

MDM ABAP API: Searching A Non Qualifier Field From A Flat Lookup Inside The Main Table



Applies to:

SAP SRM 5.5 Support Package SAPKIBKT12 with MDM_TECH 555: Add-On Installation. For more information, visit the [Master Data Management homepage](#).

Summary

The purpose of this article is to help developers for searching MDM Field inside a qualified lookup table which is further inside a main table.

Author: Animesh Sundriyal

Company: ITC Infotech Pvt. Ltd.

Created on: 19 November 2010

Author Bio



Animesh Sundriyal is an ABAP/ABAP WEBDYNPRO Consultant from past 3 years and 10 months currently working in the MDM API integration with ABAP Webdynpro.

Table of Contents

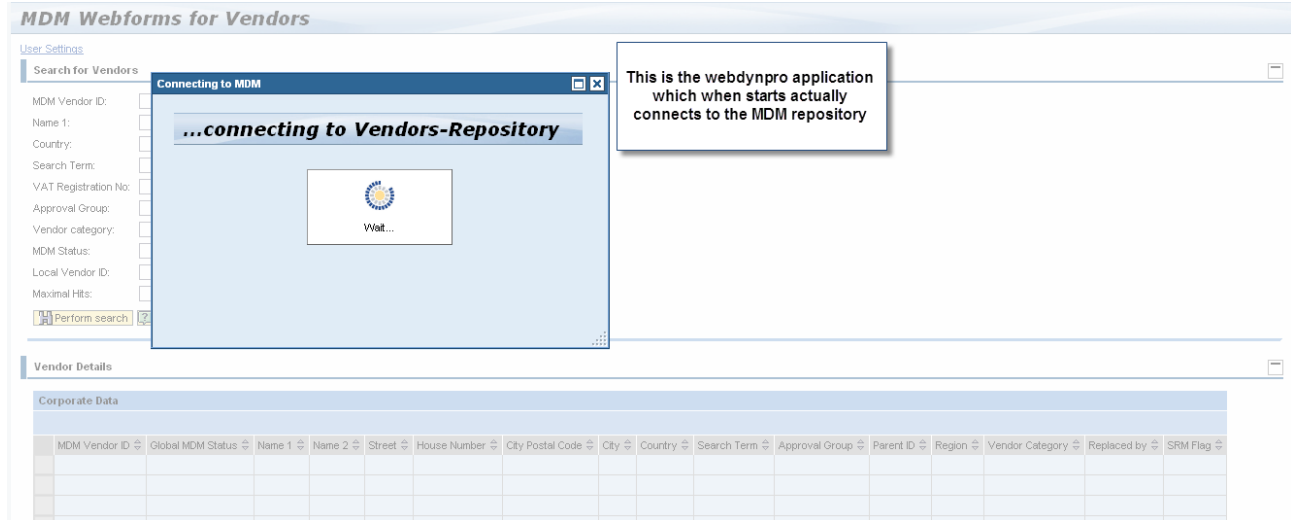
What to Do	3
Business Scenario	3
Technical Scenario (MDM Repository)	5
ABAP Coding	7
Related Content	14
Disclaimer and Liability Notice	15

What to Do

I want to show you with the help of a screenshot firstly that what is the business scenario exactly and then I will show the SE38 code for how to achieve it, in between I will show the MDM Repository to explain what exactly to search.

Business Scenario

When we start the webdynpro application it gets connected to the MDM repository and finally looks like the second screenshot.



Now lets see the scenario that what we have to search and how it looks like, in the first screen I will show you how data comes into table normally and in the next screen I will show you the explicit VAT Registration Number search.

MDM Webforms for Vendors

User Settings

Search for Vendors

MDM Vendor ID:

Name 1:

Country:

Search Term:

VAT Registration No:

Approval Group:

Vendor category:

MDM Status:

Local Vendor ID:

Maximal Hits:

here one can see that all the corporate data which has test as pattern in it has come, however this is not what i want to show, this is just to differentiate with the VAT Registration Number search

