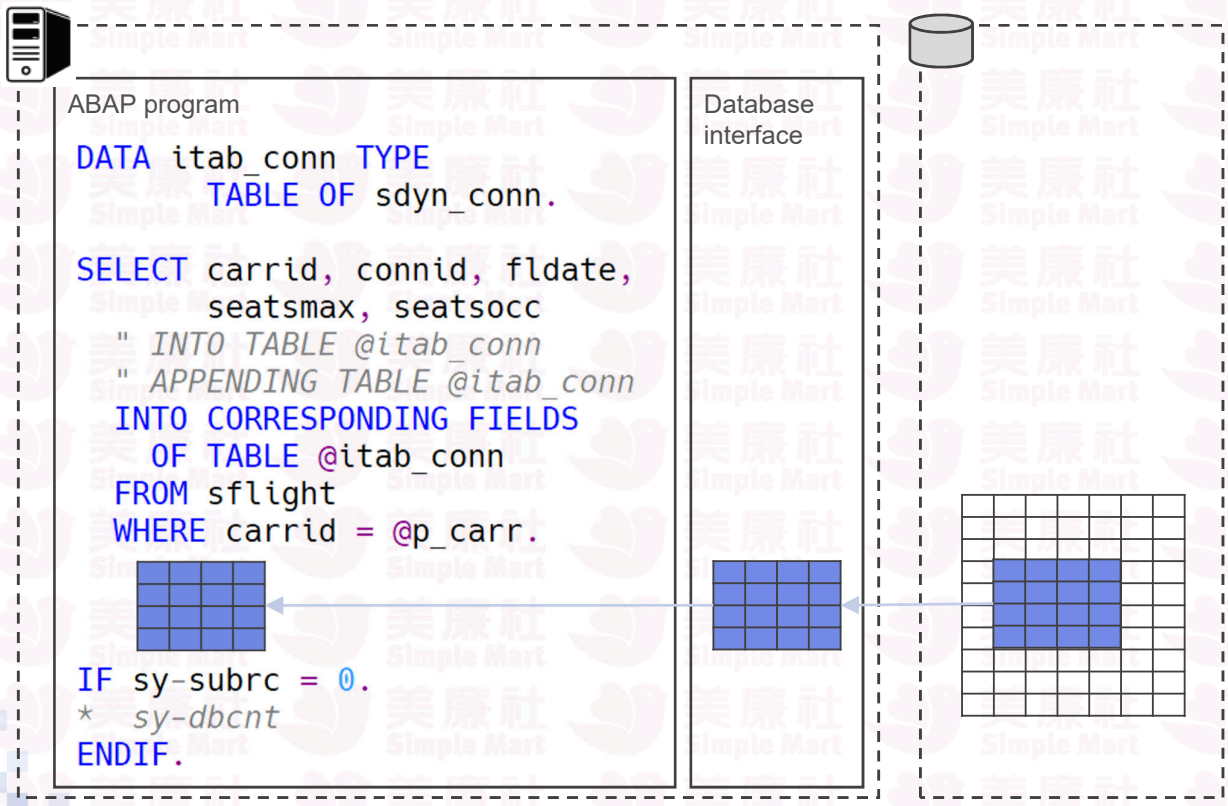
Fields in the Target Structure that have the Same Names as those in the Field List



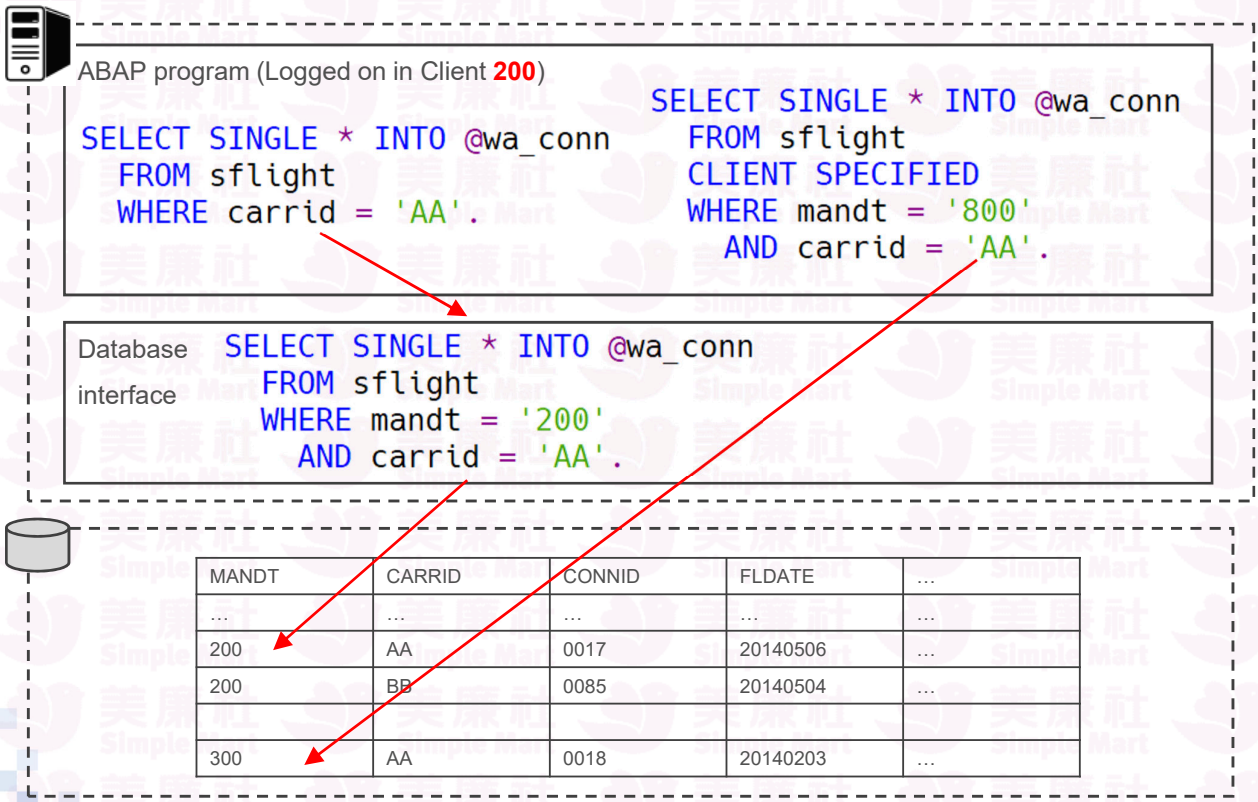
Reading Several rows using a loop



Reading Several Rows Using an Array Fetch

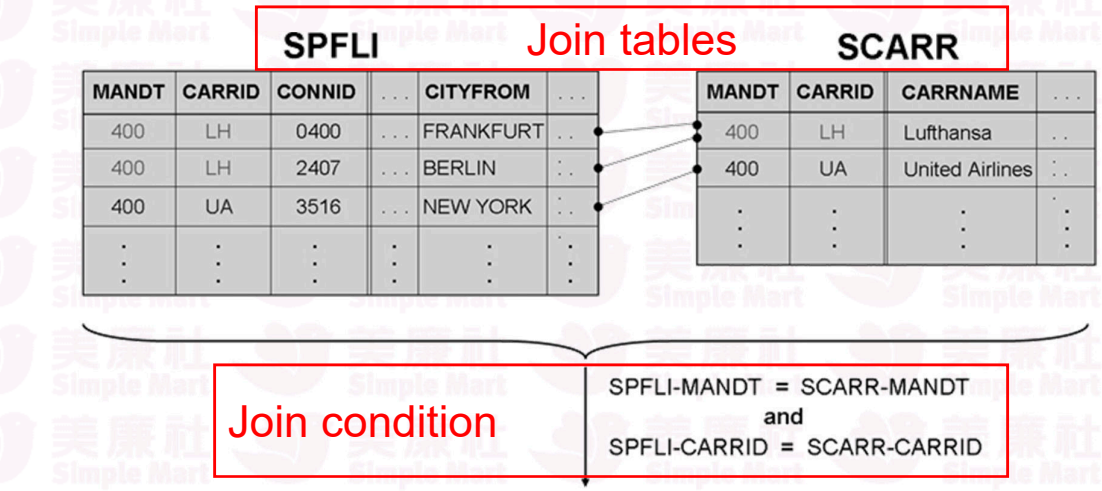


Reading Data From Client-Specific Tables



MANDT	CARRID	CONNID	FLDATE	…
…	…	…	…	…
200	AA	0017	20140506	…
200	BB	0085	20140504	…
300	AA	0018	20140203	…

Entries for Defining a Table Join



SPFLI

MANDT	CARRID	CONNID	...	CITYFROM	...
400	LH	0400	...	FRANKFURT	..
400	LH	2407	...	BERLIN	..
400	UA	3516	...	NEW YORK	..
...	...	...	...	...	...

SCARR

MANDT	CARRID	CARRNAME	...
400	LH	Lufthansa	..
400	UA	United Airlines	..
...	...	...	...

Table Join

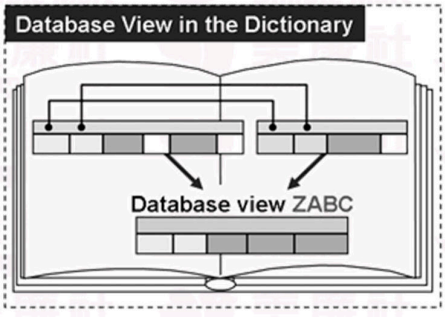
Join columns

MANDT	CARRID	CONNID	CITYFROM	CARRNAME
400	LH	0400	FRANKFURT	Lufthansa
400	LH	2407	BERLIN	Lufthansa
400	UA	3516	NEW YORK	United Airlines
...	...	...	...	...

© 2018 NTT DATA Taiwan Co., Ltd.

Implementation Options for Table Joins

View definition in the dictionary + selection in the program



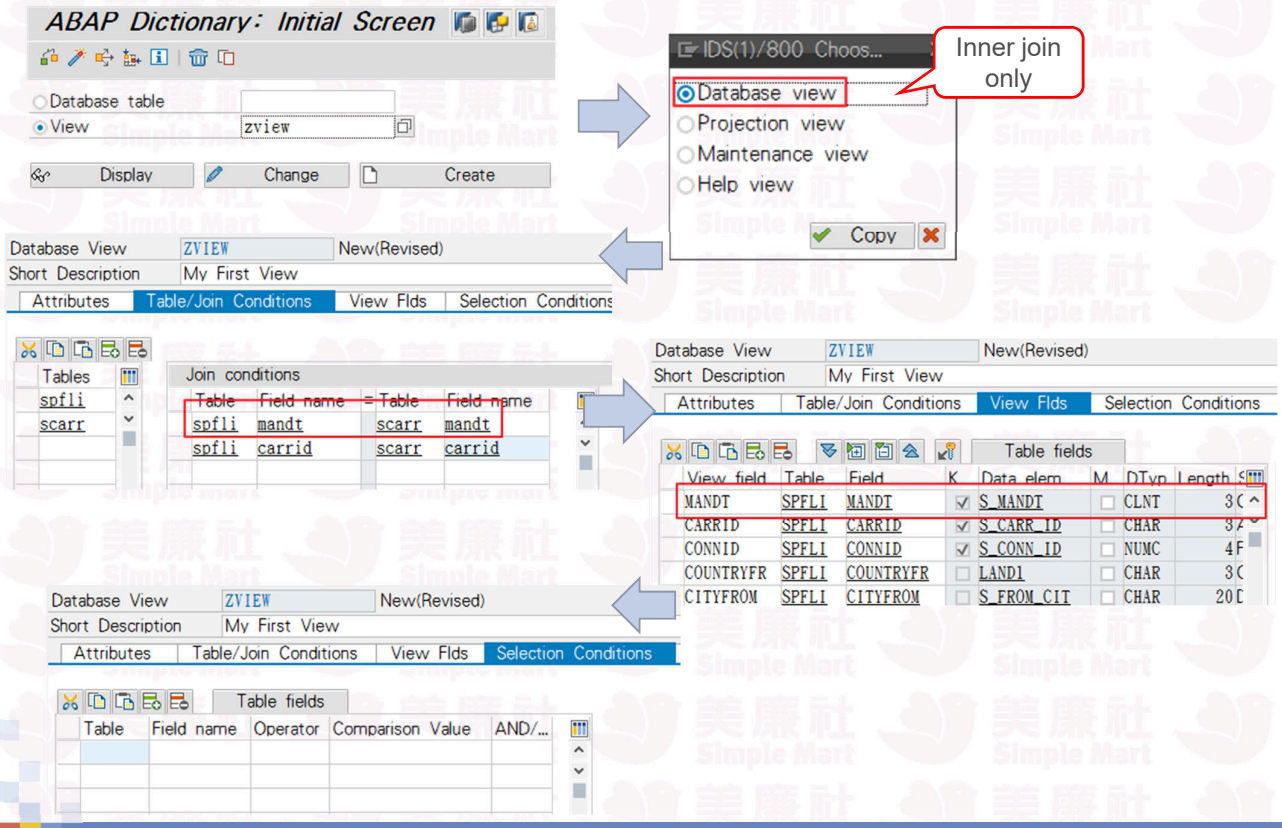

```
SELECT *
  FROM zabc
  WHERE ...
```

Direct join implementation in the program (ABAP join)

```
SELECT *
  FROM spfli INNER JOIN scarr
    on spfli~carrid = scarr~carrid
  WHERE ...
```

© 2018 NTT DATA Taiwan Co., Ltd.

Create View in ABAP Dictionary (SE11)

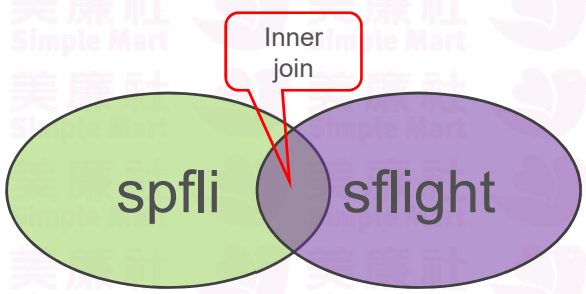


Inner Join

```
SELECT spfli~carrid, spfli~connid,
       spfli~airpfrom, spfli~airpto,
       sflight~fldate, sflight~seatsmax
  INTO TABLE @itab
  FROM spfli INNER JOIN sflight
    on spfli~carrid = sflight~carrid
   AND spfli~connid = sflight~connid
 WHERE spfli~carrid = 'LH'
   AND spfli~connid = '0400'.
```

A	B	C
a1	b1	c1
a2	b2	c2
a3	b3	c3

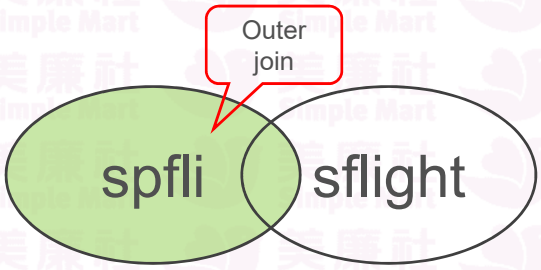
A	D	E
a1	d1	e1
a3	d2	e2
a3	d3	e3



A	B	C	D	E
a1	b1	c1	d1	e1
a3	b3	c3	d2	e2
a3	b3	c3	d3	e3

Outer Join

```
SELECT spfli~carrid, spfli~connid,
       spfli~airpfrom, spfli~airpto,
       sflight~fldate, sflight~seatsmax
  INTO TABLE @itab
*  FROM spfli RIGHT OUTER JOIN sflight
  FROM spfli LEFT OUTER JOIN sflight
    on spfli~carrid = sflight~carrid
   AND spfli~connid = sflight~connid
  WHERE spfli~carrid = 'LH'
    AND spfli~connid = '0400'.
```



A	B	C
a1	b1	c1
a2	b2	c2
a3	b3	c3

A	D	E
a1	d1	e1
a3	d2	e2
a3	d3	e3

A	B	C	D	E
a1	b1	c1	d1	e1
a2	b2	c2		
a3	b3	c3	d2	e2
a3	b3	c3	d3	e3

© 2018 NTT DATA Taiwan Co., Ltd.

Union (ABAP 740)

```
SELECT a AS c1, b AS c2, c AS c3, d AS c4
  FROM demo_join1
UNION DISTINCT
SELECT d AS c1, e AS c2, f AS c3, g AS c4
  FROM demo_join2
  INTO TABLE @DATA(result_distinct).

SELECT a AS c1, b AS c2, c AS c3, d AS c4
  FROM demo_join1
UNION ALL
SELECT d AS c1, e AS c2, f AS c3, g AS c4
  FROM demo_join2
  INTO TABLE @DATA(result_all).
```

© 2018 NTT DATA Taiwan Co., Ltd.

For All Entries

```
SELECT carrid, connid
  INTO TABLE @it_spfli
  FROM spfli
  WHERE cityfrom = @p_cityf
    AND cityto = @p_cityt.

" check, if at least one table entry is found
IF it_spfli[] IS NOT INITIAL.
  SELECT carrid, connid,fldate
    INTO TABLE @it_sflight
    FROM sflight
    FOR ALL ENTRIES IN @it_spfli
    WHERE carrid = @it_spfli-carrid
      AND connid = @it_spfli-connid.
ENDIF.
```

itab_spfli

LH	0400	...
LH	0402	...

```
where (    carrid = 'LH'
       and connid = '0400')
or     (   carrid = 'LH'
       and connid = '0402')
or     (...
```


OPEN SQL ⇔ New OPEN SQL

SQL92 Standard

Advanced Open SQL
in ABAP 7.4 SP2 and beyond

Open SQL
releases before ABAP 7.4

Before ABAP 740

```
TYPES: BEGIN OF st,
         carrid TYPE sbook-carrid,
         connid TYPE sbook-connid,
         fldate TYPE sbook-fldate,
         bookid TYPE sbook-bookid,
       END OF st.
DATA itab TYPE TABLE OF st.

SELECT carrid connid fldate bookid
  INTO CORRESPONDING FIELDS OF TABLE itab
  FROM sbook
  WHERE carrid = 'LH'
    AND connid = '0400'.
```

In ABAP 740

```
TYPES: BEGIN OF st,
         carrid TYPE sbook-carrid,
         connid TYPE sbook-connid,
         fldate TYPE sbook-fldate,
         bookid TYPE sbook-bookid,
       END OF st.
DATA itab TYPE TABLE OF st.

SELECT carrid, connid, fldate, bookid
  INTO CORRESPONDING FIELDS OF TABLE @itab
  FROM sbook
  WHERE carrid = 'LH'
    AND connid = '0400'.
```

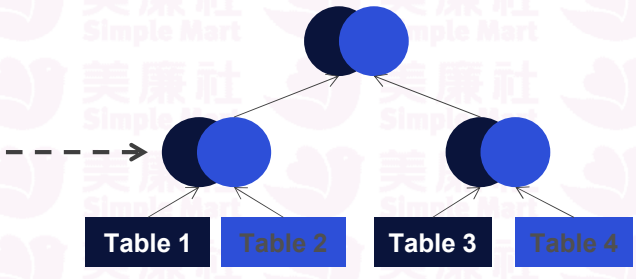
New Open SQL syntax

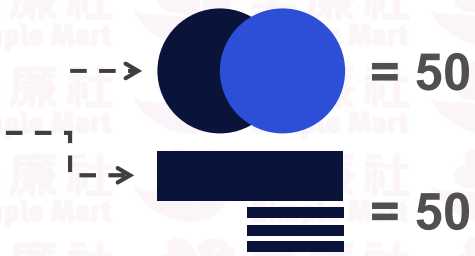
- ABAP variables escaped with “@”
  - Host variables -- global or local variables are used in Open SQL.
  - When the escape character “@” is used, the syntax check is performed in a strict mode.
    - Strict mode: Nearly all former warnings transform into syntax errors.
- Comma-separated column list
  - Don’t mix comma-separated fields in Open SQL with the colon-comma syntax for repeated ABAP statements.
- Arithmetic expressions using +, -, *, /, DIV, MOD, ABS, FLOOR, CEIL
- Common semantics
- String concatenation
- SQL CASE, COALESCE

© 2018 NTT DATA Taiwan Co., Ltd.

Further Limitations Removed

- RIGHT OUTER JOIN now supported
- Use brackets in JOIN expressions
  - To override implicit left-bracketing
- ON conditions extended:
  - No field of the right table required in ON condition
  - No longer limited to Equi-Joins
  - Fields of the right table usable in WHERE clause of LEFT OUTER JOIN
  - Implicit client handing
- Maximum number of tables in JOINs increased
- Maximum number of subqueries increased





© 2018 NTT DATA Taiwan Co., Ltd.

