

Alternate way of **READ TABLE**

- Alternate syntax you can use instead of READ TABLE
- New syntax of doing same
- Reduce number of lines
- Facing challenges while debugging with new syntax ?



Swapnil More
SAP ABAP Developer @SVKM
| S/4 Hana |OO-ABAP | Js |

```

184 *~~Get line from table to Work Area.
185
186 *--Old syntax
187
188 READ TABLE lt_knal INTO DATA(wa_knal) With KEY kunnr = lv_kunnr.
189
190 *-- New Syntax
191
192 DATA(wa2_knal) = VALUE #( lt_knal[ kunnr = lv_kunnr ] optional ).
193

```

The screenshot shows the SAP ABAP Development Workbench interface. The code editor on the left displays ABAP code, with line 188 highlighted in red. The variables viewer on the right shows the state of variables in the work area.

Sta...	Variable	Val...	Val.	Ch...	Hexac
	WA_KNA1-NAME1		Mahindra Insurance Broker		4D00
	WA_KNA1-LAND1		IN		4900
	WA2_KNA1-NAME1		Mahindra Insurance Broker		4D00
	WA2_KNA1-LAND1		IN		4900



Swapnil More
SAP ABAP Developer @SVKM
| S/4 Hana |OO-ABAP | Js |

```
194
195  ┌-- Read table with Index 1.
196  | └-- old way
197  |      READ TABLE lt_knal INTO DATA(wa_index1) INDEX 1.
198
199  |      -- New Syntax / Alternate way
200  |      DATA(WA2_INDEX1) = VALUE #( lt_knal[ 1 ] OPTIONAL ).
```

The screenshot shows the SAP SE Editor interface. On the left is the code editor with lines 191 to 202. Lines 195 and 196 are comments. Line 197 contains the old syntax: `READ TABLE lt_knal INTO DATA(wa_index1) INDEX 1.`. Line 199 is a comment: `--- New Syntax / Alternate way`. Line 200 contains the new syntax: `DATA(WA2_INDEX1) = VALUE #(lt_knal[1] OPTIONAL).`. Line 202 is a write statement: `WRITE: 'Debugger'.`. The right side of the interface shows the variable viewer with tabs for Variables 1, Variables 2, Locals, Globals, and Auto. The Variables 1 tab is selected, showing two entries: `WA_INDEX1-LAND1` and `WA2_INDEX1-LAND1`, both with values `IN`.

```
191  
192 DATA(wa2_knal) = VALUE #( lt_knal[ kunnr = lv_kunnr ] optional ).  
193  
194  
195 -- Read table with Index 1.  
196 |-- Old way  
197 READ TABLE lt_knal INTO DATA(wa_index1) INDEX 1.  
198  
199 --- New Syntax / Alternate way  
200 DATA(WA2_INDEX1) = VALUE #( lt_knal[ 1 ] OPTIONAL ).  
201  
202 WRITE: 'Debugger'.  
Variables 1 Variables 2 Locals Globals Auto  
Sta... Variable Val... Val.  
WA_INDEX1-LAND1 IN  
WA2_INDEX1-LAND1 IN
```



Swapnil More
SAP ABAP Developer @SVKM
| S/4 Hana | OO-ABAP | Js |

```

204  |-- Get a particular field from one table to another variable.
205  |  |--Old way
206
207      READ TABLE lt_knal INTO DATA(WA_GETFIELD) WITH KEY kunnr = lv_kunnr.
208  IF SY-subrc = 0.
209      lv_name = wa_getfield-name1.
210  ENDIF.
211
212  |-- New way / Alternate way
213
214  lv_name1 = VALUE #( lt_knal[ kunnr = lv_kunnr ]-name1 OPTIONAL ) .
215

```

The screenshot shows the SAP ABAP development environment. On the left is the code editor with the following content:

```

204  |-- Get a particular field from one table to another variable.
205  |  |--Old way
206
207      READ TABLE lt_knal INTO DATA(WA_GETFIELD) WITH KEY kunnr = lv_kunnr.
208  IF SY-subrc = 0.
209      lv_name = wa_getfield-name1.
210  ENDIF.
211
212  |-- New way / Alternate way
213
214  lv_name1 = VALUE #( lt_knal[ kunnr = lv_kunnr ]-name1 OPTIONAL ) .
215
216  WRITE: 'Debugger'.