Vendor Details

Corporate Data

MDM Vendor ID	Global MDM Status	Name 1	Name 2	Street	House Number	City Postal Code	City	Country	Search Term	Approval Group	Parent ID	Region	Vendor Category	Re
10047702	Discontinuation	A/S Albert V. Jensen (TEST åu)		Vejlevej 1		87000	Horsens	DE	ALBERT V.	TEMP	No Hierarchy	02		
10032958	Active	A/S Albert V. Jensen (TEST åu)		Vejlevej 1	8	87000	Horsens	DK	ALBERT V.	TEMP	No Hierarchy			
10032871	Active	BB Mailtest22		Street		55552	City	DE		TEMP	No Hierarchy	11	004200600000000-Other	
10032874	Active	BB Test		asdfsdf		55545	KH	DE		TEMP	No Hierarchy			

Row 5 of 25

System Specific Data

VAT Registration number 1257

Local System	Local Vendor ID	MDM Status	Accounting Group	Creator	Changed by	Last change	New Vendor	Language	VAT Registration No	Block
FR	BB1234	Approved	0001	BECKERB	PANDEYS	03.11.2010 18:19:14	12345			
FR		In Work		KLEINT01	KLEINT01	26.02.2009 00:00:00				
ES	12345	Approved	ACRNACSER	PASQUALA	PANDEYS	03.11.2010 18:19:14		ES	1257	

Merge Details

The next screenshot is important as I have referred it in next sections.

MDM Webforms for Vendors

User Settings

Search for Vendors

MDM Vendor ID:

Name 1:

Country:

Search Term:

VAT Registration No: ← explicit VAT registration number search with 1257

Approval Group:

Vendor category:

MDM Status:

Local Vendor ID:

Maximal Hits:

Vendor Details

Corporate Data

MDM Vendor ID	Global MDM Status	Name 1	Name 2	Street	House Number	City Postal Code	City	Country	Search Term	Approval Group	Parent ID	Region	Vendor Category	Replaced by	SRM Flag
10032871	Active	BB Mailtest22		Street		55552	City	DE		TEMP	No Hierarchy	11	004200600000000-Other	0	X

Row 1 of 1

Result found, only one Corporate data which has 1257 VAT Registration Number as system specific data

System Specific Data

Local System	Local Vendor ID	MDM Status	Accounting Group	Creator	Changed by	Last change	New Vendor	Language	VAT Registration No	Block
FR	BB1234	Approved	0001	BECKERB	PANDEYS	03.11.2010 18:19:14	12345	FR		
FR		In Work		KLEINT01	KLEINT01	26.02.2009 00:00:00				
ES	12345	Approved	ACRNACSER	PASQUALA	PANDEYS	03.11.2010 18:19:14		ES	1257	

Merge Details

Technical Scenario (MDM Repository)

Now lets dwell into the MDM Respository, in the screenshot below one can see that we define a logical name to the actual MDM repository name.

Change View "MDM Repositories": Overview

MDMAPIC

Change View "MDM Repositories": Overview

New Entries

Dialog Structure

- MDM Repositories
- MDM Server Connections
- Mdm Dbms Hosts
- ABAP Implementation for
- MDM API Trace Configur:

Logical object name	MDM Repository name	MDM Connection
CSM_CUSTOMERS	CSM_CUST_01	SRM_MDM_SERVER
CSM_ITEM	CSM_ITEM_01	SRM_MDM_SERVER
CSM_VENDOR	CSM_VENDOR_01	SRM_MDM_SERVER

In transaction MDMAPIC we assign the logical name to the actual MDM repository

The below screenshot is of MDM Console which is showing the MDM Repository where our search field is present.

SAP MDM Console

This is MDM Console

Console Hierarchy

- SAP MDM Servers
 - 194.99.131.41
 - CSM_CUST_01
 - CSM_CUST_TEST
 - CSM_ITEM_01
 - CSM_VENDOR_01
 - Vendors
 - Approval Groups
 - Countries
 - Global Status Network
 - Local Account Groups
 - Local Block Functions
 - Local Languages
 - Local Systems
 - MDM Status
 - MDM Status Network
 - Parent ID
 - Regions
 - Responsibilities
 - SRM relevant Flag
 - Vendor roles
 - Vendor categories
 - System Specific Data
 - Images
 - Image Variants
 - Sounds
 - Videos
 - Binary Objects
 - Text Blocks
 - Copy Blocks
 - Text HTMLs
 - PDFs
 - Masks
 - Relationships