New OPEN SQL with Inline Declaration

```
TYPES: BEGIN OF st,
         carrid TYPE sbook-carrid,
         connid TYPE sbook-connid,
         fldate TYPE sbook-fldate,
         bookid TYPE sbook-bookid,
       END OF st.
DATA itab TYPE TABLE OF st.

SELECT carrid, connid, fldate, bookid
  INTO CORRESPONDING FIELDS OF TABLE @itab
  FROM sbook
  WHERE carrid = 'LH'
    AND connid = '0400'.
```

ITAB

CARRID	CONNID	FLDATE	BOOKID
LH	0400	1995-02-28	00000001
LH	0400	1995-02-28	00000002
LH	0400	1995-02-28	00000003

```
SELECT carrid, connid, fldate, bookid
  INTO TABLE @DATA(itab)
  FROM sbook
  WHERE carrid = 'LH'
    AND connid = '0400'.
```


JOINs

- Use brackets in JOIN expressions

Before ABAP 740

```
SELECT vbak~vbeln AS vbak_vbeln
       vbap~posnr AS vbap_vbeln
       likp~vbeln AS likp_vbeln
       lips~posnr AS lips_posnr
  INTO TABLE itab
  FROM ( vbak INNER JOIN vbap
       ON vbak~vbeln = vbap~vbeln )
     INNER JOIN
     ( likp INNER JOIN lips
       ON likp~vbeln = lips~vbeln )
     ON vbap~vbeln = lips~vgbel
    AND vbap~posnr = lips~vgpos
  WHERE vbak~vbtyp = 'C'.
```

ABAP 740

```
SELECT vbak~vbeln AS vbak_vbeln,
       vbap~posnr AS vbap_vbeln,
       likp~vbeln AS likp_vbeln,
       lips~posnr AS lips_posnr
  INTO TABLE @itab
  FROM ( vbak INNER JOIN vbap
       ON vbak~vbeln = vbap~vbeln )
     INNER JOIN
     ( likp INNER JOIN lips
       ON likp~vbeln = lips~vbeln )
     ON vbap~vbeln = lips~vgbel
    AND vbap~posnr = lips~vgpos
  WHERE vbak~vbtyp = 'C'.
```

Arithmetic Calculations - + - * /

- Only the operators +, -, and * can be used in arithmetic expressions with integer type or decimal type.
- CAST()
  – AS FLTP

```
SELECT carrid, connid, fldate, loccuram,
       loccuram + 10 AS loccur_add,
       loccuram - 10 AS loccur_sub,
       loccuram * 10 AS loccur_mul,
       CAST( loccuram AS FLTP ) / CAST( 10 AS FLTP )
         AS loccur_div
  INTO TABLE @DATA(itab)
  FROM sbook
  WHERE carrid = 'LH'
    AND connid = '0400'.
```

© 2018 NTT DATA Taiwan Co., Ltd.

Arithmetic Calculations – Function

Syntax	Valid Argument Types
ABS(arg)	All numeric types except decimal floating point numbers.
CEIL(arg)	Dictionary types DEC, CURR, and QUAN. ABAP type p with decimal places.
FLOOR(arg)	Dictionary types DEC, CURR, and QUAN. ABAP type p with decimal places.
DIV(arg1,arg2)	Dictionary types INT1, INT2, INT4, DEC, CURR, and QUAN without decimal places. ABAP types b, s, I and p without decimal places.
MOD(arg1,arg2)	Dictionary types INT1, INT2, and INT4, DEC, CURR, and QUAN without decimal places. ABAP types b, s, and i plus p without decimal places

© 2018 NTT DATA Taiwan Co., Ltd.

String Concatenation & SQL CASE

- String Concatenation
  - string1 && string2

- SQL CASE
  - CASE …
    WHEN … THEN …
    WHEN … THEN …
    ELSE …
    END

```
SELECT addrnumber, name1,
       CASE tel_extens
         WHEN ' ' THEN tel_extens
         ELSE tel_number && '-' && tel_extens
       END AS tel,
       CASE fax_extens
         WHEN ' ' THEN fax_number
         ELSE fax_number && '-' && fax_extens
       END AS fax
  FROM adrc
  INTO TABLE @DATA(itab).
```

© 2018 NTT DATA Taiwan Co., Ltd.

SQL - coalesce

- COALESCE ( )
  - Replace NULL value.

```
SELECT COALESCE( s_carrid, '--' ) AS carrid2,
       COALESCE( s_carrname, '--' ) AS carrname2,
       p_connid
  INTO TABLE @DATA(itab)
  FROM scarr AS s RIGHT OUTER JOIN spfli AS p
    ON s~carrid = p~carrid.
```

ITAB

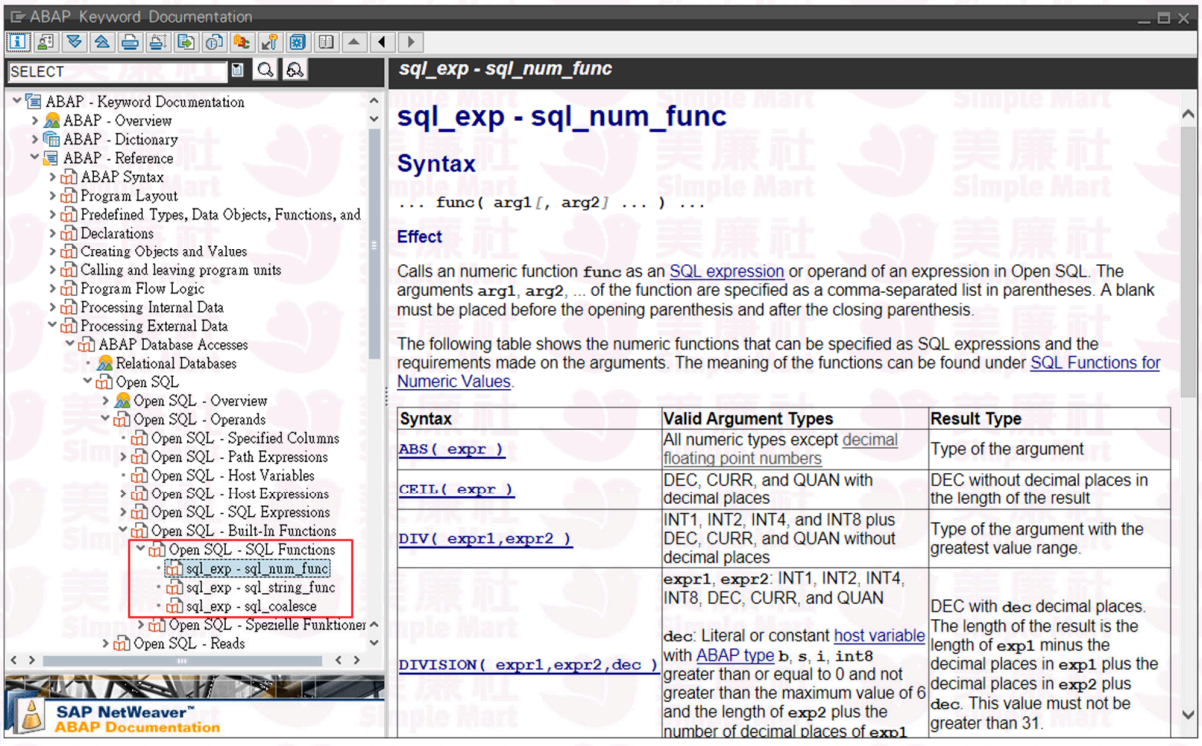
CARRID2	CARRNAME2	CONNID
--	--	0017
--	--	0064
AC	Air Canada	0820
AF	Air France	0820
--	--	1699
--	--	1984
LH	Lufthansa	0400
LH	Lufthansa	0402
LH	Lufthansa	0454
LH	Lufthansa	0455

© 2018 NTT DATA Taiwan Co., Ltd.

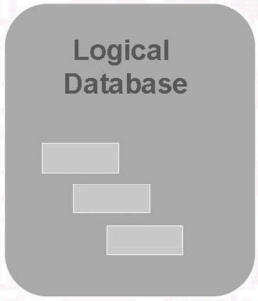
SQL - string

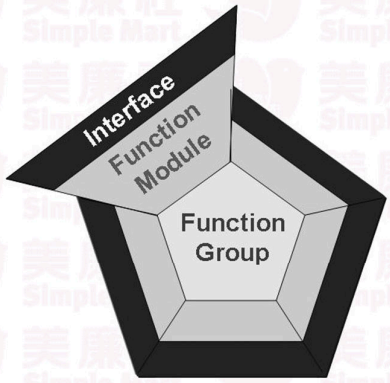
```
SELECT matnr, maktx ,
       CASE LENGTH( maktx )
         WHEN 7 THEN SUBSTRING( maktx,1,3 )
         ELSE SUBSTRING( maktx,5,6 )
       END AS maktx2
  FROM makt
  WHERE matnr IN ('100-100', '100-200')
  INTO TABLE @DATA(lt_makt).
```

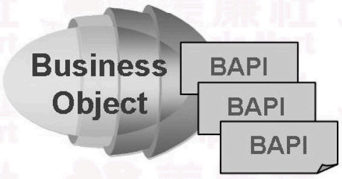

and others...

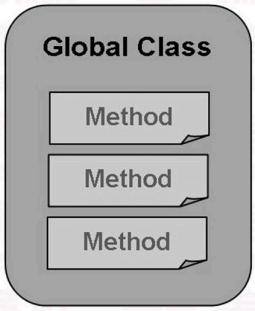


Reuse Components for Data Retrieval



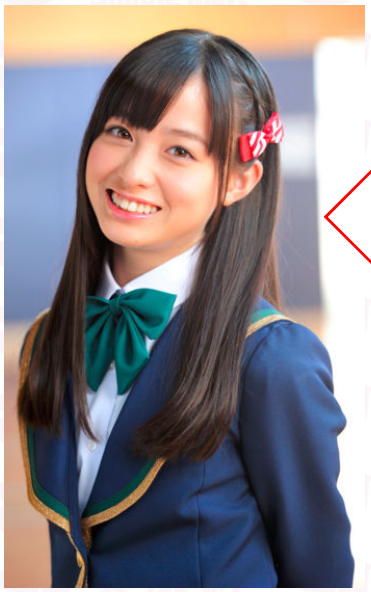


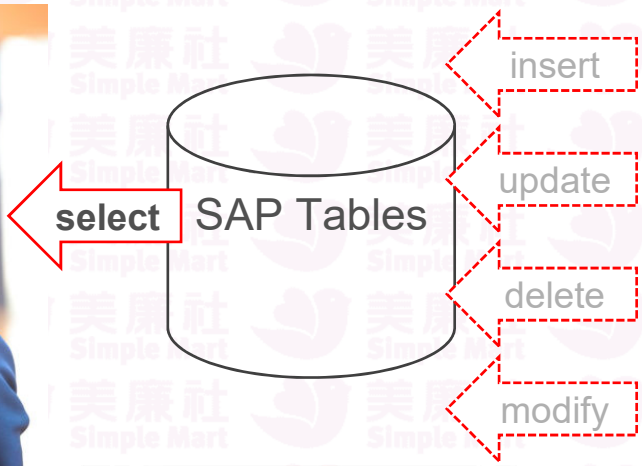


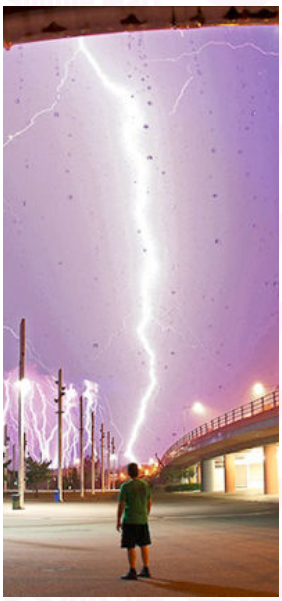


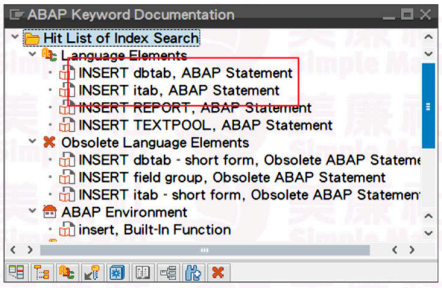
© 2018 NTT DATA Taiwan Co., Ltd.

Database Change Accesses









© 2018 NTT DATA Taiwan Co., Ltd.

Introduction to the Eclipse/ADT

✓**Install and Setup Eclipse**

✓**Developing Programs in Eclipse**

© 2018 NTT DATA Taiwan Co., Ltd.

Client System Requirement (@2018Q3)

Component	Prerequisite
Eclipse Platform	Oxygen (4.7)
Operating System	•Windows 7/8/10, or •Apple Mac OS X 10.10, Universal 64-Bit, or •Linux distribution
Java Runtime	JRE version 1.8, 32-Bit or 64-Bit
SAP GUI	•For Windows OS: SAP GUI for Windows 7.40 •For Apple Mac or Linux OS: SAP GUI for Java 7.40
Microsoft VC Runtime	For Windows OS: DLLs VS2010 for communication with the back-end system is required. NOTE: Install either the x86 or the x64 variant, accordingly to your 32- or 64-Bit Eclipse installation.

© 2018 NTT DATA Taiwan Co., Ltd.

Install and Setup – Step 1

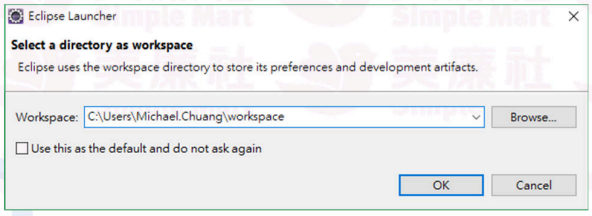
- 先至 此網址 參考 SAP Development Tools 支援的 Eclipse 版本。截至 2018 Q3，建議使用的版本為 Oxygen
  - https://tools.hana.ondemand.com/

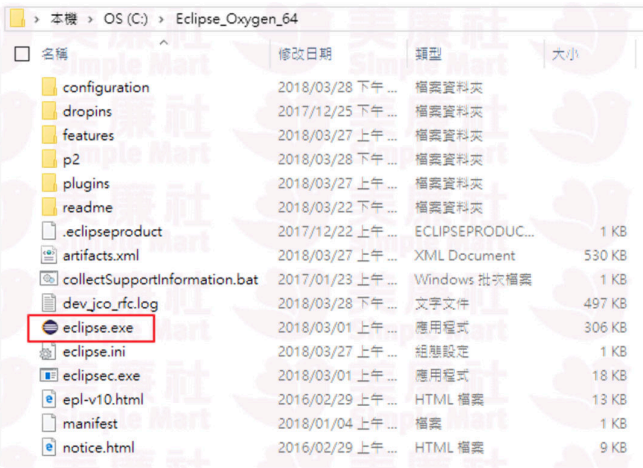
Tab	Tool	Available for Eclipse	
		Neon	Oxygen
ABAP	ABAP Development Tools for SAP NetWeaver	✓	✓
BW	Modeling Tools for SAP BW/4HANA and SAP BW powered by SAP HANA	✓	✓
Cloud	SAP Cloud Platform Tools Latest SDK downloads: • Java Web: neo-java-web-sdk-1.148.8.3.zip • Java Web Tomcat 7: neo-java-web-sdk-2.95.13.1.zip • Java Web Tomcat 8: neo-java-web-sdk-3.49.11.1.zip • Java EE 6 Web Profile: neo-javaee6-wp-sdk-2.130.3.1.zip • Java EE 7 Web Profile TomEE 7: neo-javaee7-wp-sdk-1.19.11.1.zip	✓	✓
Cloud Integration	SAP Cloud Platform Integration Tools	✓	
HANA	SAP HANA Tools	✓	✓
IdM	SAP Identity Management Configuration Lifecycle Tools	✓	
Mobile	SAP Mobile Tools		
SAPUI5	UI Development Toolkit for HTML5	✓	✓

© 2018 NTT DATA Taiwan Co., Ltd.

Install and Setup – Step 2

- 下載合適的 Eclipse 版本
- 直接解壓縮到任意路徑下
- 執行 Eclipse.exe
- 使用預設的工作目錄即可

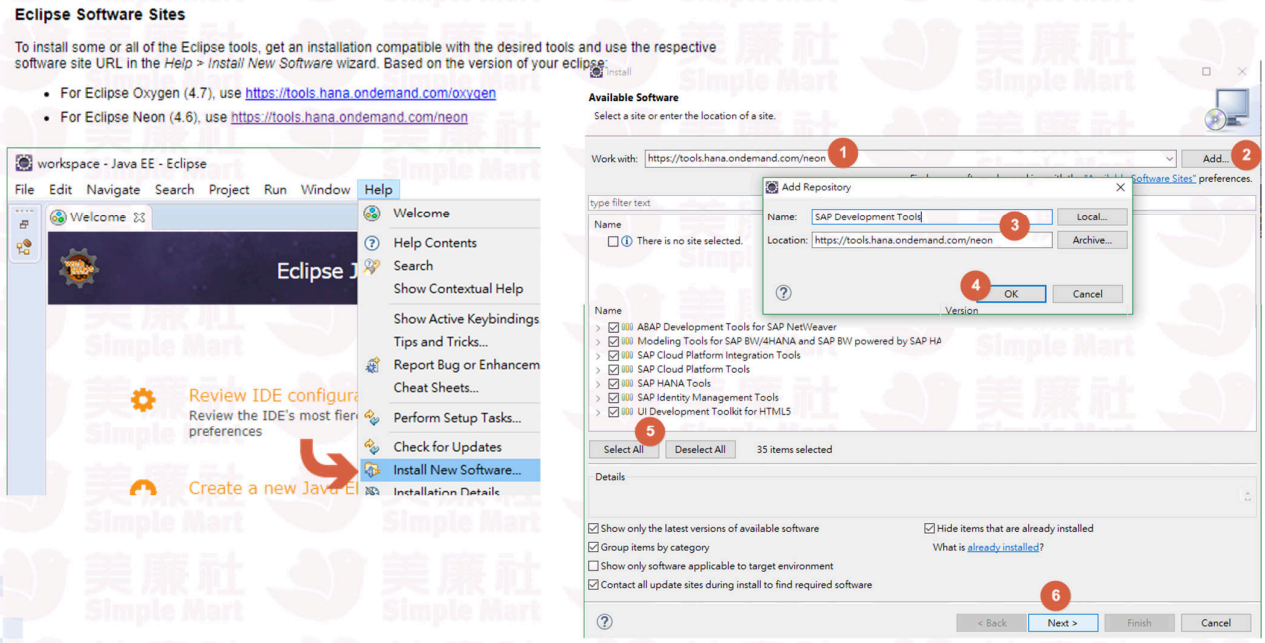




© 2018 NTT DATA Taiwan Co., Ltd.

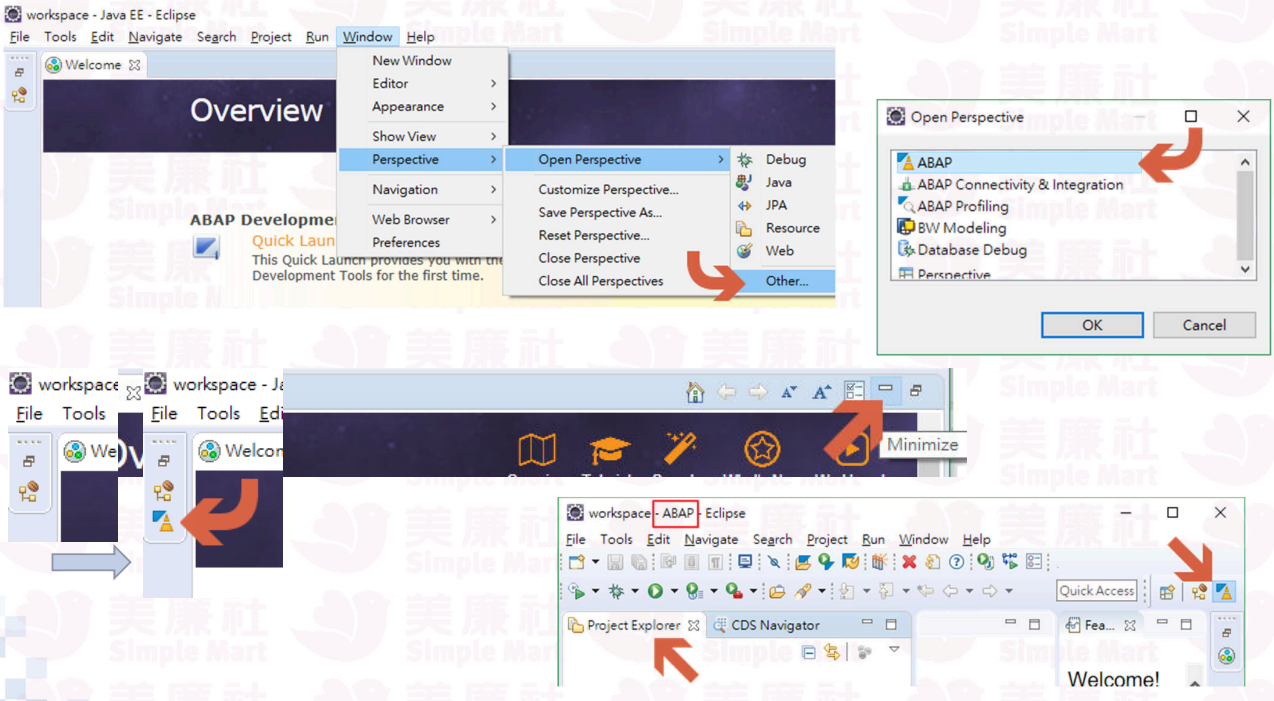
Install and Setup – Step 3

- 安裝 ABAP Development Tools(ADT)