```

On the right is the variable viewer window titled "Variables 1". It displays the following data:

Sta...	Variable	Val...	Val.	Ch...	Hexade
	LV_NAME	Mahindra Insurance Broker			4D00E
	LV_NAME1	Mahindra Insurance Broker			4D00E



Swapnil More
SAP ABAP Developer @SVKM
| S/4 Hana |OO-ABAP | Js |

```
215  
216  □ *--Check particular Value is in Internal Table.  
217  | *--Old way  
218  
219    READ TABLE lt_knal TRANSPORTING NO FIELDS WITH KEY kunnr = lv_kunnr.  
220  □ IF SY-SUBRC = 0.  
221  |  
222  | ENDIF.  
223  
224  *-- New way / Alternate way  
225  
226  □ IF LINE_EXISTS( lt_knal[ kunnr = lv_kunnr ] ) .  
227  |  
228  | ENDIF.  
229
```



Swapnil More
SAP ABAP Developer @SVKM
| S/4 Hana |OO-ABAP | Js |

```

229
230  ┌--Get Index Number of Particular Entry from Internal Table.
231  ┌--Old way
232
233    READ TABLE lt_knal TRANSPORTING NO FIELDS WITH KEY kunnr = lv_kunnr.
234  ┌ IF sy-subrc = 0.
235    DATA(lv_tabx) = sy-tabix.
236  ENDIF.
237
238  *-- New way / Alternate way
239
240  DATA(LV_TABX2) = LINE_INDEX( lt_knal[ kunnr = lv_kunnr ] ).
241

```

The screenshot shows the SAP ABAP IDE interface. On the left, the code editor displays the ABAP code provided above. On the right, the debugger interface is open, specifically the 'Variables' tab. The variables pane shows two entries:

Sta...	Variable	Val...	Val.	Ch...	Hexadecimal Value
	LV_TABX		9		09000000
	LV_TABX2		9		09000000

The variable 'LV_TABX' has a value of 9, and 'LV_TABX2' also has a value of 9. The ABAP code uses both the 'READ TABLE' statement with a key and the 'LINE_INDEX' function to achieve the same result.



Swapnil More
SAP ABAP Developer @SVKM
| S/4 Hana | OO-ABAP | Js |

```
241
242     --> Modify Entry in Table.
243         -->Old way
244
245     READ TABLE lt_knal ASSIGNING FIELD-SYMBOL(<FS_KNA1>) WITH KEY kunnr = lv_kunnr.
246     IF sy-subrc = 0.
247         <fs_knal>-name1 = 'SWAPNIL MORE'. "or you can use variable or text element
248     ENDIF.
249
250     --> New way / Alternate way
251
252     lt_knal[ kunnr = lv_kunnr ]-name1 = 'Swapnil More'.
253
254
```



Swapnil More
SAP ABAP Developer @SVKM
| S/4 Hana |OO-ABAP | Js |

Note: While Using these Syntaxes in case of
Table Expression use with Try <> Catch
with *cx_sy_itab_line_not_found* EndTry
block.

GitHub link : Access direct code

<https://github.com/swapnilm06/SAP-ABAP-HANA/blob/main/New%20syntax%20of%20READ%20table.txt>



Swapnil More
SAP ABAP Developer @SVKM
| S/4 Hana | OO-ABAP | Js |

REPOST....SHARE....FOLLOW



Swapnil More

SAP ABAP Developer @SVKM

| S/4 Hana |OO-ABAP | Js |

[Follow](#)

Click here