In this repository our search field is present

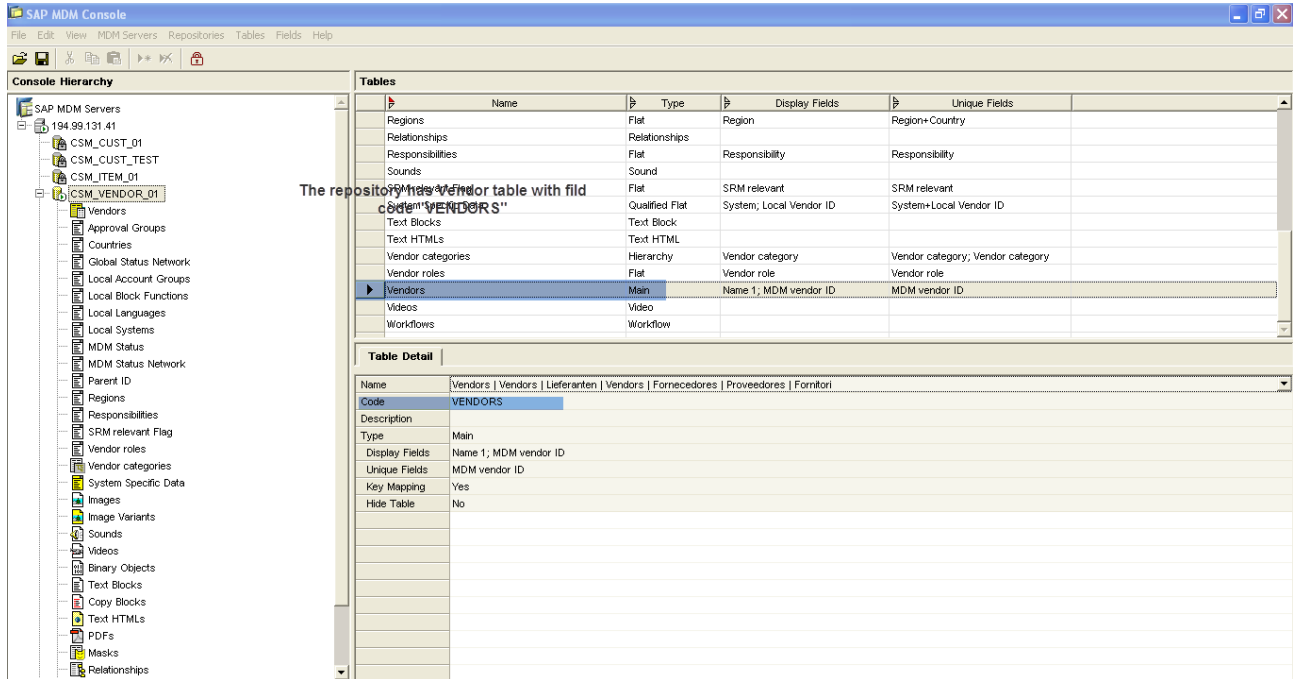
This is the actual name of the repository and we gave it a logical name i showed in previous screenshot

Name	Type	Display Fields	Unique Fields
Approval Groups	Flat	Approval Group	Approval Group
Binary Objects	Binary Object		
Copy Blocks	Copy Block		
Countries	Flat	Code	Code
Global Status Network	Flat	Global Status	Global Status
Image Variants	Image Variants		
Images	Image		
Local Account Groups	Flat	Accounting Group	Local System+Accounting Group
Local Block Functions	Flat	Block Function	System+Block Function
Local Languages	Flat	ISO Code	System+ISO Code
Local Systems	Flat	Local System	Local System
Masks	Masks	Name	
MDM Status	Flat	MDM Status	MDM Status