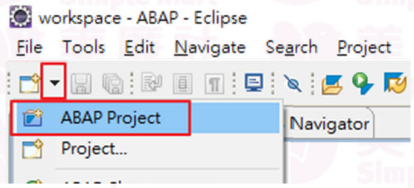


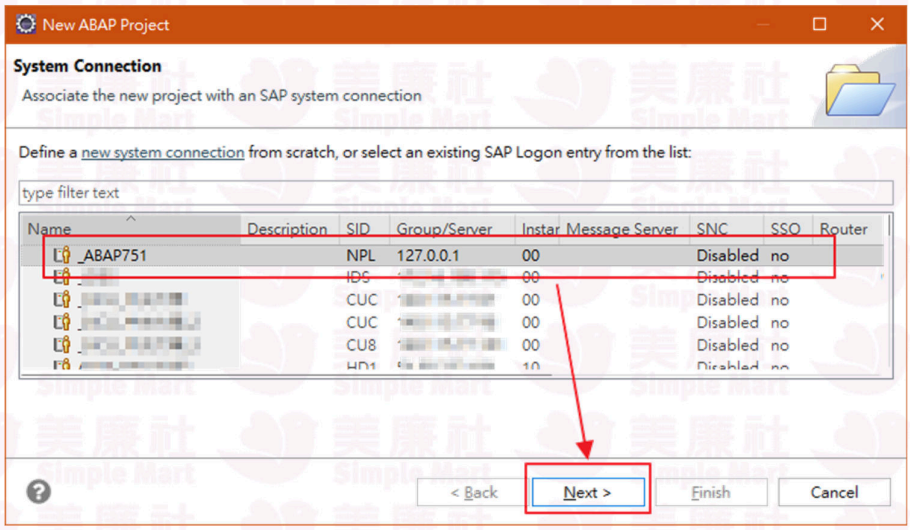
Install and Setup – Step 4

- Switch to ABAP Environment

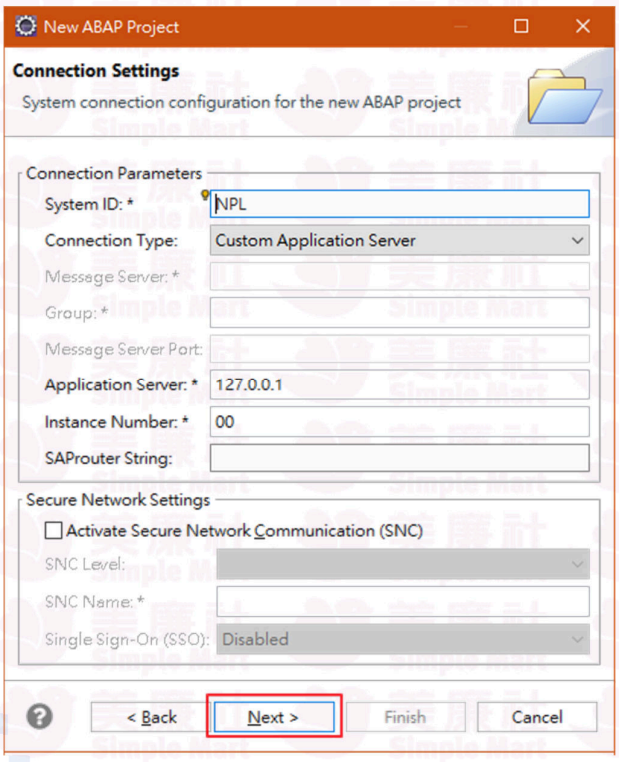


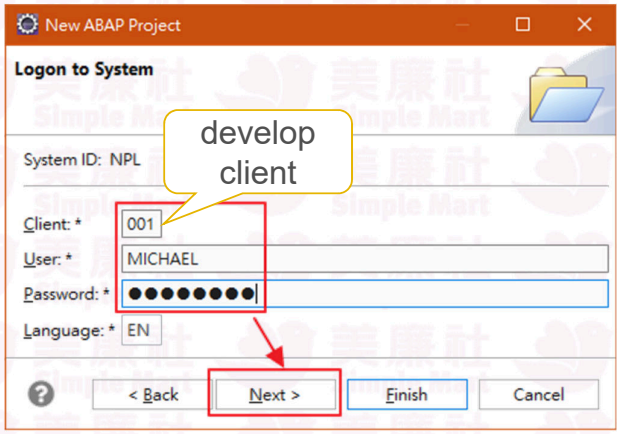
Set Server Connection – Step 1



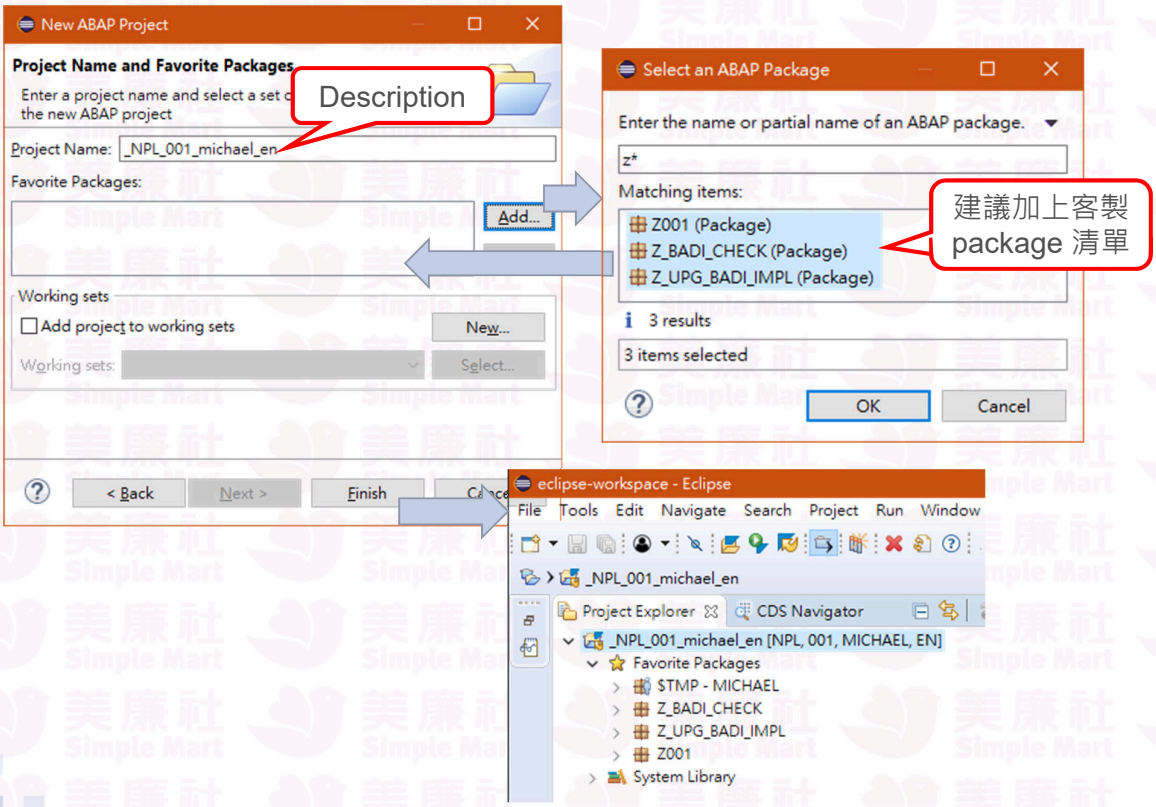


Set Server Connection – Step 2





Set Server Connection – Step 3



© 2018 NTT DATA Taiwan Co., Ltd.

NTT DATA

Introduction to the Eclipse/ADT

✓Install and Setup Eclipse

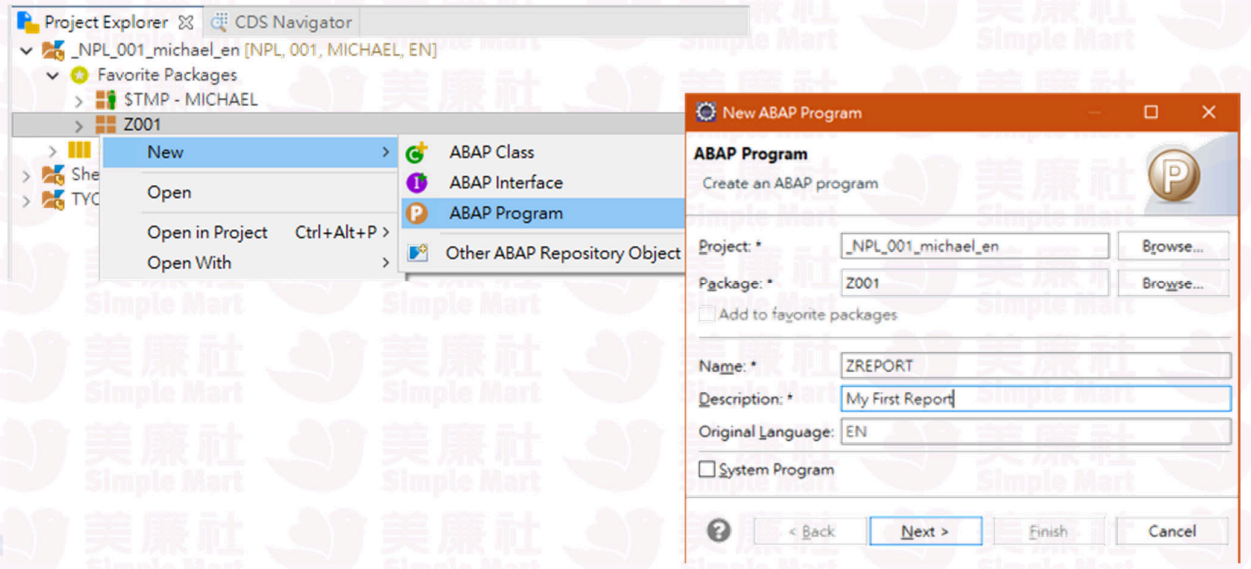
✓Developing Programs in Eclipse

© 2018 NTT DATA Taiwan Co., Ltd.

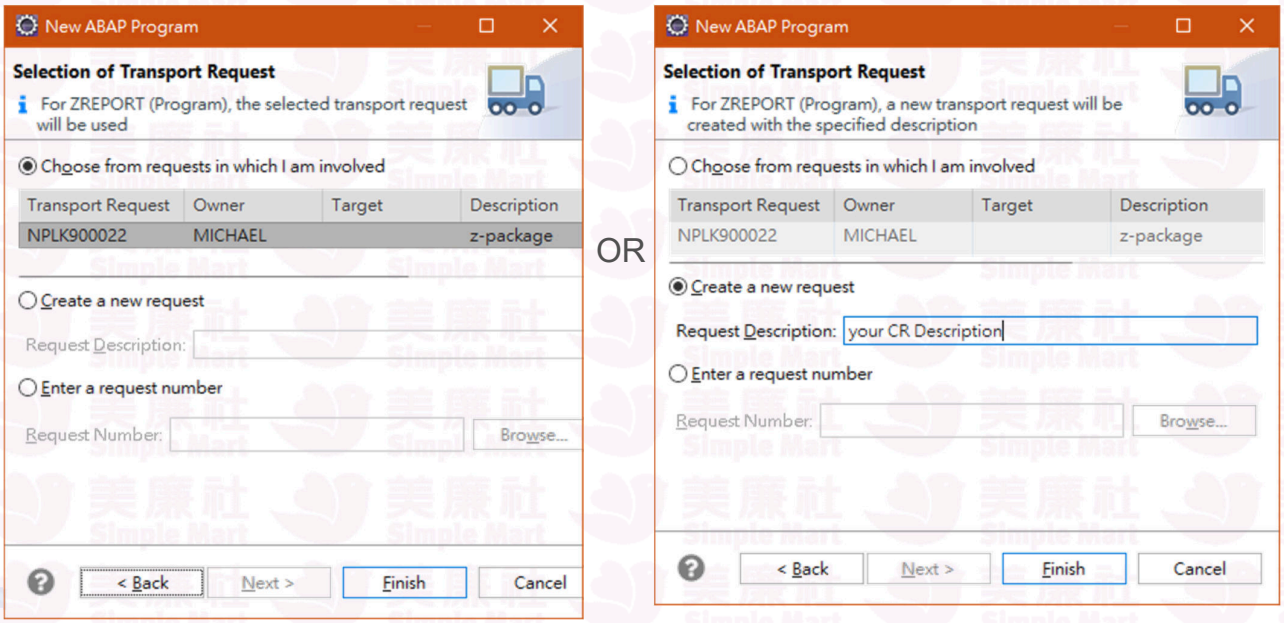
NTT DATA

Create an ABAP Program by Eclipse

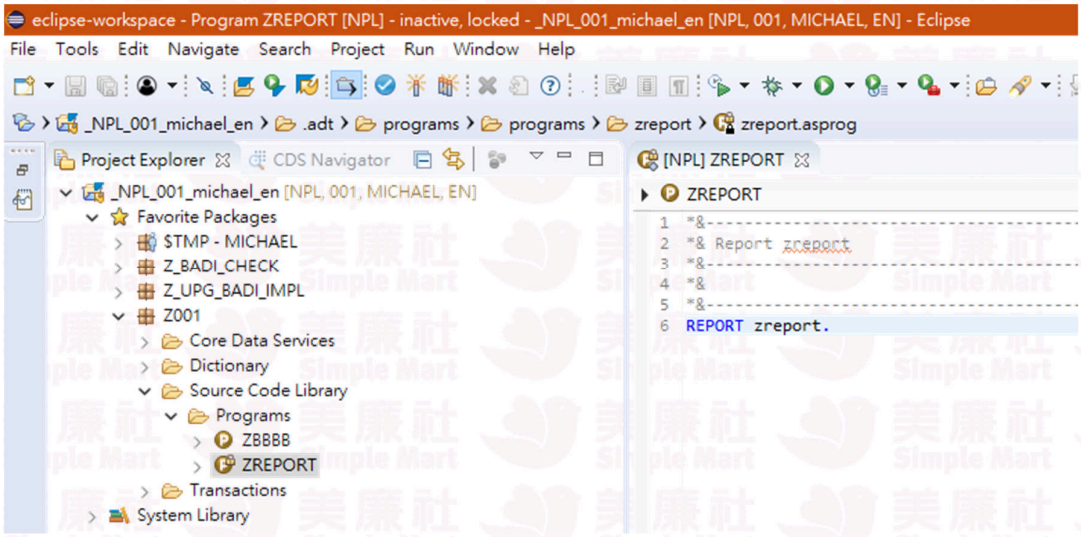
- $TMP：測試用，無法上傳至正式環境
- Zxxx：客製開發使用



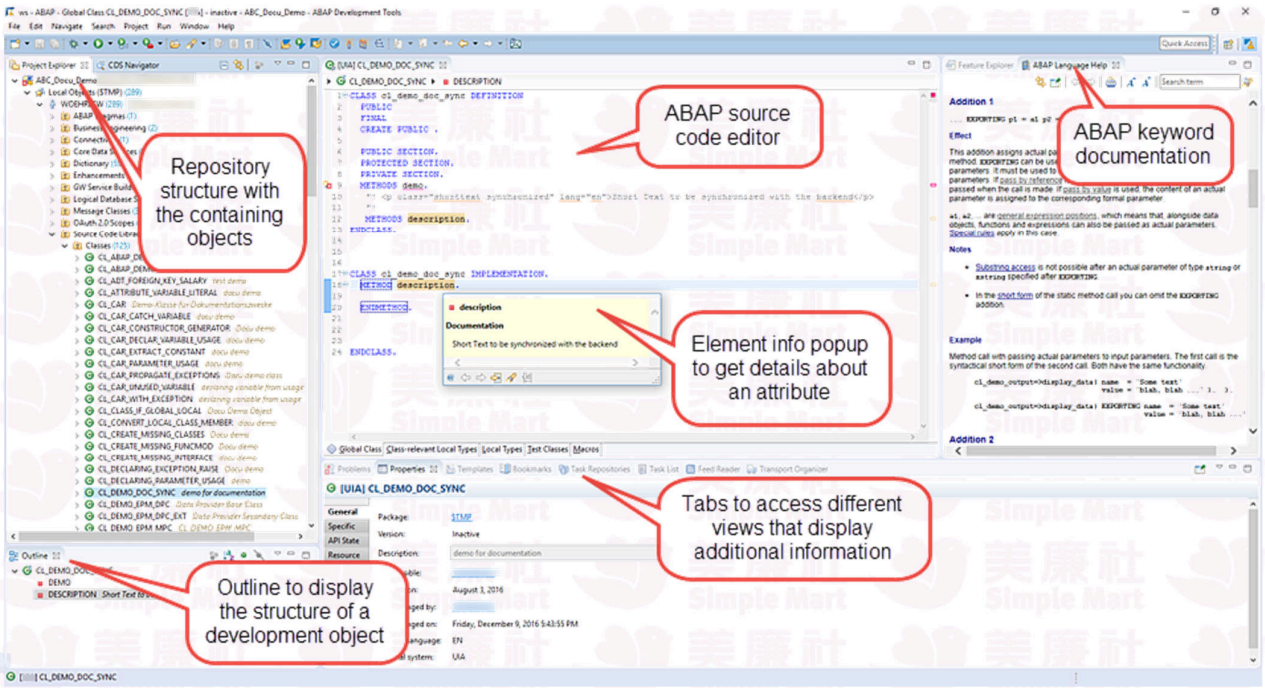
Assign CR



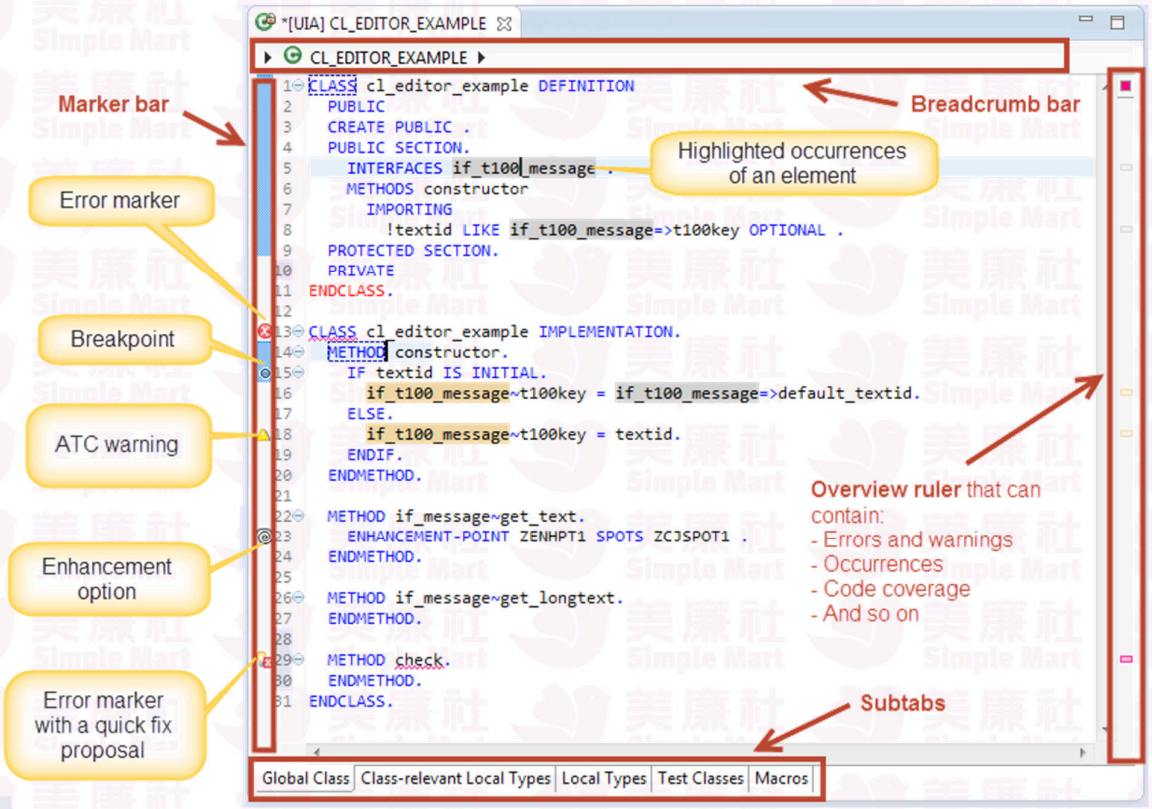
Start Write ABAP Code in Eclipse



Example screen when editing an ABAP Code with ADT

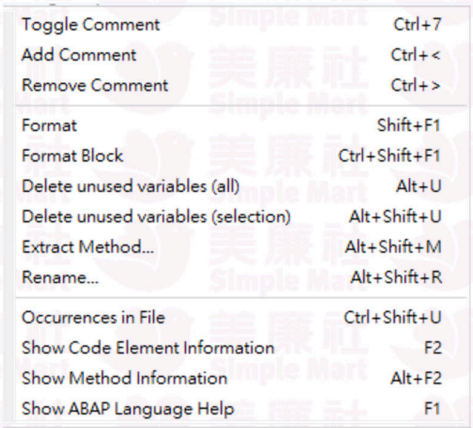


ABAP Source Code Editor

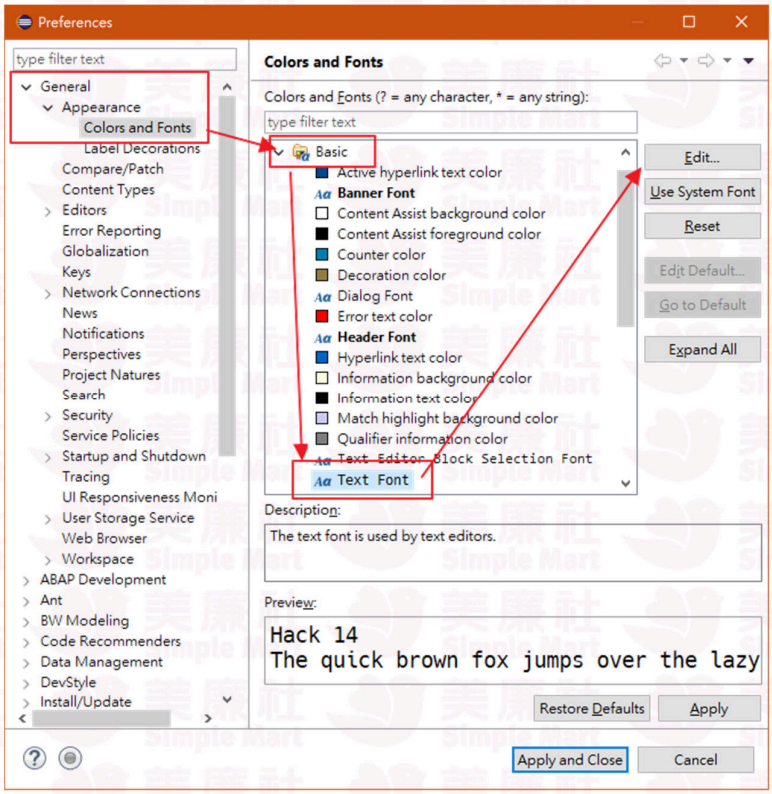


Hot Key in ABAP Editor(Eclipse)

