

# Reading and Writing Data from Web Dynpro for ABAP View Context



## Applies to:

Web Dynpro ABAP. For more information, visit the [ABAP homepage](#).

## Summary

This tutorial explains how to read and write data to and from a views context. It deals with both tables as well as variables.

**Author:** Mazin Mahmood

**Company:** Keane India Pvt. Ltd.

**Created on:** 26 November 2009

## Author Bio

Mazin Mahmood is a Software Engineer at Keane India Private Limited and is working on Enterprise Portals, Web Dynpro for ABAP and Product and Portfolio Management.

## Table of Contents

Requirement .....	3
Assumptions .....	3
Code.....	5
Scenario 1: Code to obtain values from the context node to variable .....	5
Scenario 2: Code to write values from variable to context node.....	5
Scenario 3: Code to obtain values from the context node to local internal table.....	5
Scenario 4: Code to write values from local internal table to context node .....	6
Disclaimer and Liability Notice.....	7

## Requirement

The heart of Web Dynpro ABAP is data exchange between context and the UI.  
This can be in the form of a table or it can be in the form of a variable etc.

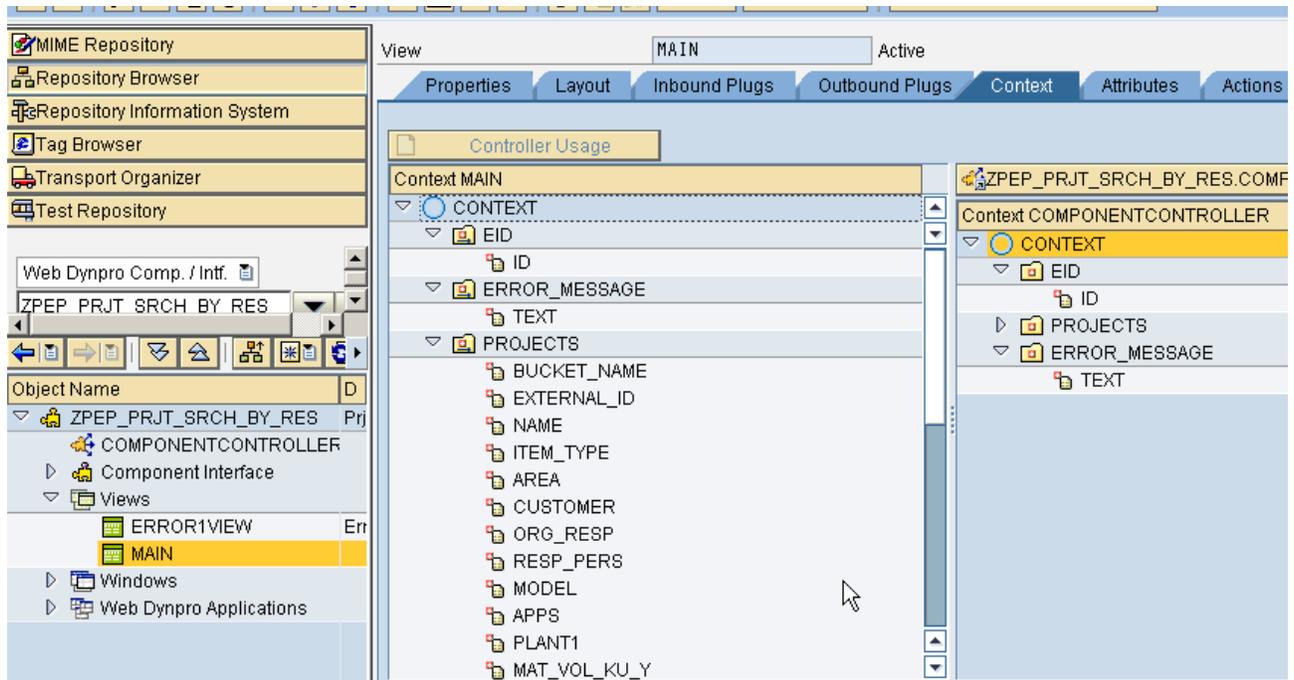
## Assumptions

Let us assume that we have a context for the Component controller as given below.