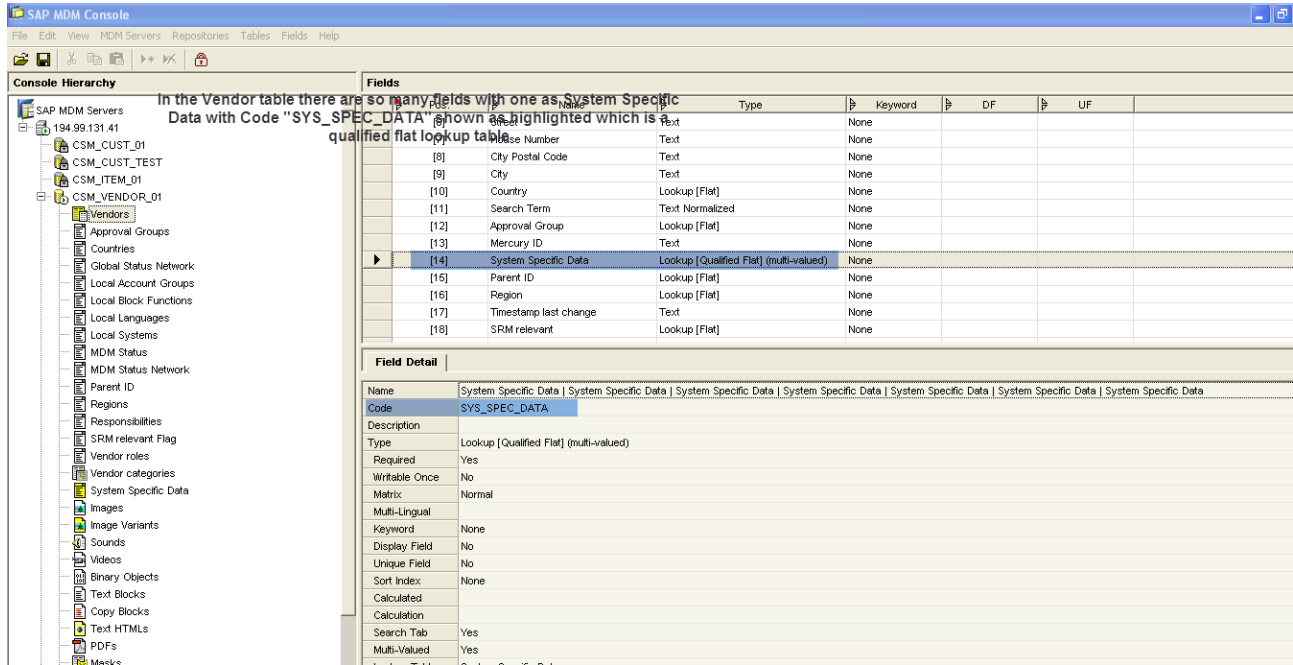
Table Detail

Name	Approval Groups Approval Groups Approval Groups Approval Groups Approval Groups Approval Groups Approval Groups
Code	APPROVAL_GROUP
Description	Group for approval of this vendor
Type	Flat
Display Fields	Approval Group
Unique Fields	Approval Group
Key Mapping	Yes
Hide Table	No

The below repository has a Vendor table with the Code " VENDORS".



In the below screenshot the fields inside the Vendor table is shown with System Specific Data as one field with field code "SYS_SPEC_DATA" which is a lookup table..



Now lets see the fields inside the lookup table System Specific Data, and we will find that VAT Registration Number on which basis we have to perform our search is found here.

The screenshot shows the SAP MDM Console interface. On the left is the 'Console Hierarchy' tree, where 'System Specific Data' is highlighted. A red arrow points to it with the text 'We are looking inside System Specific Data Table'. The main area displays a 'Fields' table with columns: Pos., Name, Type, Keyword, DF, UF, and Qualifier. The table lists various fields, with 'VAT Registration Number' (Pos. 13) selected. A red arrow points to this row with the text 'Field on whose basis we have to perform the search'. Below the table is the 'Field Detail' section for 'VAT Registration Number', showing its code as 'VAT_REGISTRATION_NUMBER'. A red arrow points to this code with the text 'Field Code is VAT_REGISTRATION_NUMBER'.

Pos.	Name	Type	Keyword	DF	UF	Qualifier
[1]	System	Lookup [Flat]	None	[1]	[1]	No
[2]	Local Vendor ID	Text	None			No
[3]	Local Account Group	Lookup [Flat]	None	[2]	[1]	Yes
[4]	Created	Create Stamp	None			Yes
[5]	Created By	Text	None			Yes
[6]	Changed	Time Stamp	None			Yes
[7]	Timestamp	Text	None			Yes
[8]	Changed By	Text	None			Yes
[9]	New vendor	Text	None			Yes
[10]	MDM Status	Lookup [Flat]	None			Yes
[11]	Language	Lookup [Flat]	None			Yes
[12]	Notepad	Text	None			Yes
[13]	VAT Registration Number	Text	None			Yes

Field Detail	
Name	VAT Registration Number
Code	VAT_REGISTRATION_NUMBER
Description	
Type	Text
Required	No
Writable Once	No
Matrix	Normal
Multi-Lingual	No
Key-Word	None
Display Field	No
Unique Field	No
Sort Index	Normal
Calculated	No
Calculation	
Qualifier	Yes
Cache	Yes
Width	20
Sort Type	Case Insensitive

So now the field VAT Registration Number on which basis we have to search is actually inside System Specific Data table which in turn is inside Vendor table.