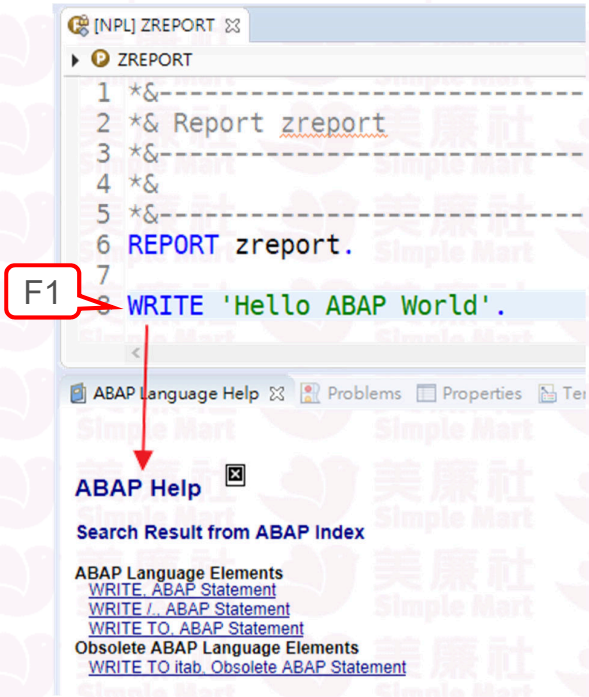
- Hot key
  - Ctrl + Shift + , 區塊註解
  - Ctrl + Shift + . 取消區塊註解
  - Alt + / 開啟字彙選擇框（自動完成框; SE80: Ctrl + space）
  - Ctrl + L 跳到指定行號（SE80: Ctrl + O）
  - Ctrl + Shift + F 自動排版
  - Shift + F1 Pretty Printer
  - Ctrl + 1 跳出修正建議框
  - Ctrl + F2 syntax check
  - Ctrl + D 刪除一行
  - F3 (SE80: Double-click / F2)
  - Alt + <- (左方向鍵) (SE80: F3(back))



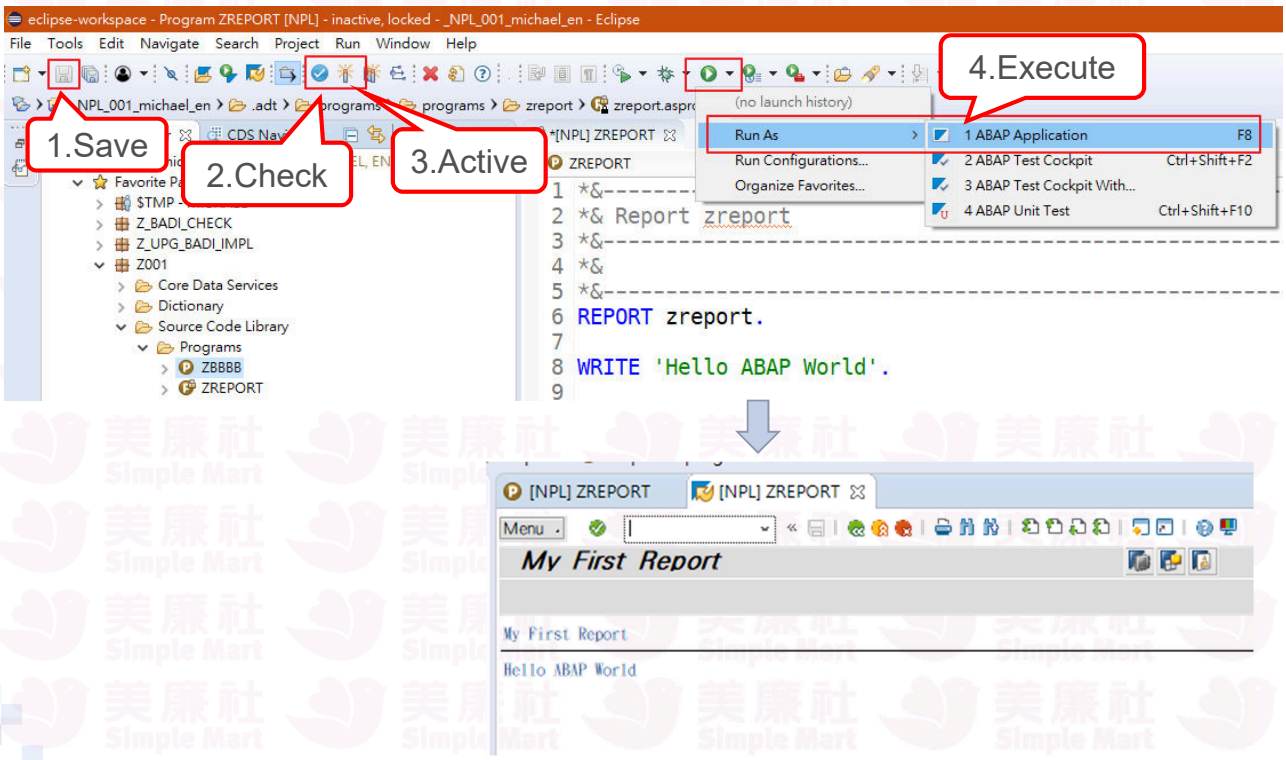
ABAP Source Code Editor



Keyword Documentation in the ABAP Editor



Active / Execute ABAP Report

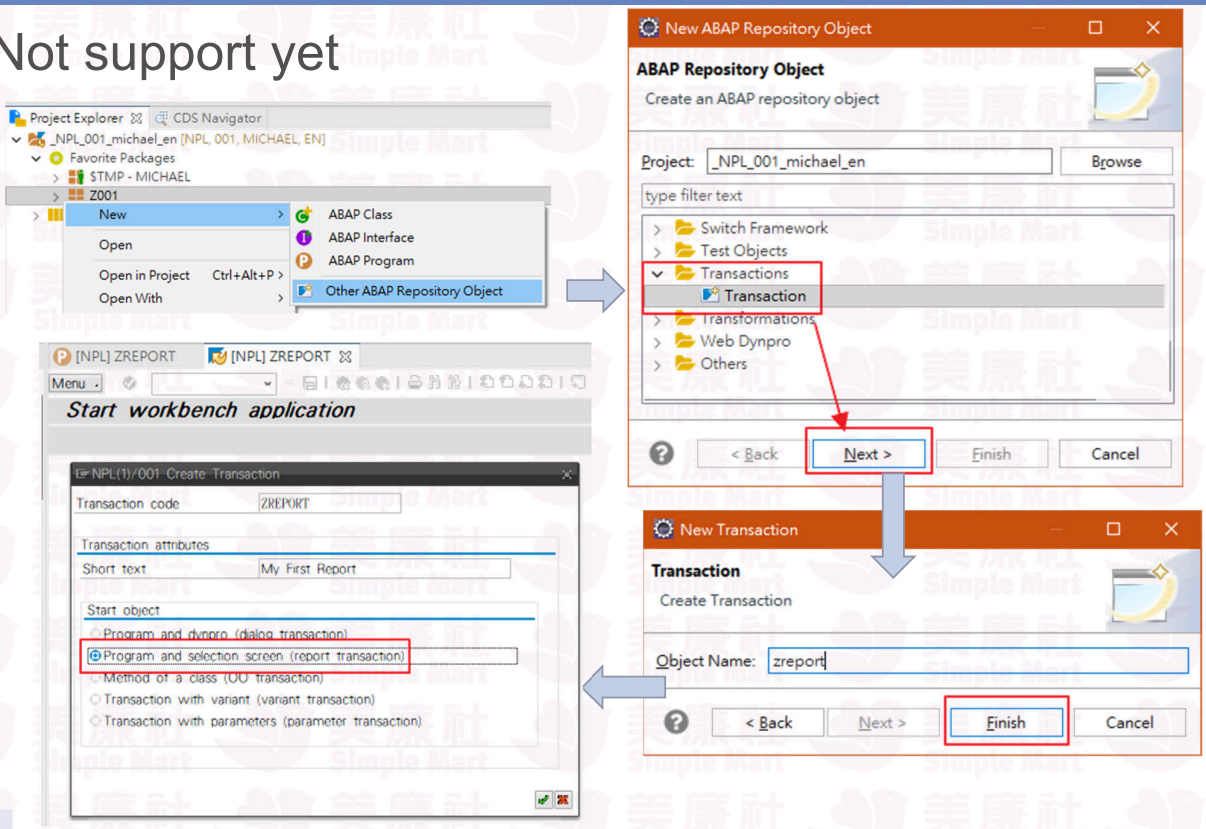


© 2018 NTT DATA Taiwan Co., Ltd.

NTT DATA

Creating a Transaction (Step 1)

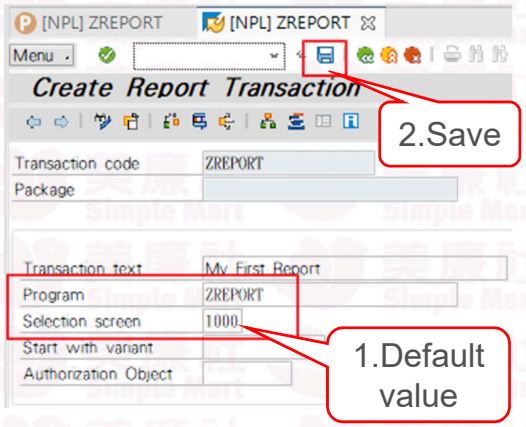
- Not support yet



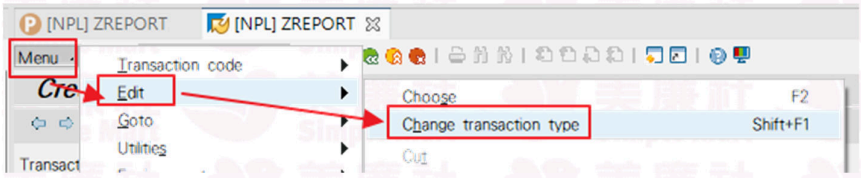
© 2018 NTT DATA Taiwan Co., Ltd.

NTT DATA

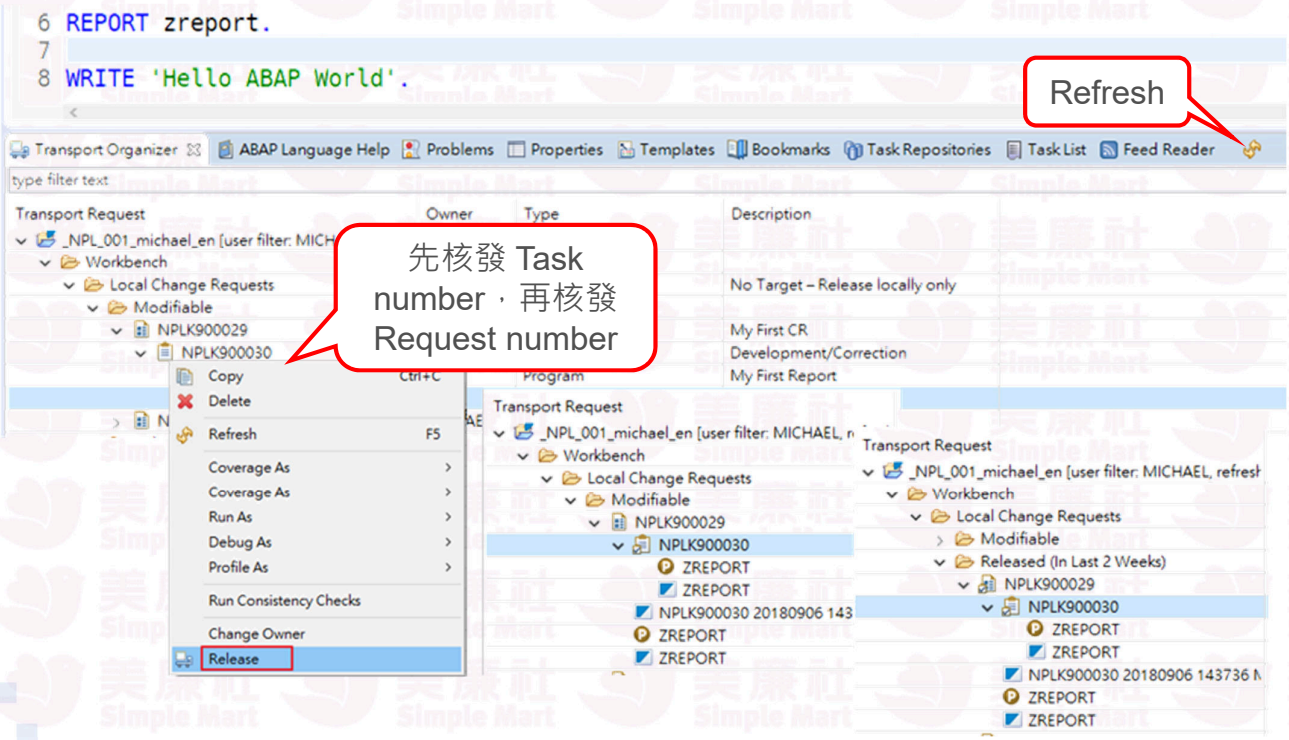
Creating a Transaction (Step 2)