The screenshot displays the SAP Web Dynpro ABAP development environment. On the left, the 'Object Navigator' shows the project structure for 'ZPEP\_PRJT\_SRCH\_BY\_RES', with 'COMPONENTCONTROLLER' selected. The main area shows the 'Context Usage' for 'COMPONENTCONTROLLER'. The context structure is as follows:

- Context COMPONENTCONTROLLER
  - CONTEXT
    - EID
      - ID
    - PROJECTS
      - TEXT
      - EXTERNAL\_ID
    - ERROR\_MESSAGE
      - TEXT

Let us also assume that this context is mapped to the view 'MAIN' as shown below



Here, we observe that the nodes EID and ERROR\_MESSAGE are mapped directly to the view context and the Node PROJECTS is independant and not mapped to component controller context.

Here, the Node EID and ERROR\_MESSAGE have a cardinality of 1..1 while the node PROJECTS has a cardinality of 1..n

## Code

### Scenario 1: Code to obtain values from the context node to variable

```

DATA lo_nd_eid TYPE REF TO if_wd_context_node.
DATA lo_el_eid TYPE REF TO if_wd_context_element.
DATA ls_eid TYPE wd_this->element_eid.
DATA lv_id LIKE ls_eid-id.
lo_nd_eid = wd_context->get_child_node( name = 'EID').
lo_el_eid = lo_nd_eid->get_element( ).
lo_el_eid->get_attribute(
  EXPORTING
    name = `ID`
  IMPORTING
    value = lv_id ).

```

**Note:** This code gets the value within the attribute ID of the node EID and places it in the local variable lv\_id.

### Scenario 2: Code to write values from variable to context node

```

DATA lo_nd_error_message TYPE REF TO if_wd_context_node.
DATA lo_el_error_message TYPE REF TO if_wd_context_element.
DATA ls_error_message TYPE wd_this->element_error_message.
DATA lv_text LIKE ls_error_message-text.
lo_nd_error_message = wd_context->get_child_node( name = 'ERROR_MESSAGE' ).
lo_el_error_message = lo_nd_error_message->get_element( ).
lo_el_error_message->set_attribute(
  EXPORTING
    name = `TEXT`
    value = lv_text ).

```

**Note:** This code changes the value within the attribute TEXT of the node ERROR\_MESSAGE from the value in the local variable lv\_text.

### Scenario 3: Code to obtain values from the context node to local internal table

```

DATA lo_nd_projects TYPE REF TO if_wd_context_node.
DATA lo_el_projects TYPE REF TO if_wd_context_element.
DATA ls_projects TYPE wd_this->elements_projects.
DATA wa_projects TYPE wd_this->element_projects.

lo_nd_projects = wd_context->get_child_node( name = wd_this->wdctx_projects ).
lo_el_projects = lo_nd_projects->get_element( ).
CALL METHOD lo_nd_projects->get_static_attributes_table
  IMPORTING
    table = ls_projects.

```

**Note:** This code gets the internal table within the context and copies it into the internal table ls\_projects.

If my context node is named Node1, then I can declare an internal table with the same type as the that of the node by declaring it of type wd\_this->elements\_node1. And a work area of the same type can be declared similarly by making it of type wd\_this->element\_node1.

#### Scenario 4: Code to write values from local internal table to context node

```
Data project type standard table of dpr_ts_Favourites_d.  
DATA wa type dpr_ts_Favourites_d.  
DATA lo_nd_projects TYPE REF TO if_wd_context_node.  
DATA lo_el_projects TYPE REF TO if_wd_context_element.  
DATA it_table type wd_this->elements_projects.  
DATA wa_table type wd_this->element_projects.  
lo_nd_projects = wd_context->get_child_node( name = wd_this->wdctx_projects ).  
lo_el_projects = lo_nd_projects->get_element( ).  
lo_nd_projects->bind_table( it_table ).
```

**Note:** This code binds the table `it_table` to the context node 'PROEJCTS'. It is important to note that the type of the internal table and the context must be the same. It is ensured as mentioned in the below note.

If my context node is named `Node1`, then I can declare an internal table with the same type as the that of the node by declaring it of type `wd_this->elements_node1`. And a work area of the same type can be declared similarly by making it of type `wd_this->element_node1`.

In the above code sample, since our node is named `Projects`, the internal table and work area are declared as `wd_this->elements_projects` and `wd_this->element_projects` respectively.

## 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.