ABAP Coding

The business scenario and the insight into MDM repository is now over, let's see how to code for searching VAT Registration Number value from here. Follow the comments to understand the code appropriately.

```

data : mr_mdm_api type ref to CL_MDM_GENERIC_API,
      ms_cdt_language_code type MDM_CDT_LANGUAGE_CODE.

TYPE-
POOLS: mdmif." An important type pool to get the already defined MDM related variables

*Instantiating the MDM Repository
CREATE OBJECT mr_mdm_api
EXPORTING
  iv_log_object_name = 'CSM_VENDOR'"Here in program we have to define the Logical
Name i showed in a screenshot in Tcode MDMAPIC

.

  ms_cdt_language_code-language = 'eng'.
  ms_cdt_language_code-country = 'US'.
  ms_cdt_language_code-region = '___'.

*Connecting to the MDM Repository
mr_mdm_api->mo_accessor->connect( ms_cdt_language_code )."engUS ___

* This section of code actually retrieve the Vendor table record or the Corporate data
over the screenshot i showed earlier
* by first getting the MDM Qualified links and later the actual records

```

```

DATA lr_data_ref          TYPE REF TO data.
DATA lv_statusx          TYPE string.
DATA ls_qualifier_query  TYPE mdm_qualifier_query.
DATA lt_query            TYPE mdm_query_table.
DATA ls_query            TYPE mdm_query.
DATA lt_query_local      TYPE mdm_query_table.
DATA ls_result_set_definition TYPE mdm_search_result_definition.
DATA lt_result_set       TYPE mdm_search_result_table.
DATA ls_result_set       LIKE LINE OF lt_result_set.
FIELD-SYMBOLS <fs_data> TYPE ANY.

```

lv_statusx = '1257'. "we are passing the VAT Registration Number as search string here

```

CREATE DATA lr_data_ref type string.
ASSIGN lr_data_ref->* TO <fs_data>.
<fs_data> = lv_statusx.
ls_qualifier_query-value = lr_data_ref.

```

```
ls_qualifier_query-qualifier_field_code = 'VAT_REGISTRATION_NUMBER'.
```

ls_query-
parameter_code = 'SYS_SPEC_DATA'. "Field code for system specific data table

```
ls_query-operator = 'CS'. "EQUALS
```

```
ls_query-dimension_type = MDMIF_SEARCH_DIM_QUAL. "Field search dimension
```

```
ls_query-constraint_type = mdmif_search_constr_text. "text search constraint
```

```

CREATE DATA lr_data_ref TYPE mdm_qualifier_query.
ASSIGN lr_data_ref->* TO <fs_data>.
<fs_data> = ls_qualifier_query.

```

```

ls_query-value_low = lr_data_ref.
APPEND ls_query TO lt_query.

```

* This query will fetch the MDM Qualified Links for the actual search for table Vendors

* The beauty of this query is it is searching in Vendor table and in the parameter lt_query we

* have System Specific Data with the VAT Registration Number with the desired value 1257

```

CALL METHOD mr_mdm_api->mo_core_service->query
  EXPORTING
    iv_object_type_code      = 'VENDORS'" Field Code for Vendor table
    is_result_set_definition = ls_result_set_definition
    it_query                 = lt_query "The above query which we just created
  IMPORTING
    et_result_set           = lt_result_set.

```



```

DATA lt_result TYPE table of ZST_MDM_VENDOR."A structure which contains all the Field Code of the fields of the actual MDM repository Vendor table
DATA ls_result TYPE zst_mdm_vendor.

```

```

LOOP AT lt_result_set INTO ls_result_set.