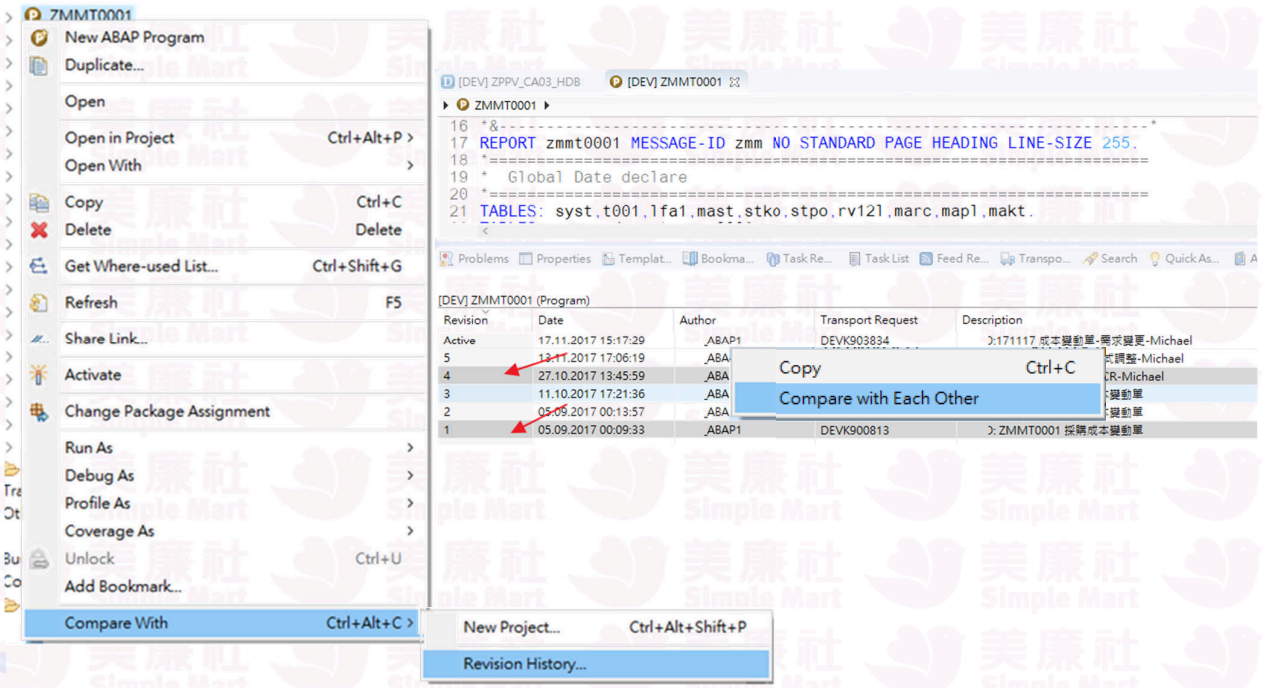
If transaction type is error



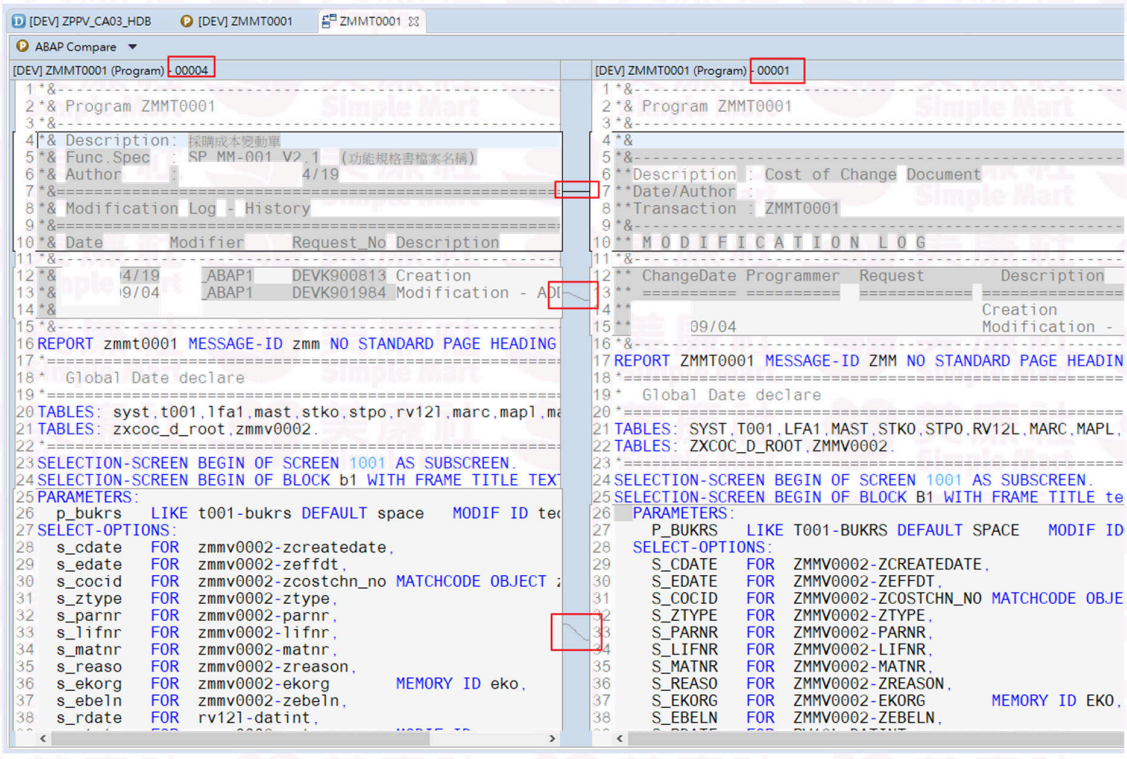
Release CR in Eclipse



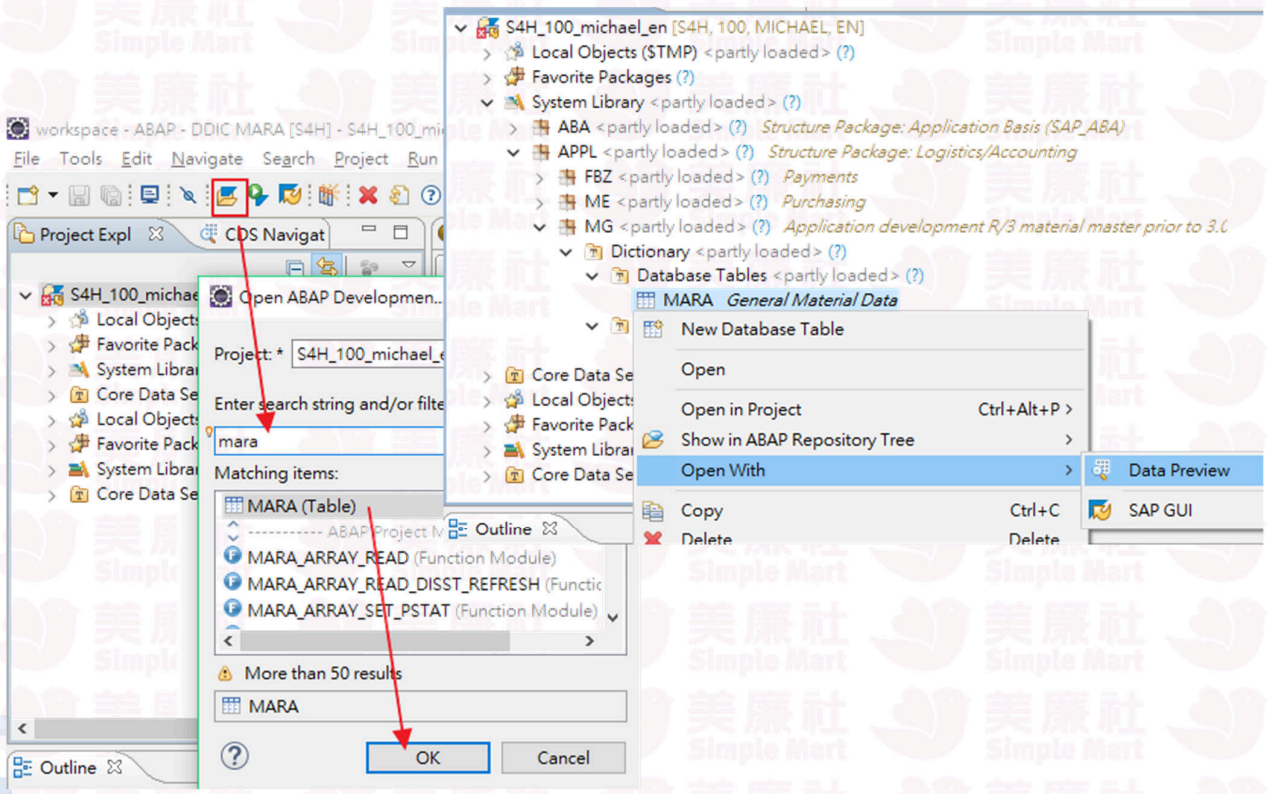
Version Compare in Eclipse- 1



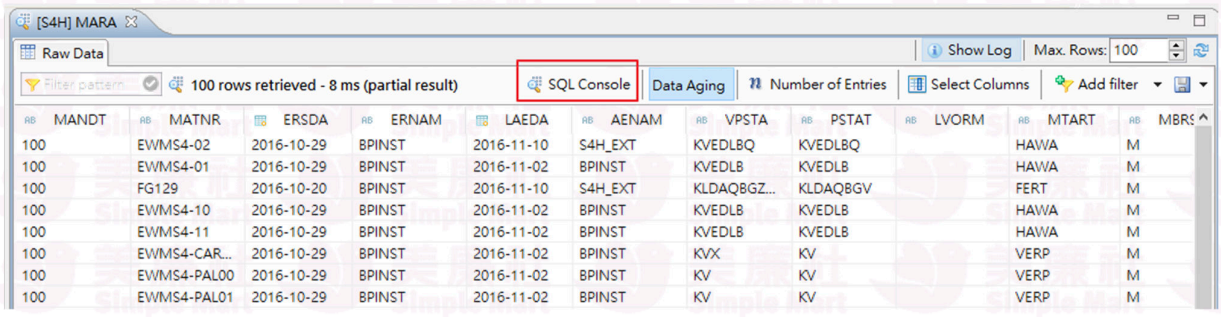
Version Compare in Eclipse- 2

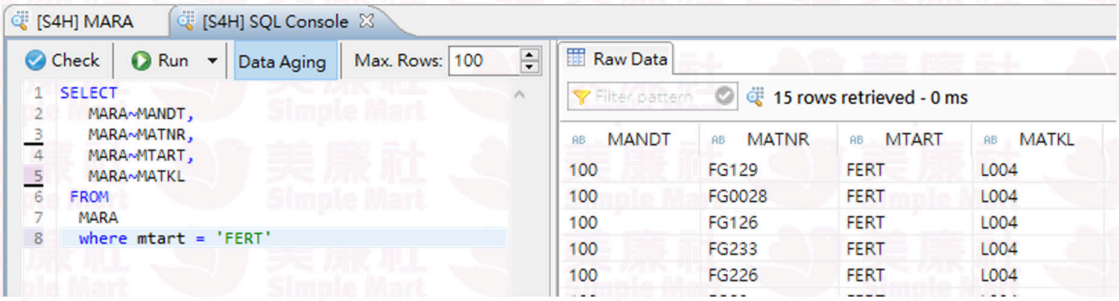


Data Preview - 1



Data Preview - 2





Code-to-Data with ABAP

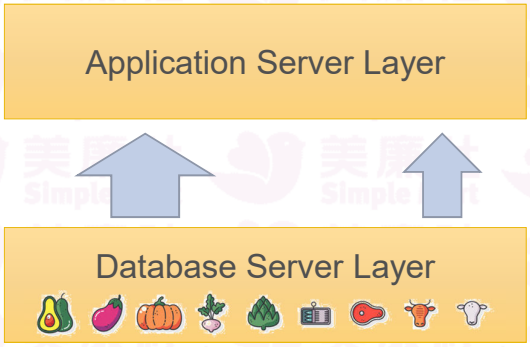
✓Overview

✓ABAP CDS (Core Data Service) View

© 2018 NTT DATA Taiwan Co., Ltd.

New Programming Approach

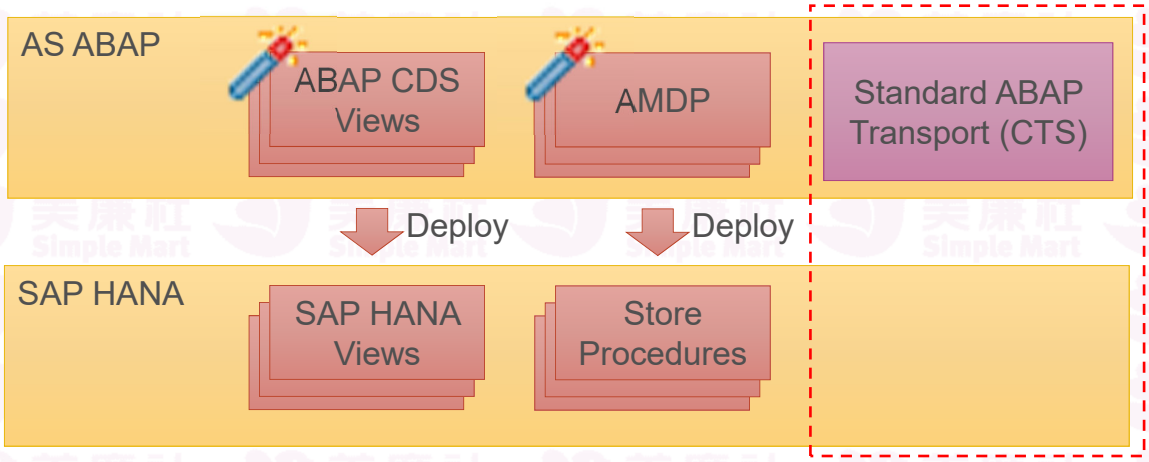
Classical approach
(data to code)



Future approach
(code to data)

© 2018 NTT DATA Taiwan Co., Ltd.

Overview



© 2018 NTT DATA Taiwan Co., Ltd.

Code-to-Data with ABAP

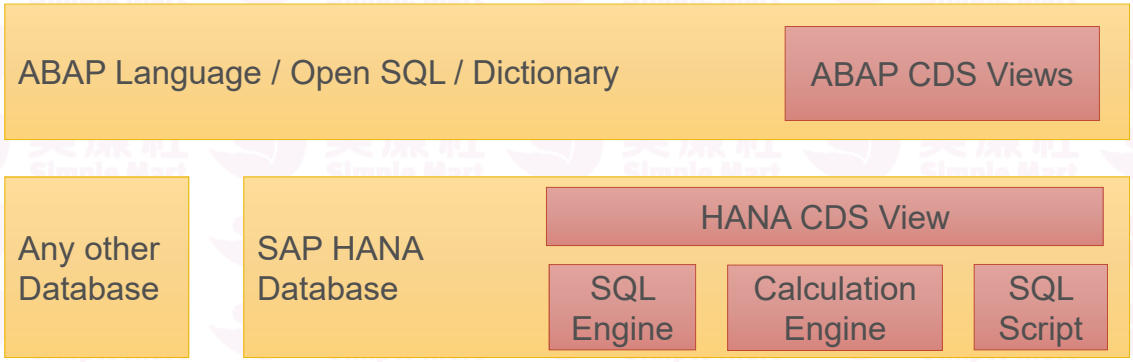
✓Overview

✓ABAP CDS (Core Data Service) View

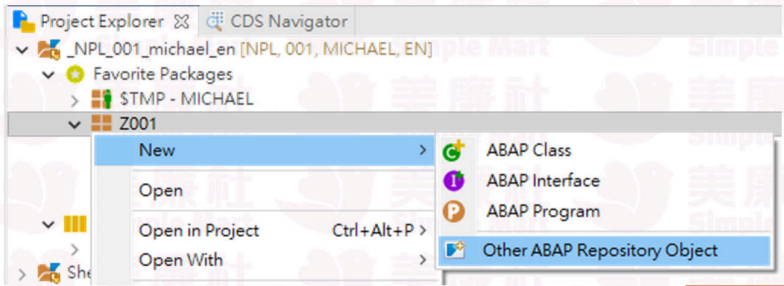
© 2018 NTT DATA Taiwan Co., Ltd.

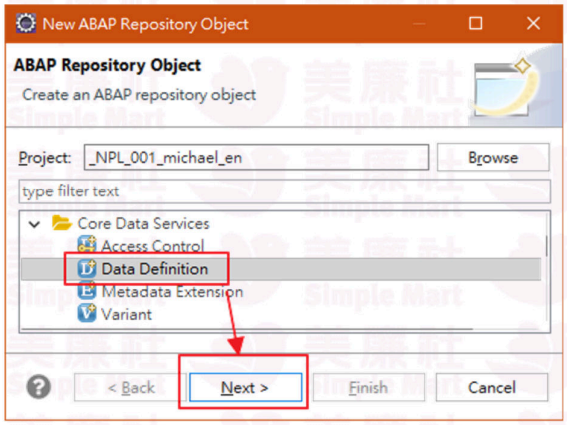
Core Data Service @ABAP

- Support pushing calculations to the DB layer using extended view functionality
- Simplify consumption of relational data models
  - Definition of semantically enriched data models provided as CDS views in the ABAP Dictionary
  - Open SQL on CDS views

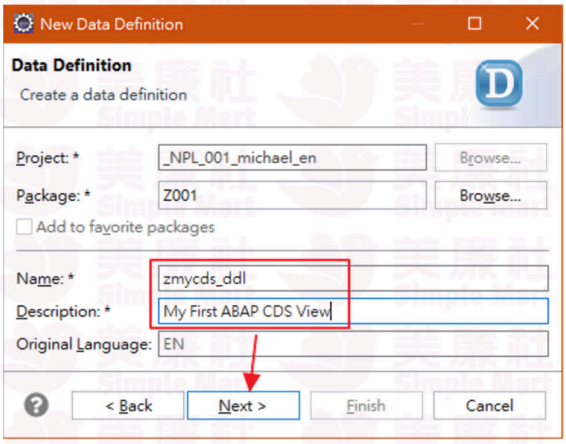


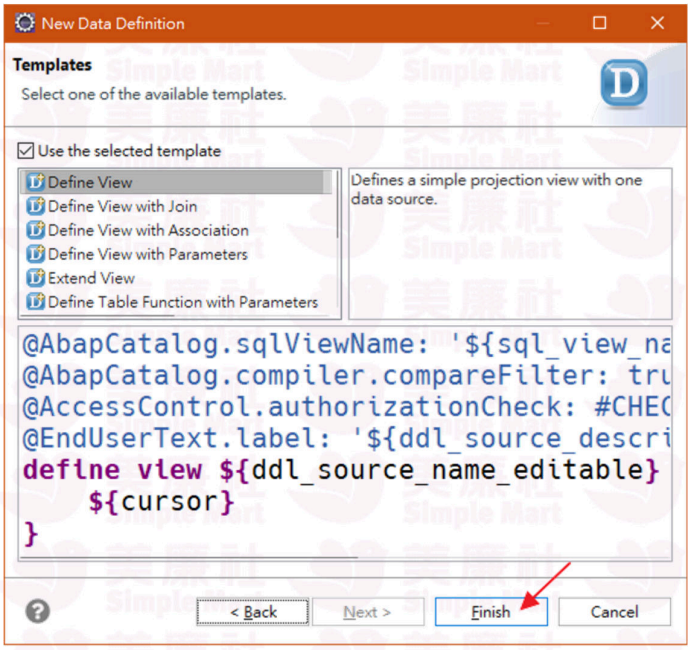
Create ABAP CDS View (Step1)



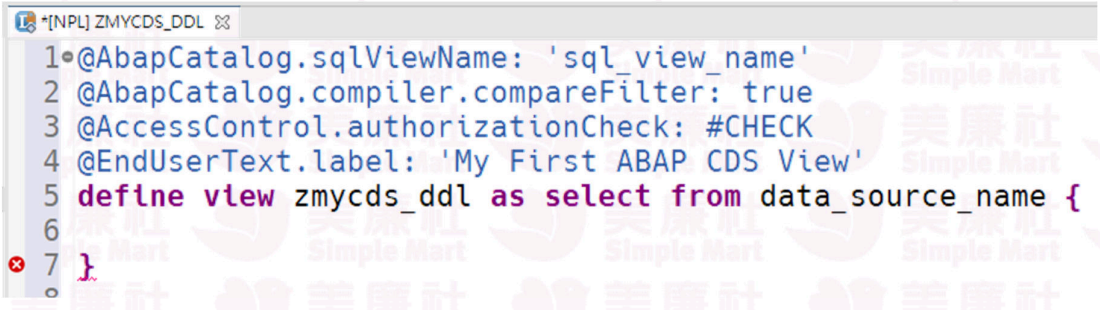


Create ABAP CDS View (Step2)

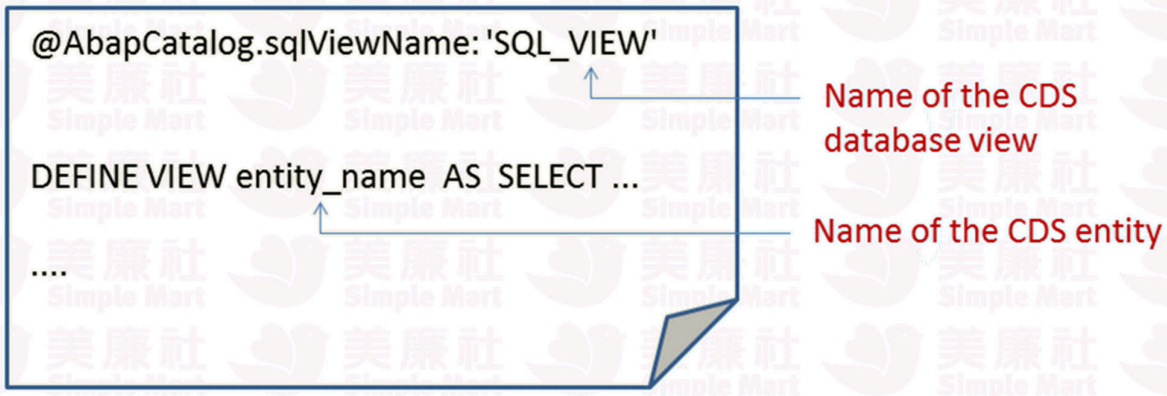




Create ABAP CDS View (Result)



ABAP CDS View Definition & ABAP Access




```
@AbapCatalog.sqlViewName: 'ZCUSTOMER'
define view zcust_book_view_entity as
  select from scustom join sbook
    on scustom.id = sbook.customid
{
  scustom.id,
  scustom.name,
  sbook.bookid
}
```

```
REPORT zdemo.

SELECT id, name, bookid
  FROM zcustomer
  INTO TABLE @DATA(result_data2).

SELECT id, name, bookid
  FROM zcust_book_view_entity
  INTO TABLE @DATA(result_data).
```

© 2018 NTT DATA Taiwan Co., Ltd.

Use Comments in CDS

- '--' 或 '//' 表示該行註解的開始
- Hot key : Ctrl + 7
- '/*' … '*/' 表示區塊(跨行)註解

```
@AbapCatalog.sqlViewName: 'zcds_test'  -- for SE11 查詢時的名稱
@AbapCatalog.compiler.compareFilter: true
@AccessControl.authorizationCheck: #CHECK -- 與 access control 元件搭配使用
//@AccessControl.authorizationCheck: #NOT_REQUIRED
@EndUserText.label: 'test' -- for SE11 的元件說明
--@Metadata.allowExtensions: true    -- 允許欄位另外設定欄位說明
--@OData.publish: true     -- 自動生成對應的 OData Serview
-- test description
define view zcds_test_hdb
  as -- for ADT 查詢時的名稱
  select from       t001 as a
    left outer join t001k on  a.mandt = t001k.mandt
                          and a.bukrs = t001k.bwkey
{
  key a.mandt, -- key 暫時測試不出效果
      @EndUserText.label: 'Customer ID'
      a.bukrs,
      @EndUserText: {
        label: 'AAAA'
      }
      t001k.bwkey
      /* test
      descript */
}
```

© 2018 NTT DATA Taiwan Co., Ltd.

Use Annotations in CDS View

- Enrich CDS data models with additional metadata
  - Annotations begin with @
- Different ABAP specific information can be added
  - Name of the associated SQL / ABAP Dic view – mandatory
  - Buffering
  - Client dependency
  - Ref info. for amount and quantity columns
  - …

```
@AbapCatalog.sqlViewName: 'ZMYCDS'
@AbapCatalog.compiler.compareFilter: true
@AccessControl.authorizationCheck: #CHECK
@ClientDependent: true  //default
@EndUserText.label: 'My First ABAP CDS View'
define view zmycds_ddl as
select from spfli inner join sflight
  on spfli.mandt = sflight.mandt
 and spfli.carrid = sflight.carrid
 and spfli.connid = sflight.connid
{
  spfli.mandt,
  spfli.carrid as airlilne,
  spfli.connid as fligntnumber,
  spfli.airpfrom,
  spfli.airpto,
  @Semantics.quantity: {
    unitOfMeasure: 'DISTID'
  } spfli.distance,    // type QUAN
  @Semantics.unitOfMeasure: true
    spfli.distid,      // type UNIT
  sflight.fldate as flight_date,
  @Semantics.amount: {
    currencyCode: 'CURRENCY'
  } sflight.price,     // type CURR
  @Semantics.currencyCode: true
    sflight.currency  // type CUKY
}
```


Support for Join, WHERE, Aggregations, Grouping, Filtering

- WHERE clause supported
- JOINs supported
  - INNER, LEFT OUTER, RIGHT OUTER join
- Aggregations, Grouping, Filtering support

```
define view zmycds_ddl as
select from sbook as book
  inner join scustom as customer
     on book.mandt = customer.mandt
    and book.customid = customer.id
{
  book.carrid,
  book.loccurkey as currcode,

  sum(book.loccuram) as total_amount,
  max(book.loccuram) as max_amount,
  min(book.loccuram) as min_amount,

  customer.region,
  customer.discount
}
where customer.country = 'US'
group by book.carrid, book.loccurkey,
  customer.region, customer.discount
having sum(book.loccuram) > 100
```

Unions

- Merge the results of queries using keyword UNION
  - Column lists of the queries must
    - Have equal number of columns
    - Be of compatible types in matching order
  - UNION implies a distinct semantic
  - UNION ALL dose not include distinct

```
define view zmycds_ddl as
select from scustom
{  id, form, name,
   street, postcode, city,
   country, postbox, custtype
} where postbox <> ''
union
select from scustom
{  id, form, name,
   street, postcode, city,
   country, postbox, custtype
} where custtype = 'B'
```

© 2018 NTT DATA Taiwan Co., Ltd.

CDS - System variable

Variable	Value in Open SQL accesses
$session.user	Current user name, nominal value of the ABAP system field sy-uname.
$session.client	Current client, nominal value of the ABAP system field sy-mandt.
$session.system_language	Language of the current text environment, nominal value of the ABAP system field sy-langu

```
@AbapCatalog.sqlViewName: 'DEMO_CDS_SESSVAR'
@AccessControl.authorizationCheck: #NOT_REQUIRED
define view demo_cds_session_variables
  as select from
    demo_expressions
  {
    id,
    $session.user            as system_user,
    $session.client          as system_client,
    $session.system_language as system_language,
    $session.system_date     as system_date
  }
```

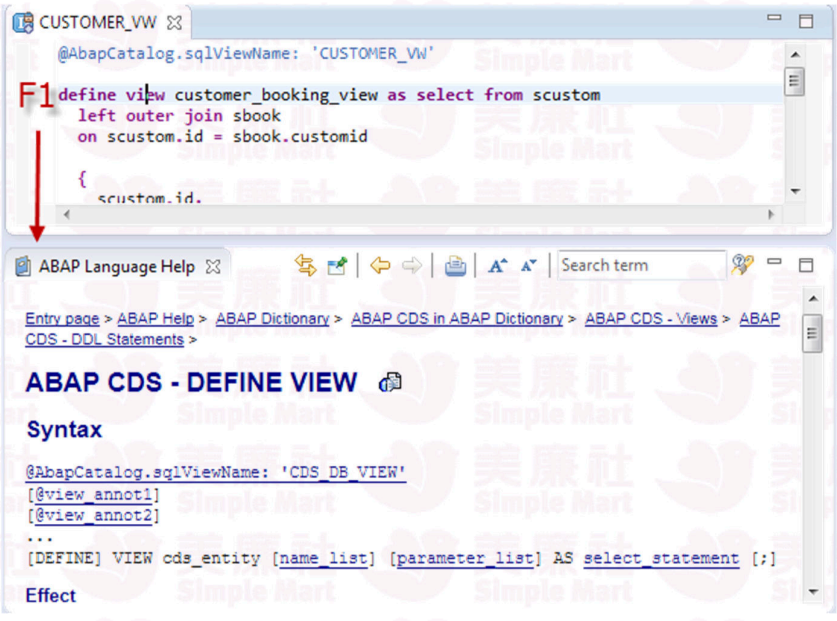
© 2018 NTT DATA Taiwan Co., Ltd.