```

* From the MDM Qualified Links which we got in the previous query this method will fetch the actual Vendor data records

* Retrieve corporate data

```

CALL METHOD mr_mdm_api->mo_core_service->retrieve_simple
  EXPORTING
    iv_object_type_code = 'VENDORS'
    it_keys              = ls_result_set-record_ids
  IMPORTING
    et_ddic_structure   = lt_result.

```

```

ENDLOOP.

```

* This section of the code will fetch the System Specific Data Record from the qualified links found in the

* SYS_SPEC_DATA field of the Vendor table to get the System Specific Data shown in the second Table of the screenshot i showed earlier

```

DATA ls_result_sys_ids TYPE mdm_qualified_link.
DATA ls_qualifier      TYPE LINE OF mdm_name_value_pair_table.
* A structure which contains all the Field Code of the fields of the actual MDM repository System Specific Data table
DATA ls_result_local   TYPE zst_mdm_system_spec_vendors.
DATA lt_local_lookup_result TYPE ztt_mdm_system_spec_vendors.
DATA ls_local_lookup_result TYPE zst_mdm_system_spec_vendors.
DATA lt_keys           TYPE mdm_keys.
DATA lv_tab_index      TYPE i.
DATA lc_table          TYPE string.
DATA et_vendor_data_local TYPE table of ZST_MDM_SYSTEM_SPEC_VENDORS.
DATA ls_search_result  TYPE zst_mdm_vendor_search_result.

```

```

FIELD-SYMBOLS <value_int> TYPE mdm_gdt_integervalue.
FIELD-SYMBOLS <value_time> TYPE mdm_cdt_date_time.
FIELD-SYMBOLS <value_string> TYPE string_data.

```

```

LOOP AT lt_result INTO ls_result.
  MOVE-CORRESPONDING ls_result TO ls_search_result.

```

```

refresh et_vendor_data_local.

```

```

lv_tab_index = 0.

```

```

LOOP AT ls_result-
sys_spec_data into ls_result_sys_ids." it_mdm_qualified_link_table INTO ls_result_sys
_ids.

* Retrieve Local system id from lookup table.
APPEND ls_result_sys_ids-qualified_lookup_id TO lt_keys.
CALL METHOD mr_mdm_api->mo_core_service->retrieve_simple
EXPORTING
    iv_object_type_code = 'SYST_SPEC_DATA'"Field Code Of System Specific Data Tab
le
    it_keys              = lt_keys
IMPORTING
    et_ddic_structure   = lt_local_lookup_result.
lv_tab_index = lv_tab_index + 1.
READ TABLE lt_local_lookup_result INDEX lv_tab_index INTO ls_local_lookup_result.
MOVE-CORRESPONDING ls_local_lookup_result TO ls_result_local.

* assign qualifier to structure.
LOOP AT ls_result_sys_ids-qualifier INTO ls_qualifier.
IF ls_qualifier-code EQ 'LOCAL_VENDOR_ID'.
    ASSIGN ls_qualifier-value->* TO <value_string>.
    ls_result_local-local_vendor_id = <value_string>.

ELSEIF ls_qualifier-code EQ 'FOLLOW_UP_VENDOR'.
    ASSIGN ls_qualifier-value->* TO <value_string>.
    ls_result_local-follow_up_vendor = <value_string>.

ELSEIF ls_qualifier-code EQ 'MDM_STATUS'.
    ASSIGN ls_qualifier-value->* TO <value_int>.
    ls_result_local-mdm_status = <value_int>.

ELSEIF ls_qualifier-code EQ 'LOCAL_ACC_GROUP'.
    ASSIGN ls_qualifier-value->* TO <value_int>.
    ls_result_local-local_acc_group = <value_int>.

ELSEIF ls_qualifier-code EQ 'CREATED'.
    ASSIGN ls_qualifier-value->* TO <value_time>.
    ls_result_local-created = <value_time>.

ELSEIF ls_qualifier-code EQ 'CREATOR'.
    ASSIGN ls_qualifier-value->* TO <value_string>.
    ls_result_local-creator = <value_string>.

ELSEIF ls_qualifier-code EQ 'DATE_LAST_CHANGE'.
    ASSIGN ls_qualifier-value->* TO <value_time>.
    ls_result_local-date_last_change = <value_time>.

ELSEIF ls_qualifier-code EQ 'CHANGED_BY'.
    ASSIGN ls_qualifier-value->* TO <value_string>.
    ls_result_local-changed_by = <value_string>.

ELSEIF ls_qualifier-code EQ 'TIMESTAMP_SY'.
    ASSIGN ls_qualifier-value->* TO <value_string>.

```

```

1s_result_local-TIMESTAMP_SY = <value_string>.

ELSEIF 1s_qualifier-code EQ 'LANGUAGE'.
  ASSIGN 1s_qualifier-value->* TO <value_int>.
  1s_result_local-language = <value_int>.

ELSEIF 1s_qualifier-code EQ 'NOTEPAD'.
  ASSIGN 1s_qualifier-value->* TO <value_string>.
  1s_result_local-notepad = <value_string>.

ELSEIF 1s_qualifier-code EQ 'VAT_REGISTRATION_NUMBER'.
  ASSIGN 1s_qualifier-value->* TO <value_string>.
  1s_result_local-VAT_REGISTRATION_NUMBER = <value_string>.

ELSEIF 1s_qualifier-code EQ 'BLOCK_FUNCTION'.
  ASSIGN 1s_qualifier-value->* TO <value_int>.
  1s_result_local-block_function = <value_int>.

ENDIF.
ENDLOOP. "local qualifier
APPEND 1s_result_local TO et_vendor_data_local." This table will contain all the
System Specific Data record corresponding to the Vendor Table Record
ENDLOOP." local system ids

ENDLOOP.

* Disconnecting from the repository

mr_mdm_api->mo_accessor->disconnect( ).

```

Now lets come to the result which I will show you with the help of Debugger and can be tallied from the fourth screenshot of the document. Firstly I will show you the value of Vendor table records.

(2) - ABAP Debugger Controls Session 2 (Exclusive)

(2) - ABAP Debugger Controls Session 2 (Exclusive)

Desktop 1 Desktop 2 Desktop 3 Standard Structures Tables Objects Detail Disps. Data Explorer Break/Watchpoints Diff

Variables 1 Variables 2 Locals Globals

St Variable Va Val. C... Hexadecimal V

St	Variable	Va	Val.	C...	Hexadecimal V
	LT_RESULT		Standard Table[1x19(128)]		

```

84
85
86 DATA lt_result TYPE table of ZST_MDM_VENDOR."A structure which c
87 DATA ls_result TYPE zst_mdm_vendor.
88
89 LOOP AT lt_result_set INTO ls_result_set.
90
91
92 * From the MDM Qualified Links which we got in the previous query t
93 * Retrieve corporate data
94 CALL METHOD mr_mdm_api->mo_core_service->retrieve_simple
95 EXPORTING
96   iv_object_type_code = 'VENDORS'
97   it_keys              = ls_result_set-record_ids
98 IMPORTING
99   et_ddic_structure   = lt_result.
100
101
102 ENDLOOP.
103
104 Actual Vendor Table Record
105
106 * This section of the code will fetch the System Specific Data Reco
107 * SYS_SPEC_DATA field of the Vendor table to get the System Specifi

```

Debugger Edit Goto Breakpoints Settings Miscellaneous System Help

(2) - ABAP Debugger Controls Session 2 (Exclusive)

ZASTEST / ZASTEST / 102 SY-SUBRC 0
 EVENT / START-OF-SELECTION SY-TABIX 8

Desktop 1 Desktop 2 Desktop3 Standard Structures Tables Objects Detail Displ. Data Explorer Break/Watchpoints Diff

Tables Table Contents

Table LT_RESULT
 Table Type Standard Table[1x19(128)]

Line	MDM_ID[(4)]	GLOBAL_MDM_STA	NAME_1[CString]	NAME_2[CString]	STREET[CString]	HOUSE_NUMBE	CITY_POSTAL_C	CITY[CString]	COUNTRY[(4)]	SEARCH_TERM	APPROVAL_GRO_
1	10032871	1	BB Mailtest22		Street		55552	City	80		7

Tally it with the fourth Screenshot Corporate Data

We will see the System Specific Data table record with the desired VAT Registration Value we are searching for.