CDS Built-in Expressions

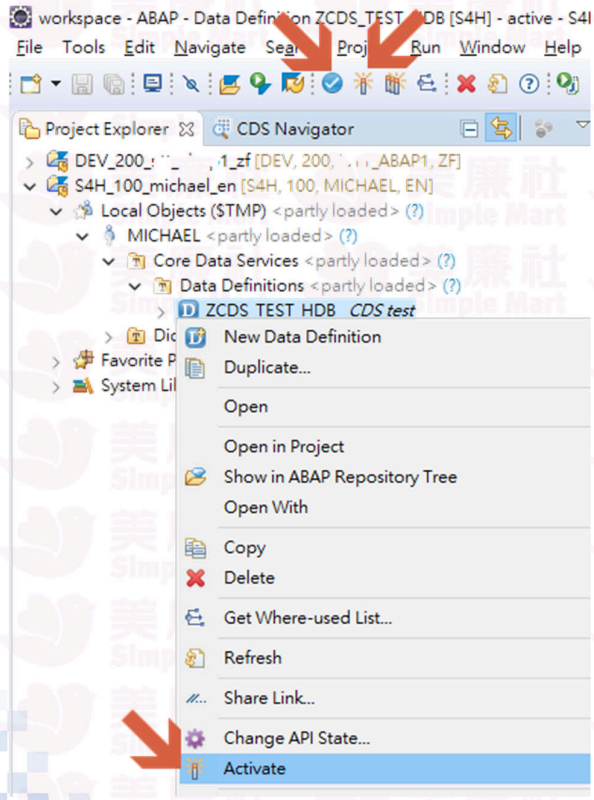
Generic SQL expression	Conversion functions	String functions	Arithmetic functions	Date and time functions
Simple- / searched-Case Coalesce	Unit conversion Currency Conversion Cast Hex2Bin Bin2Hex Decimal shift	Concat Instr Left / Right Length Lpad / Rpad Ltrim / Rtrim Replace	Abs Ceil Div (integer) Division (Dec) Floor Mod Round	Days between Add days Add month Current tmstp Date is valid Tmstp is valid Seconds btw. Tmstps Add seconds

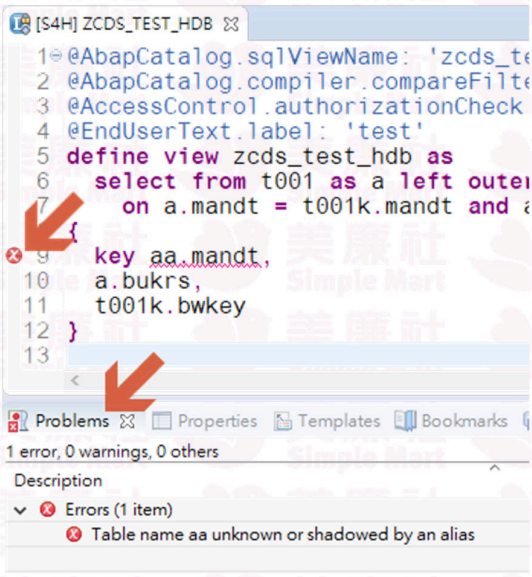
Getting Help for DDL Source Code

- Accessing CDS Language Reference (F1)

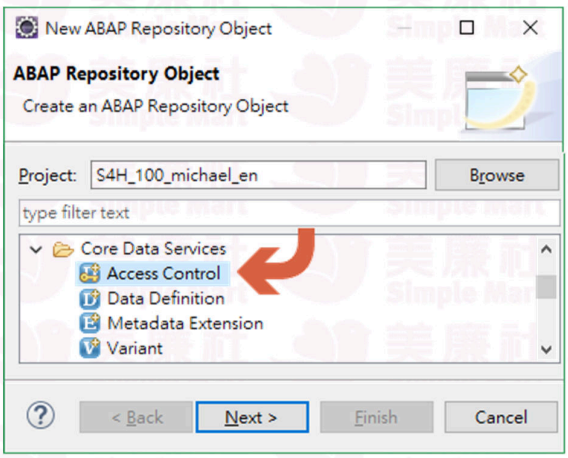


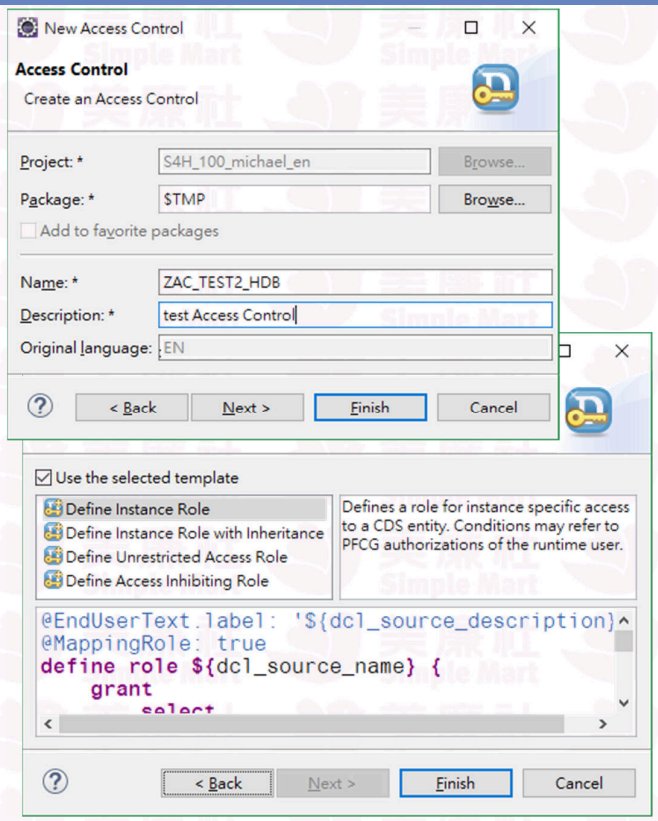
Activating Data Definitions & Display Activation Log





Adding Access Controls to CDS Entities





Access Control – Sample Code

```
@EndUserText.label: 'Access Control Test'
@MappingRole: true
define role Zac_Test_Hdb {
  grant select on zcds_test_hdb
    where ( bukrs )= aspect pfcg_auth( TR_BUKRS , BUKRS , ACTVT = '03' )
      and bukrs = '1000';
}
```

CDS Entity

Authority Object

Authority Field

Defining a CDS View with Input Parameters

```
define view zcds_with_para_ddl
  with parameters
    target_curr: s_currcode,
    smoker_factor: abap.curr( 15, 2 ),
    nosmoker_factor: abap.curr( 15, 2 )
as select from sbook
{
  case sbook.smoker
    when 'X'
      then loccuram * $parameters.smoker_factor
    else loccuram * $parameters.nosmoker_factor
  end as adj_amount,
  currency_conversion(
    amount => loccuram,
    source_currency => loccurkey,
    target_currency => $parameters.target_curr,
    exchange_rate_date => order_date
    ) as euro_amount
}
where sbook.loccurkey <> $parameters.target_curr
```

Data element or predefined

Not supported on all database !!

Using a CDS View with Parameters in a CDS View

```
define view zcds_with_para_ddl
  with parameters
    target_curr: s_currcode,
    smoker_factor: abap.curr( 15, 2 ),
    nosmoker_factor: abap.curr( 15, 2 )
as select from sbook
{
```

```
define view zcds_use_para_ddl as
select from zcds_with_para_ddl
  ( target_curr: 'JPY' ,
    smoker_factor: 2 ,
    nosmoker_factor: 1 )
{
  zcds_with_para_ddl.adj_amount,
  zcds_with_para_ddl.euro_amount
}
```

© 2018 NTT DATA Taiwan Co., Ltd.

Using a CDS View with Parameters in Open SQL

```
define view zcds_with_para_ddl
  with parameters
    target_curr: s_currcode,
    smoker_factor: abap.curr( 15, 2 ),
    nosmoker_factor: abap.curr( 15, 2 )
as select from sbook
{
```

```
SELECT * INTO TABLE @DATA(itab)
  FROM zcds_with_para_ddl( target_curr = 'USD',
                           smoker_factor = 2 ,
                           nosmoker_factor = 1 ).
```

© 2018 NTT DATA Taiwan Co., Ltd.

New Open SQL vs. ABAP CDS View

Use Open SQL if you:

- Need query in one piece of code only
- Access system fields, e.g., sy-mandt
- Need features only available in Open SQL, such as FOR ALL ENTRIES

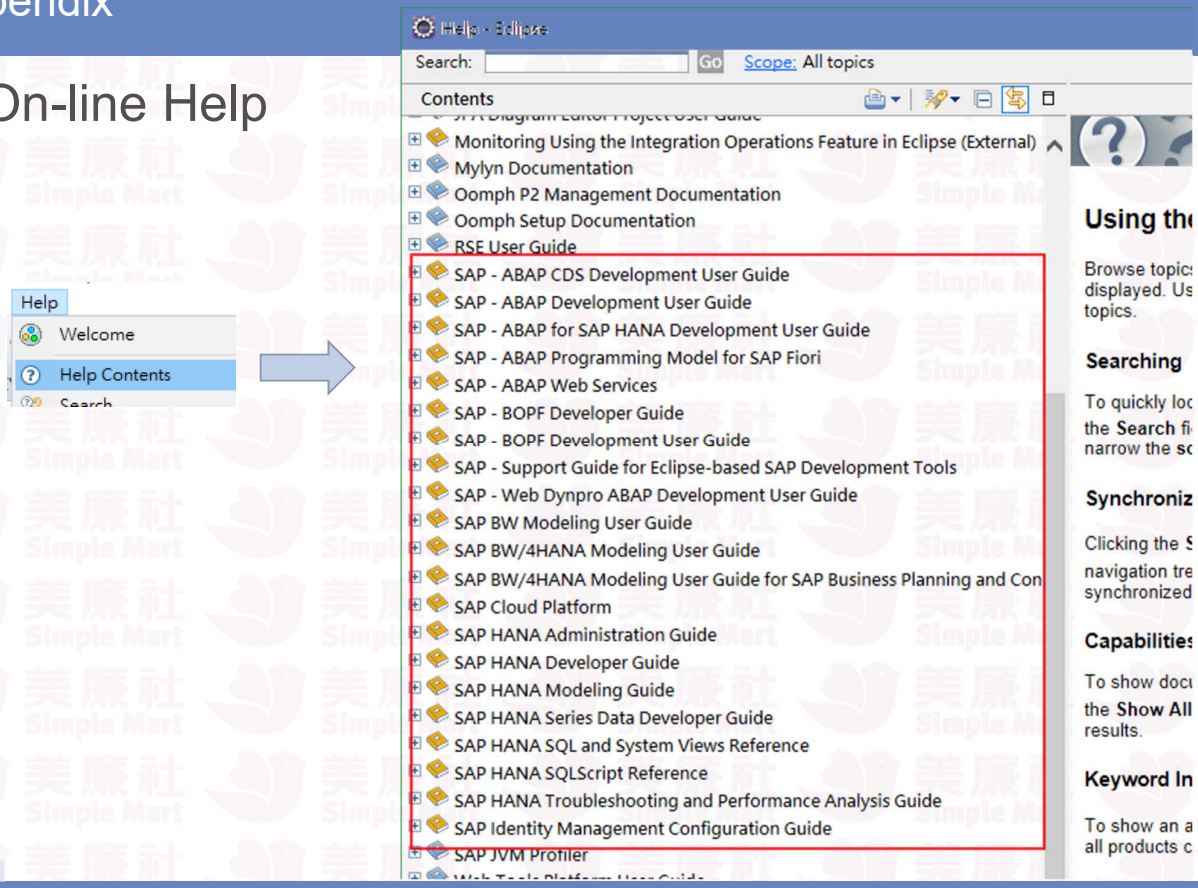
Use CDS views if you:

- Have a real re-use case – similar as Dictionary view
- Need features currently only available in DDL source like ASSOCIATION, UNION, UNION ALL

© 2018 NTT DATA Taiwan Co., Ltd.

Appendix

- On-line Help



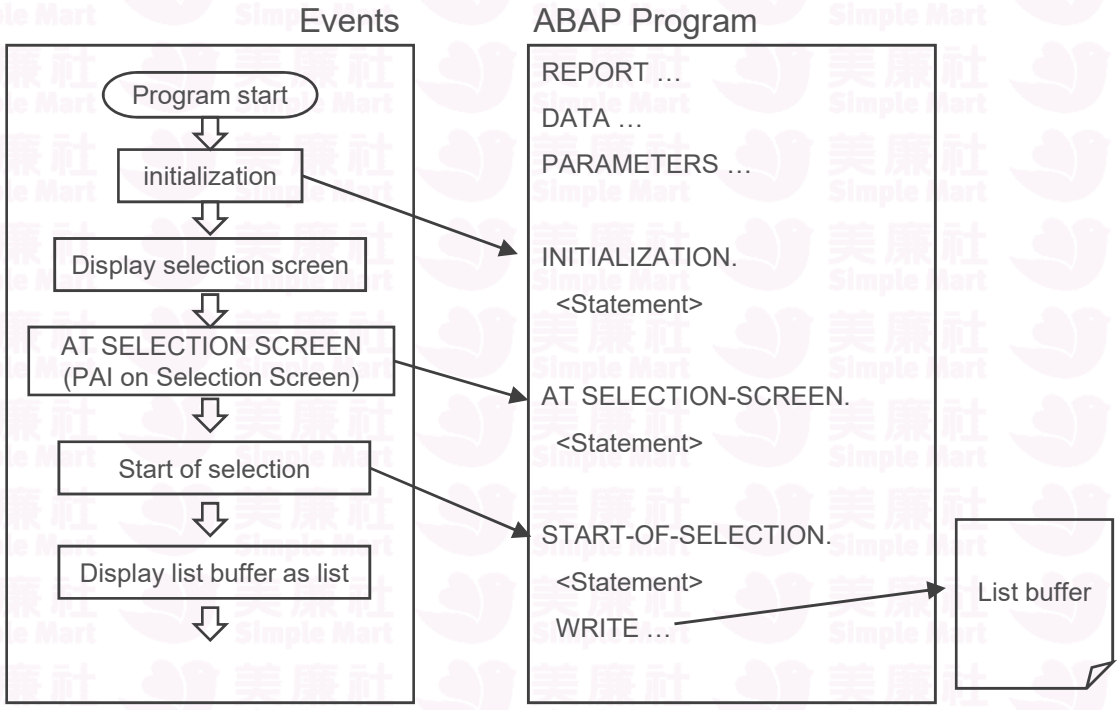
© 2018 NTT DATA Taiwan Co., Ltd.

User Dialog

- ✓ABAP Events
- ✓List
- ✓Selection Screen

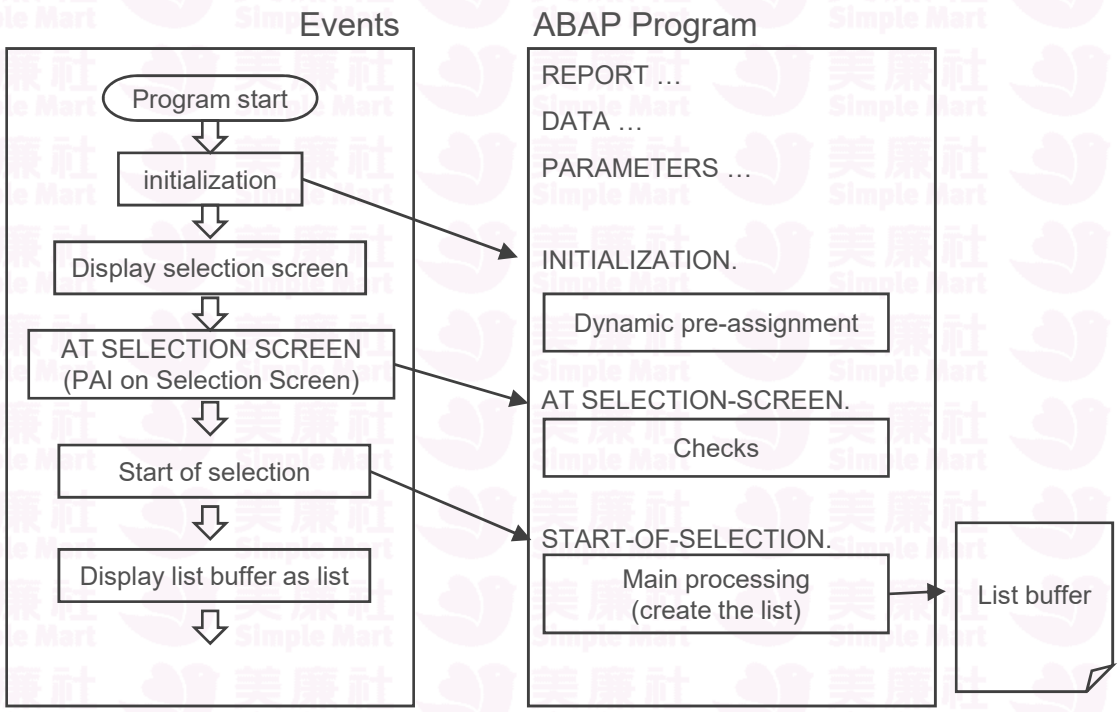
© 2018 NTT DATA Taiwan Co., Ltd.

Structures and Procedures of an Executable ABAP Program



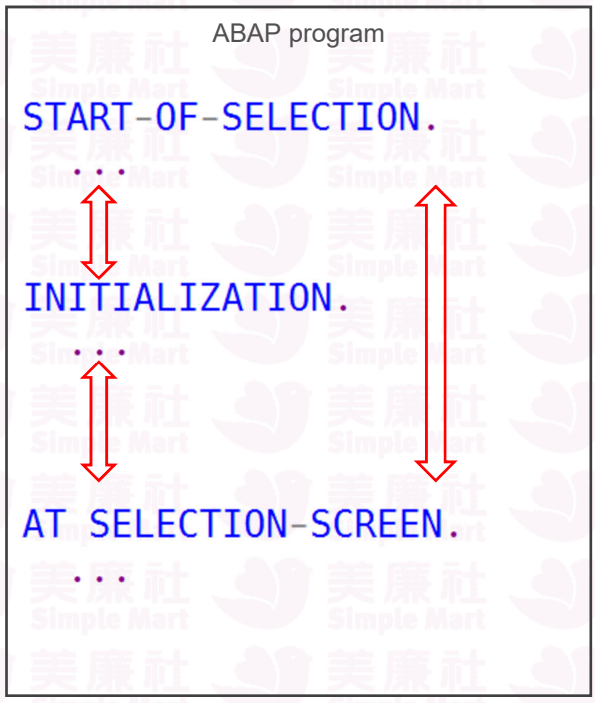
© 2018 NTT DATA Taiwan Co., Ltd.

Purpose of ABAP Events

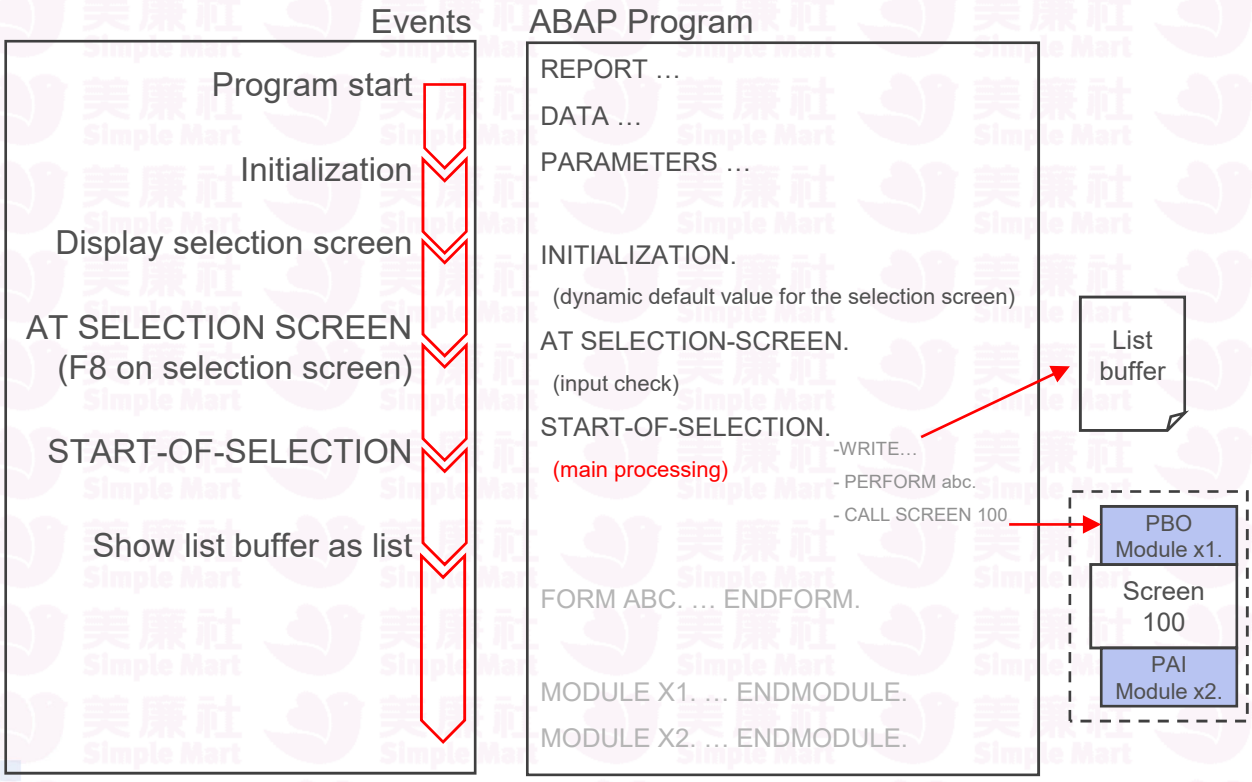


Event Block Characteristics

- Introduced by an event keyword
- Ended by the next processing block
- Cannot be nested
- Existence not absolutely necessary
- Sequence of event blocks in the source code is irrelevant
- Implicit standard event block in executable program: **START-OF-SELECTION**