Debugger Edit Goto Breakpoints Settings Miscellaneous System Help

(2) - ABAP Debugger Controls Session 2 (Exclusive)

ZASTEST / ZASTEST / 222 SY-SUBRC 0
 EVENT / START-OF-SELECTION SY-TABIX 1

Desktop 1 Desktop 2 Desktop3 Standard Structures Tables Objects Detail Displ. Data Explorer Break/Watchpoints Diff

```

193:   ls_result_local-language = <value_int>.
194:
195:   ELSEIF ls_qualifier-code EQ 'NOTEPAD'.
196:     ASSIGN ls_qualifier-value->* TO <value_string>.
197:     ls_result_local-notepad = <value_string>.
198:
199:   ELSEIF ls_qualifier-code EQ 'VAT_REGISTRATION_NUMBER'.
200:     ASSIGN ls_qualifier-value->* TO <value_string>.
201:     ls_result_local-VAT_REGISTRATION_NUMBER = <value_string>.
202:
203:   ELSEIF ls_qualifier-code EQ 'BLOCK_FUNCTION'.
204:     ASSIGN ls_qualifier-value->* TO <value_int>.
205:     ls_result_local-block_function = <value_int>.
206:
207:   ENDIF.
208:   ENDLOOP. "local qualifier
209:   APPEND ls_result_local TO et_vendor_data_local." This table wil
210:   ENDLOOP." local system id
211:
212:   System Specific Data gets populated
213:   here
214:
215:   ENDLOOP.
216:
217:
    
```

System Specific Data gets populated here

St.	Variable	Va	Val.	C.	Hexadecimal
	LT_RESULT		Standard Table[1x19(128)]		
	ET_VENDOR_DATA_LOCAL		Standard Table[3x14(100)]		

(2) - ABAP Debugger Controls Session 2 (Exclusive)

ZASTEST / ZASTEST / 222 SY-SUBRC 0
 EVENT / START-OF-SELECTION SY-TABIX 1

Desktop 1 Desktop 2 Desktop3 Standard Structures Tables Objects Detail Displ. Data Explorer Break/Watchpoints Diff

Tables Table Contents

Table ET_VENDOR_DATA_LOCAL
 Table Type Standard Table[3x14(100)]

Line	LOC_SYSTEM[(4)]	LOCAL_VENDOR	LOCAL_ACC_GR	CREATED[FlatStr]	CREATOR[CStr]	DATE_LAST_CHA	CHANGED_BY[C]	FOLLOW_UP_VE	MDM_STATUS[(4)]	TIMESTAMP_SY[LANGUAGE[(4)]	NO
1	3	BB1234	1	Structure: flat	BECKERB	Structure: flat	PANDEYS	12345	3	20101103181914	1	
2	3		0	Structure: flat	KLEINT01	Structure: flat	KLEINT01		1	20090226000000	0	
3	5	12345	9	Structure: flat	PASQUALA	Structure: flat	PANDEYS		3	20101103181914	3	

System Specific Data Value can be tallied from Screenshot 4

(2) - ABAP Debugger Controls Session 2 (Exclusive)

ZASTEST / ZASTEST / 222 SY-SUBRC 0
 EVENT / START-OF-SELECTION SY-TABIX 1

Desktop 1 Desktop 2 Desktop3 Standard Structures Tables Objects Detail Displ. Data Explorer Break/Watchpoints Diff

Tables Table Contents

Table ET_VENDOR_DATA_LOCAL
 Table Type Standard Table[3x14(100)]

Line	CREATOR[CStr]	DATE_LAST_CHA	CHANGED_BY[C]	FOLLOW_UP_VE	MDM_STATUS[(4)]	TIMESTAMP_SY[LANGUAGE[(4)]	NOTEPAD[CString]	VAT_REGISTRATI	BLOCK_FUNCTI
1	BECKERB	Structure: flat	PANDEYS	12345	3	20101103181914	1			0
2	KLEINT01	Structure: flat	KLEINT01		1	20090226000000	0			0
3	PASQUALA	Structure: flat	PANDEYS		3	20101103181914	3		1257	0

On scrolling the table to right

Here we found the desired output

Related Content

http://help.sap.com/saphelp_mdm550/helpdata/en/44/93aa6831381053e10000000a422035/frameset.htm

http://help.sap.com/saphelp_mdm71/helpdata/en/44/9ab53b675a70dbe10000000a1553f6/content.htm

<http://forums.sdn.sap.com/message.jspa?messageID=8719178>

For more information, visit the [Master Data Management homepage](#).

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.