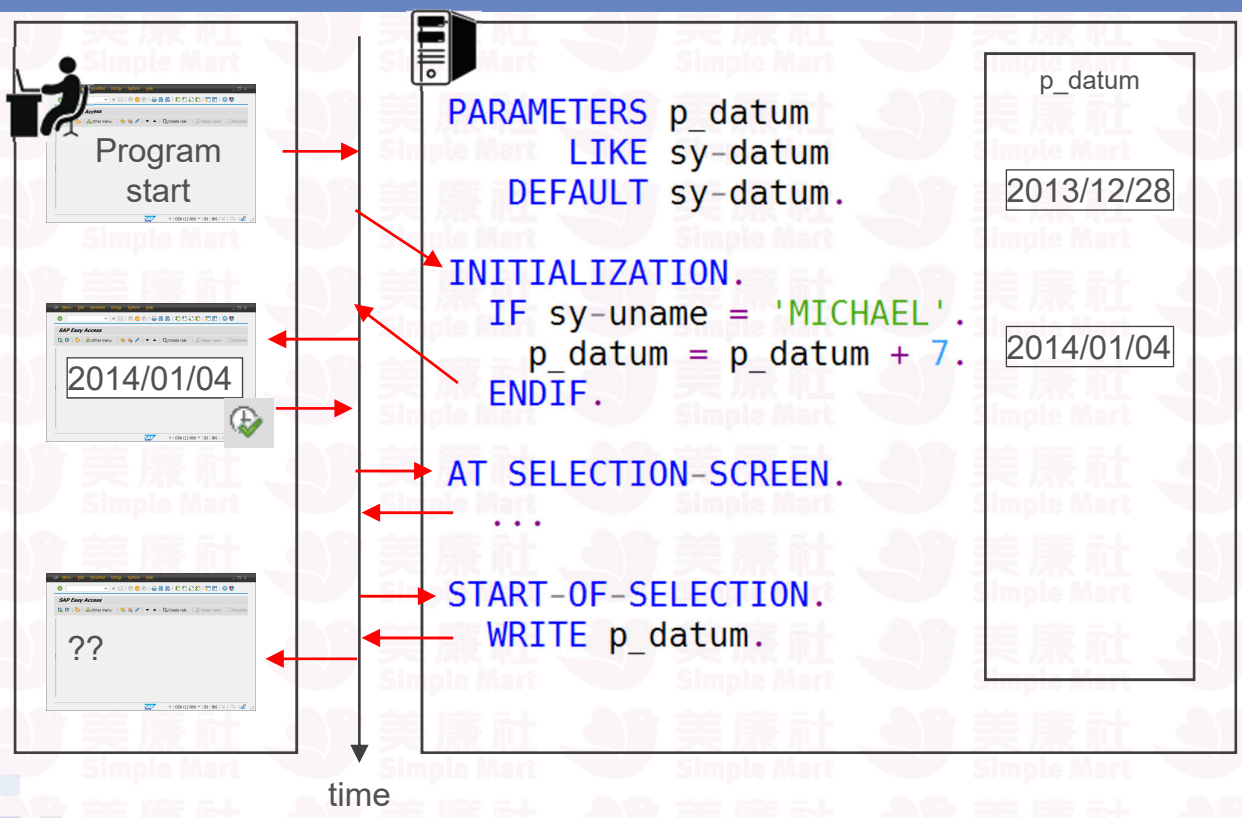
Program Structure (Overview)



© 2018 NTT DATA Taiwan Co., Ltd.

NTT DATA

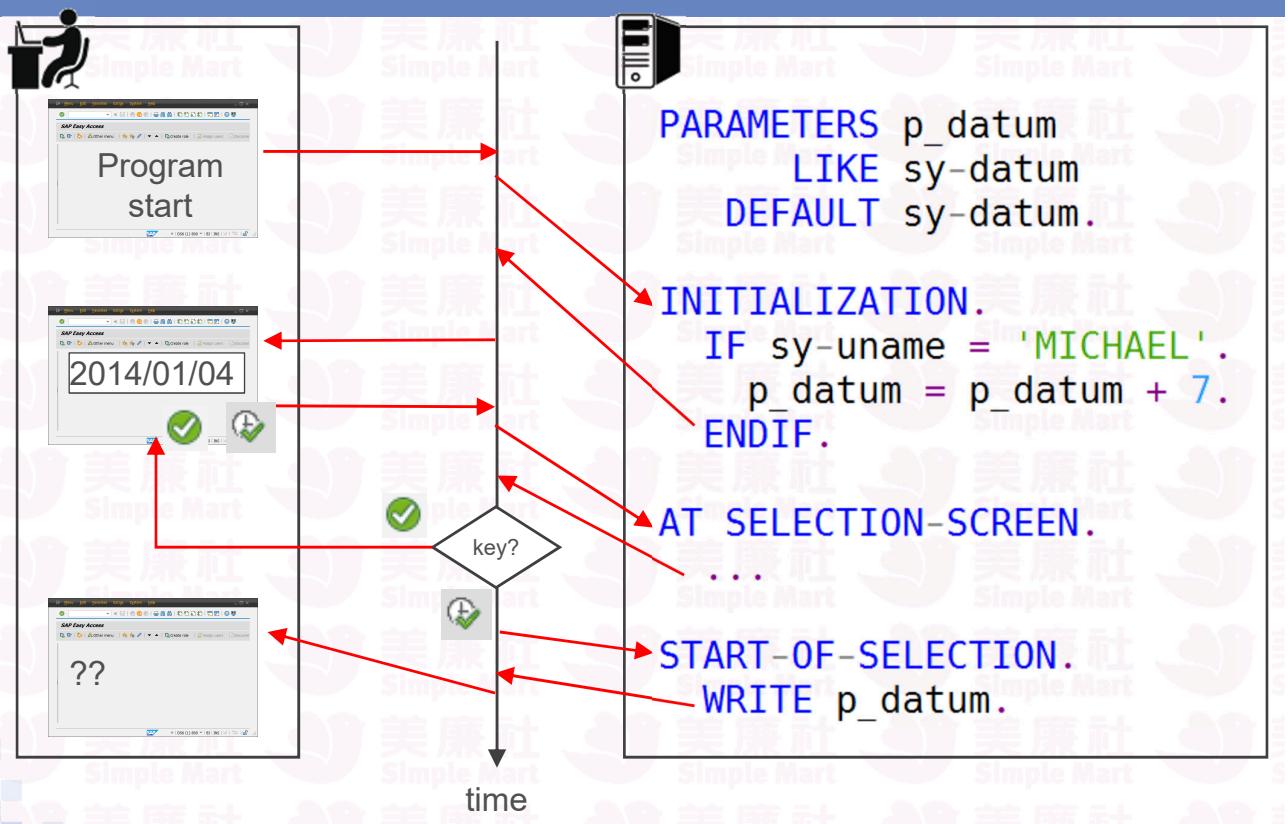
Example Program



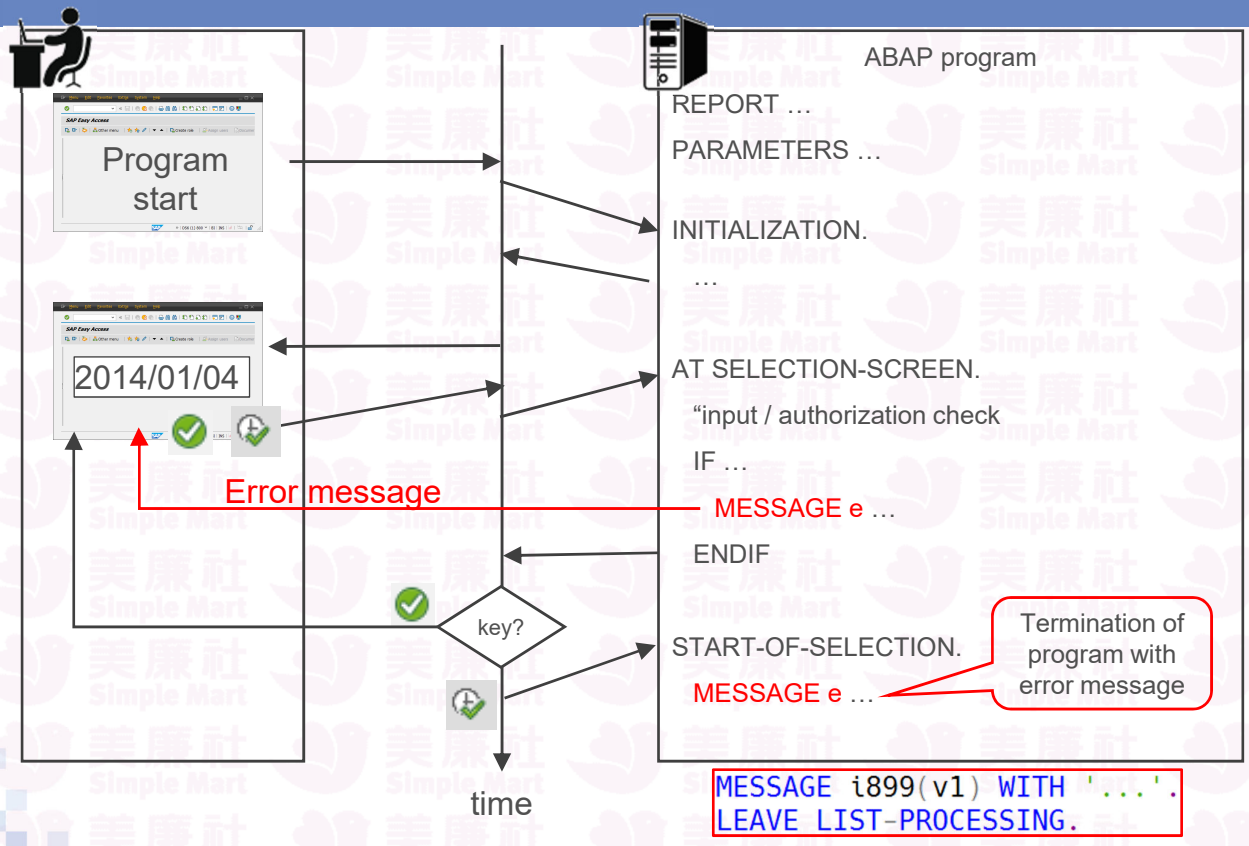
© 2018 NTT DATA Taiwan Co., Ltd.

NTT DATA

AT SELECTION-SCREEN Event



Error Dialogs in AT SELECTION-SCREEN



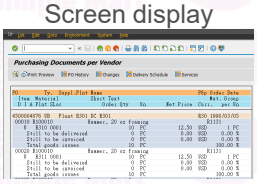
User Dialog

- ✓ABAP Events
- ✓List
- ✓Selection Screen

© 2018 NTT DATA Taiwan Co., Ltd.

Features of Lists

Screen display



Print

Prepared output acc. to user interface.

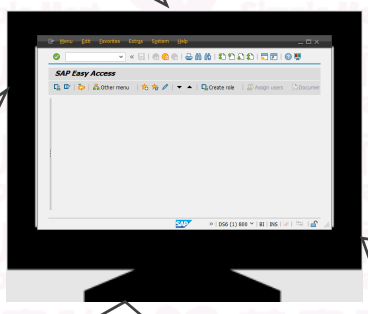
DD.MM.YYYY
MM/DD/YYYY
MM-DD-YYYY
YYYY.MM.DD
YYYY/MM/DD
YYYY-MM-DD
GYY.MM.DD (Japanese Date)
GYY/MM/DD (Japanese Date)
GYY-MM-DD (Japanese Date)
YYYY/MM/DD (Islamic Date 1)
YYYY/MM/DD (Islamic Date 2)
YYYY/MM/DD (Iranian Date)

24 Hour Format (Example: 12:05:10)
12 Hour Format (Example: 12:05:10 PM)
12 Hour Format (Example: 12:05:10 pm)
Hours from 0 to 11 (Example: 00:05:10 PM)
Hours from 0 to 11 (Example: 00:05:10 pm)

1,234,567.89
1 234 567,89
1.234.567,89

Language-dependent text display

物料编号
เลขที่วัสดุ
자재 번호
Număr material
Številka materiala
Broj materijala
رقم المادة
מק"ט
Číslo materiálu
Materialnummer
Material Number
Numéro d'article
Αριθμός Υλικού
Anyagszám
Codice materiale
品目コード
Materialennummer
Nr materiału
物料號碼
Artikelnummer
Materialnummer
Nº do material
Číslo materiálu
Номер материала
Número de material
Malzeme numarası
Nimikenumero
Materialnummer
Номер на материал
Broj materijala

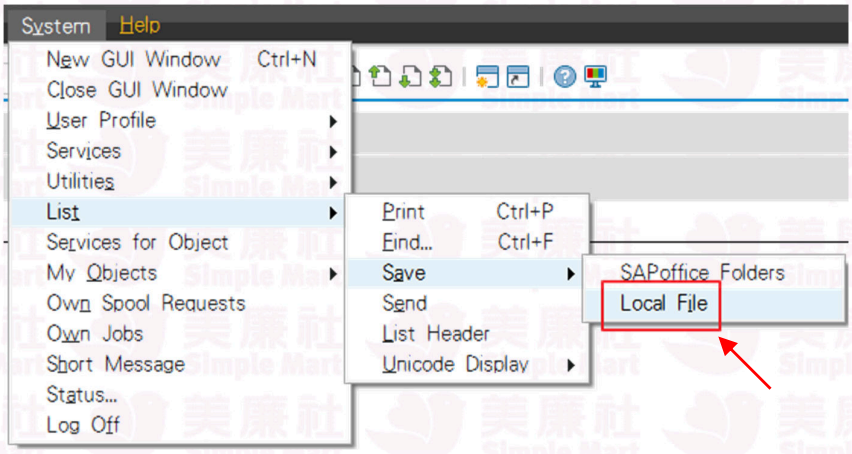


Formatted output:

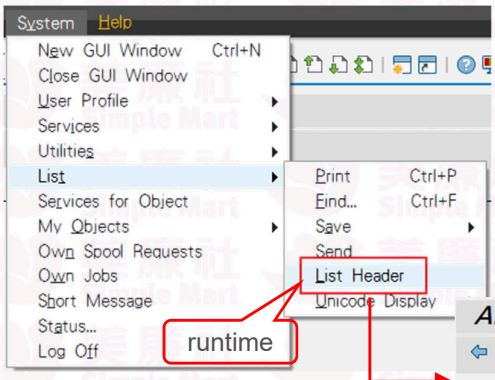
- Amounts acc. to their currency
- Lengths acc. To their unit

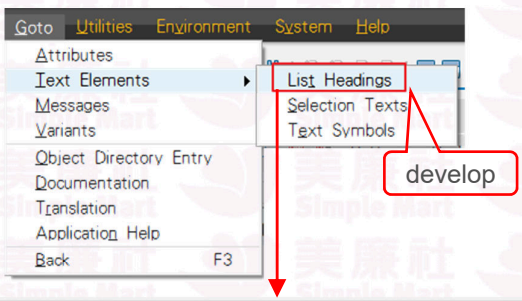
© 2018 NTT DATA Taiwan Co., Ltd.

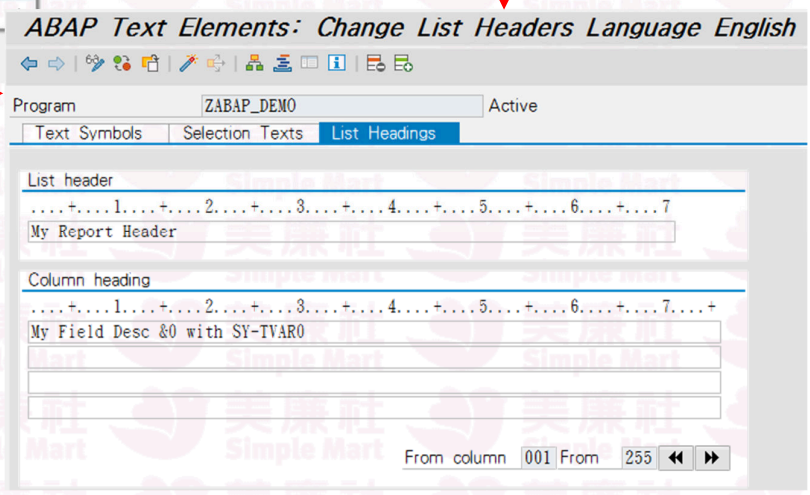
Standard List Functions



List and Column Headers







Create Page Header : TOP-OF-PAGE

```
REPORT zabap_demo NO STANDARD PAGE HEADING.

TOP-OF-PAGE.
  WRITE:/1 'Report creator:'(001), sy-uname,
         /1 'Create date:'(002), sy-datum,
         /1 'Create time:'(003), sy-uzeit.
  ULINE.

START-OF-SELECTION.
```

```
Report creator: MICHAEL
Create date: 2015/10/19
Create time: 15:55:16

AA  0017 NEW YORK           SAN FRANCISCO
AA  0064 SAN FRANCISCO      NEW YORK
```

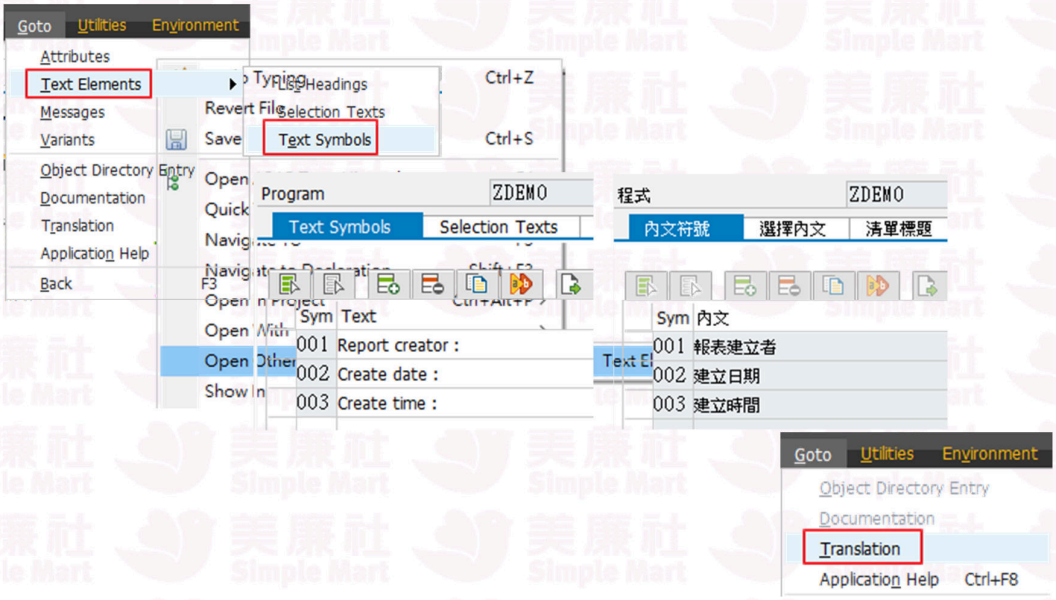

Creating a Footer : END-OF-PAGE

```
REPORT zabap_demo NO STANDARD PAGE HEADING
  LINE-COUNT 12(2).

END-OF-PAGE.
  ULINE.
  WRITE:/1 'Page:', sy-pagno.
```

```
Report creator: MICHAEL
Create date: 2015/10/19
Create time: 16:01:52

AA  0017 NEW YORK           SAN FRANCISCO
AA  0064 SAN FRANCISCO      NEW YORK
AC  0820 FRANKFURT/MAIN     MONTREAL-DORVAL
AF  0820 FRANKFURT/MAIN     FORT DE FRANCE
DL  1699 NEW YORK           SAN FRANCISCO
DL  1984 SAN FRANCISCO      NEW YORK

Page:     1
```

Multilingual Capabilities of Lists

```
WRITE:/1 'Report Creator'(001), sy-uname,
      /1 TEXT-002, sy-datlo,
      /1 'Create time'(003), sy-timlo.
```



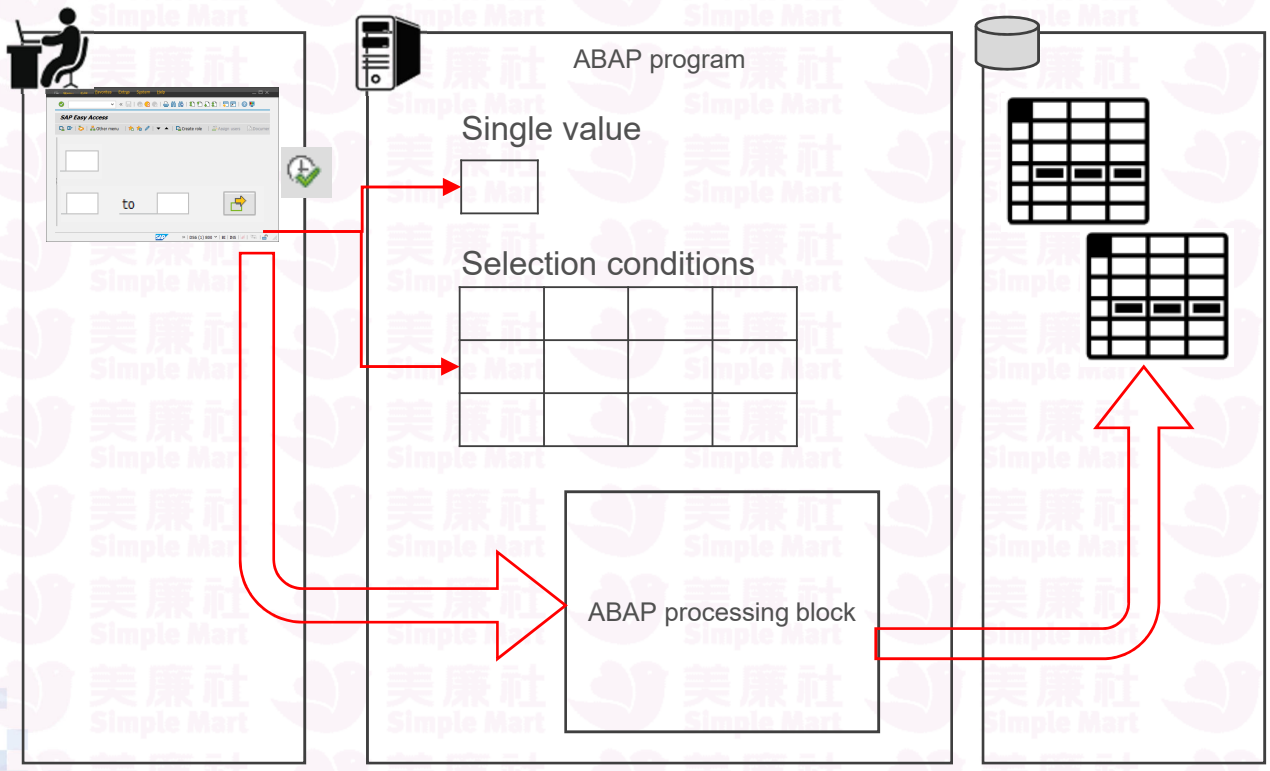
© 2018 NTT DATA Taiwan Co., Ltd.

User Dialog

- ✓ABAP Events
- ✓List
- ✓Selection Screen

© 2018 NTT DATA Taiwan Co., Ltd.

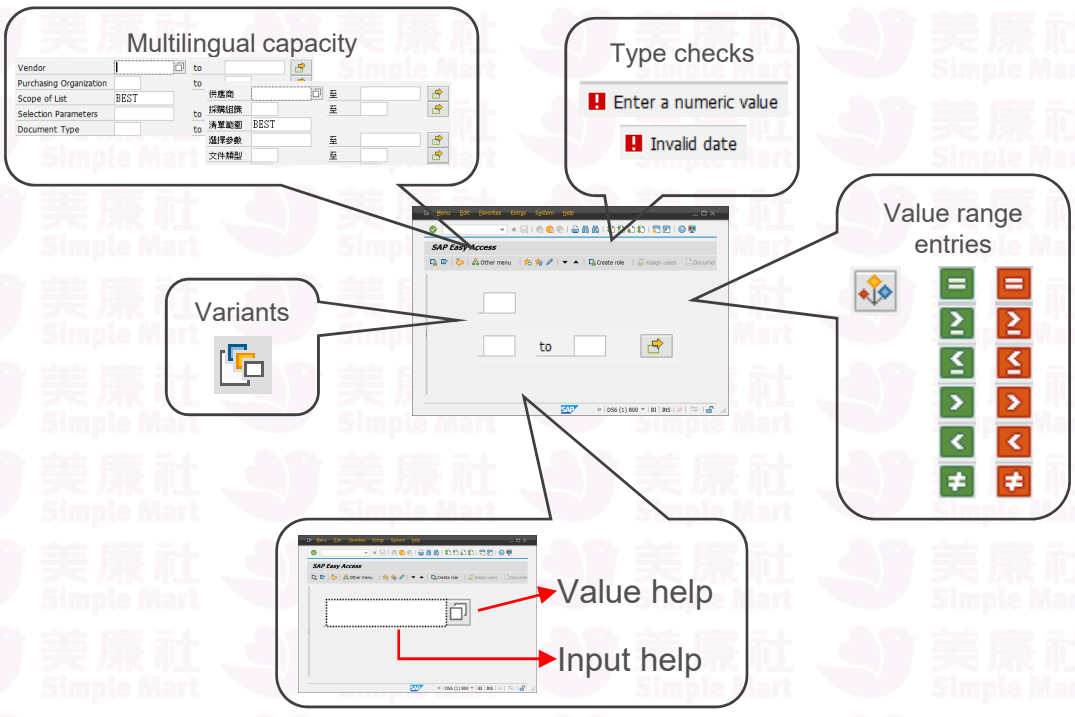
Architecture and Purpose of Selection Screen



© 2018 NTT DATA Taiwan Co., Ltd.

NTT DATA

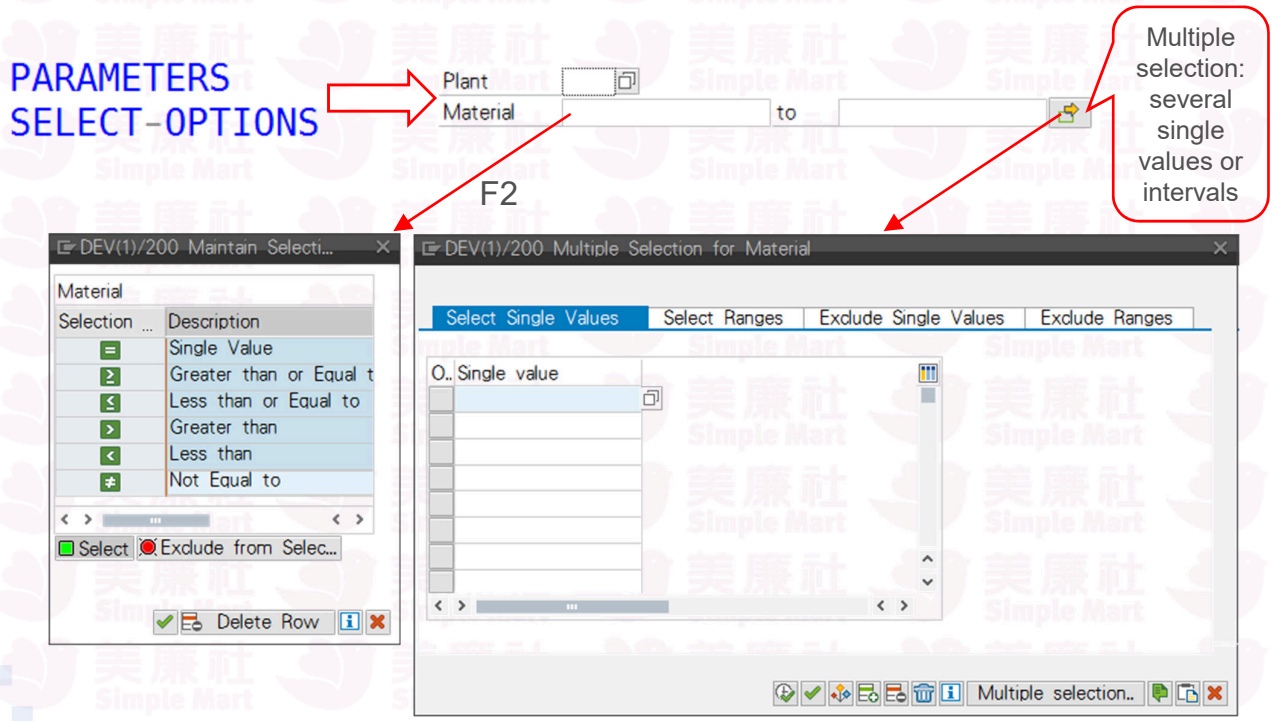
Selection Screen Attributes



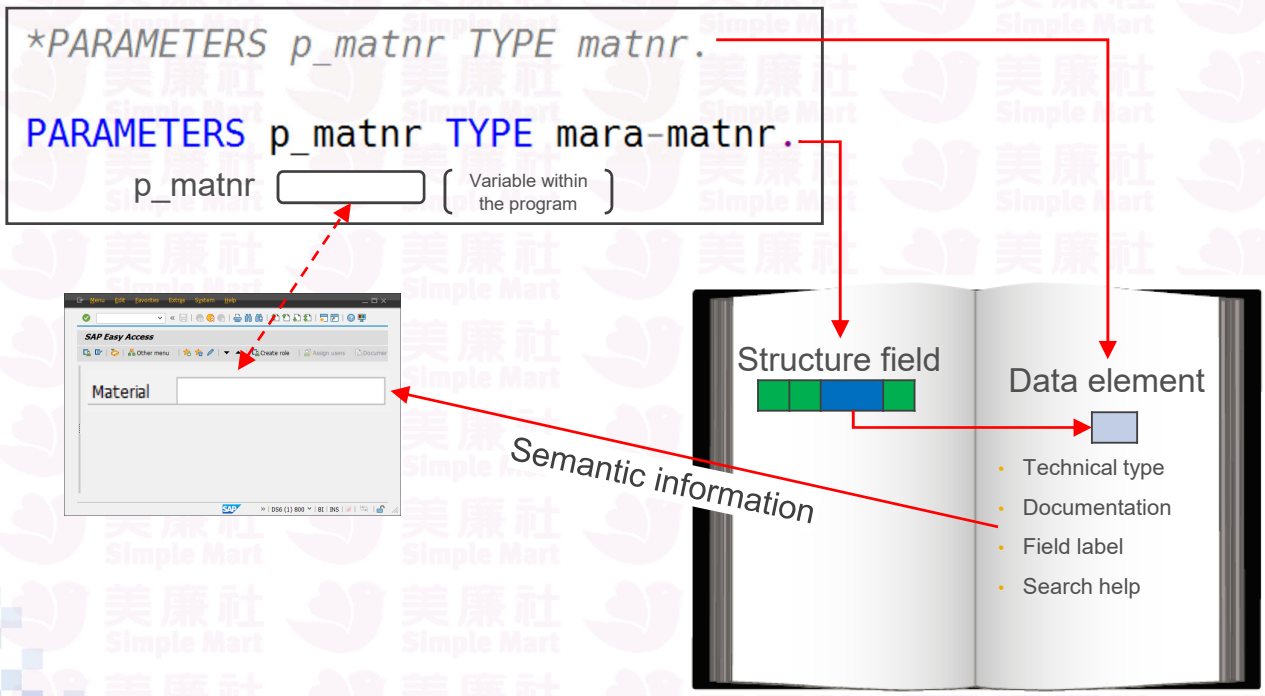
© 2018 NTT DATA Taiwan Co., Ltd.

NTT DATA

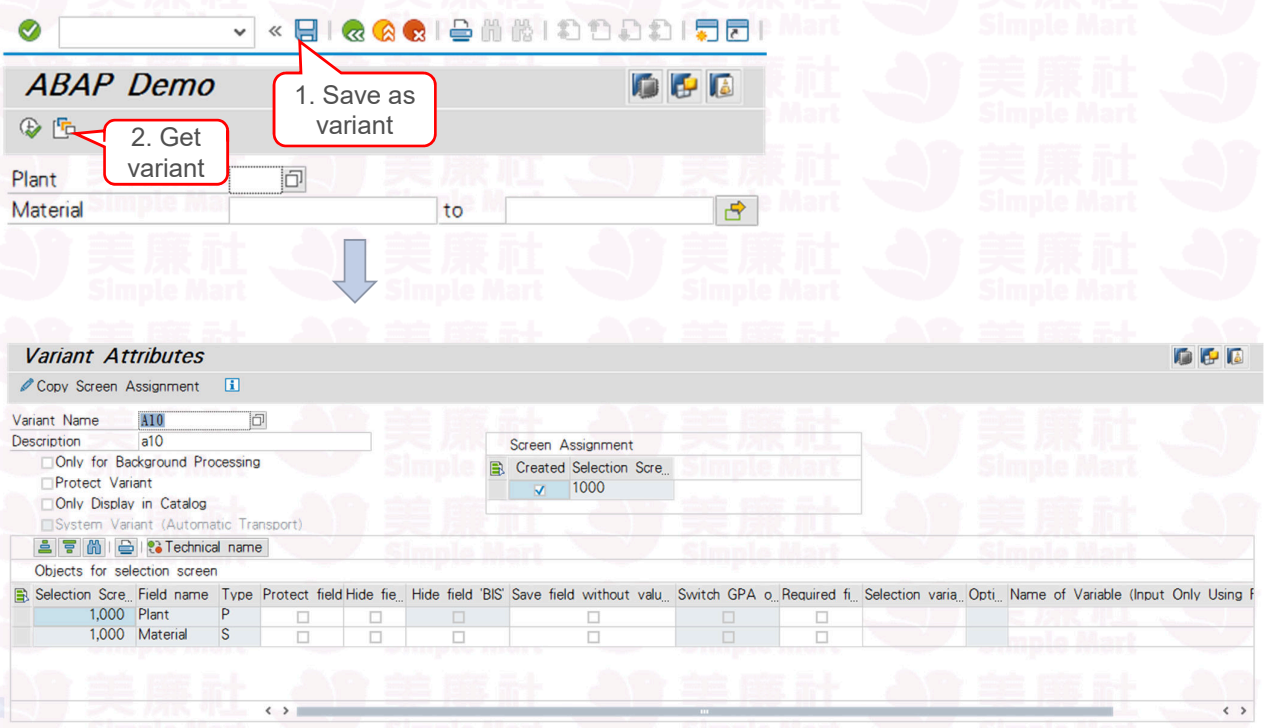
Using Selection Options



Semantic Information on Global Types on the Selection Screen

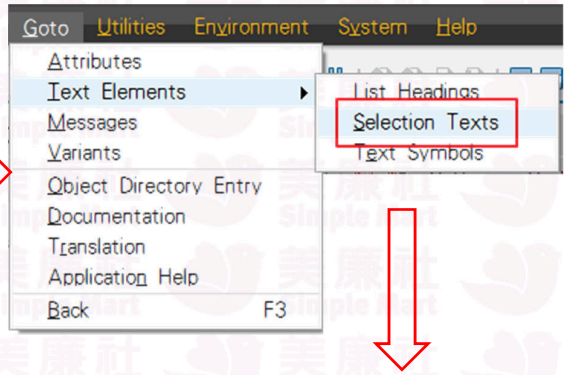


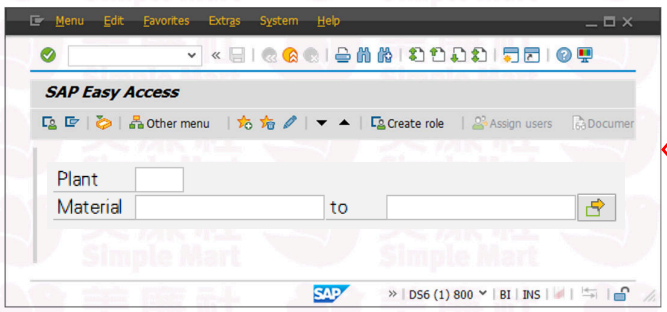
Variants

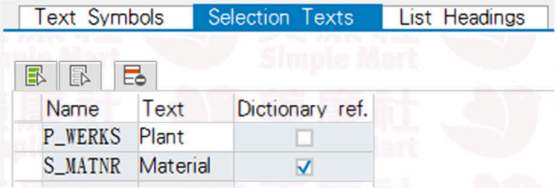


Selection Texts

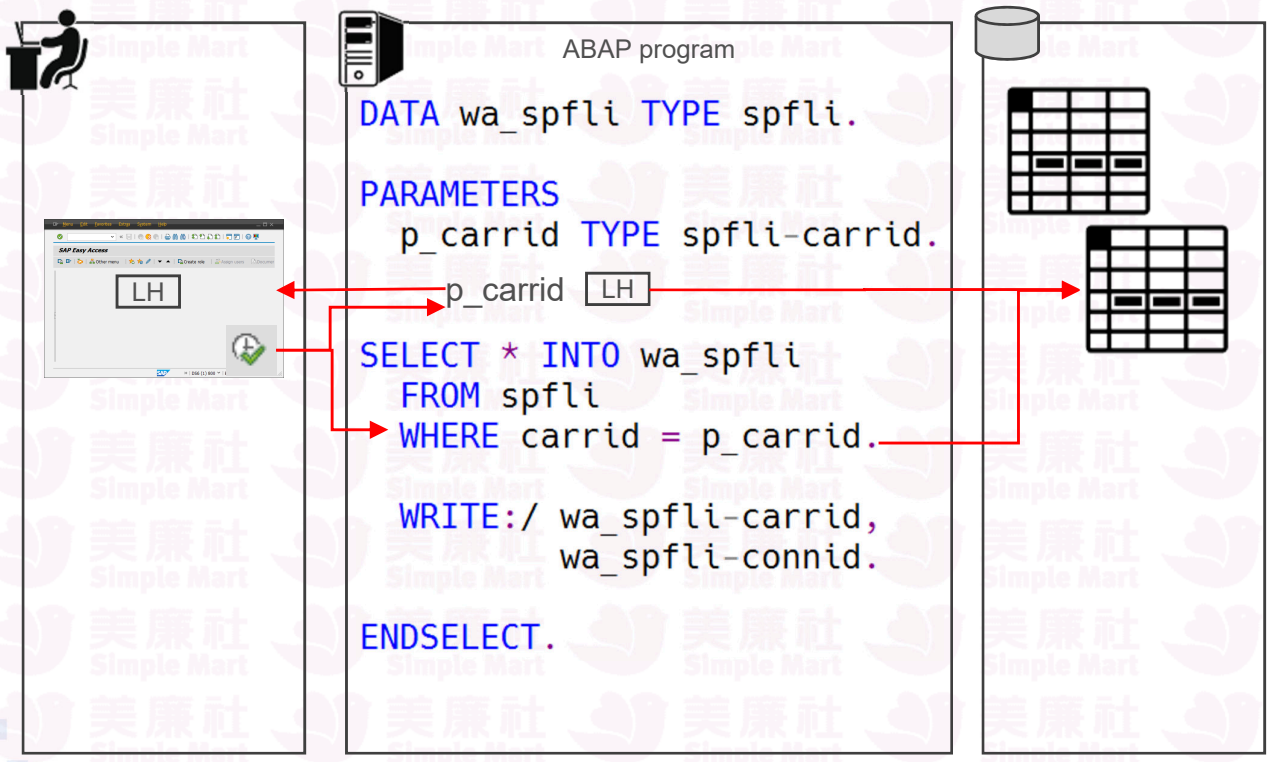
```
TABLES mara.

PARAMETERS
  p_werks TYPE t001w-werks.

SELECT-OPTIONS
  s_matnr FOR mara-matnr.
```





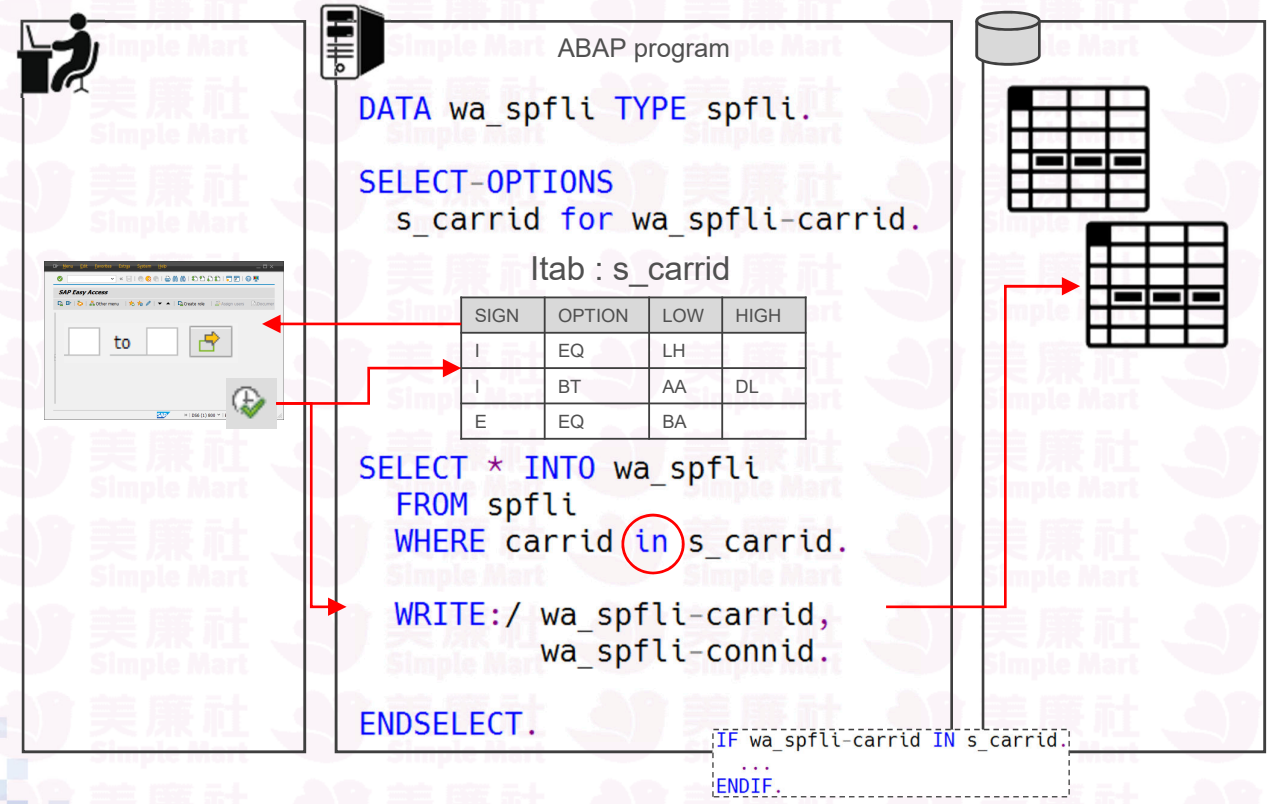


Single-Value Input (PARAMETERS Statement)



© 2018 NTT DATA Taiwan Co., Ltd.

Entering Complex Restrictions (the SELECT-OPTIONS Statement)



© 2018 NTT DATA Taiwan Co., Ltd.