









```
METHOD zif_gw_methods~get_entity.
```

```
TRY.
```

```
  CASE iv_source_name.
```

```
    WHEN 'SalesOrder'.
```

```
      zcl_demo_customer=>get(
```

```
        zcl_demo_salesorder=>get(
```

```
          CONV #( it_key_tab[ name = 'SalesOrderId' ]-value )
```

```
        )->get_kunnr( )
```

```
      )->zif_gw_methods~map_to_entity( REF #( er_entity ) ).
```

```
    WHEN OTHERS.
```

```
      zcl_demo_customer=>get(
```

```
        CONV #( it_key_tab[ name = 'CustomerId' ]-value )
```

```
      )->zif_gw_methods~map_to_entity( REF #( er_entity ) ).
```

```
  ENDCASE.
```

```
CATCH zcx_demo_bo cx_sy_itab_line_not_found INTO DATA(exception).
```

```
  RAISE EXCEPTION TYPE /iwbsp/cx_mgw_busi_exception
```

```
  EXPORTING
```

```
    textid      = /iwbsp/cx_mgw_busi_exception=>business_error
```

```
    previous    = exception
```

```
    message     = || exception->get_text( ) ||.
```

```
ENDTRY.
```

```
ENDMETHOD.
```

ZIF\_ENTITY

Use salesorder  
key to get  
customer key





# ZIF\_GW\_METHODS~GET\_ENTITY

```
METHOD zif_gw_methods~get_entity.
```

```
TRY.
```

```
  CASE iv_source_name.
```

```
    WHEN 'SalesOrder'.
```

```
      zcl_demo_customer=>get(
```

```
        zcl_demo_salesorder=>get(
```

```
          CONV #( it_key_tab[ name = 'SalesOrderId' ]-value )
```

```
        )->get kunnr( )
```

```
      )->zif_gw_methods~map_to_entity( REF #( er_entity ) ).
```

```
    WHEN OTHERS.
```

```
      zcl_demo_customer=>get(
```

```
        CONV #( it_key_tab[ name = 'CustomerId' ]-value )
```

```
      )->zif_gw_methods~map_to_entity( REF #( er_entity ) ).
```

```
  ENDCASE.
```

```
CATCH zcx_demo_bo cx_sy_itab_line_not_found INTO DATA(exception).
```

```
  RAISE EXCEPTION TYPE /iwbsp/cx_mgw_busi_exception
```

```
  EXPORTING
```

```
    textid      = /iwbsp/cx_mgw_busi_exception=>business_error
```

```
    previous    = exception
```

```
    message     = || exception->get_text( ) ||.
```

```
ENDTRY.
```

```
ENDMETHOD.
```

Use salesorder  
key to get  
customer key

# CALL\_ALL\_GETTERS

```
DATA: struct_descr TYPE REF TO cl_abap_structdescr,  
      parameter    TYPE abap_parmbind,  
      parameters   TYPE abap_parmbind_tab.  
  
ASSIGN entity->* TO FIELD-SYMBOL(<entity>).  
  
TRY.  
  struct_descr ?= cl_abap_structdescr=>describe_by_data_ref( entity ).  
  LOOP AT struct_descr->components REFERENCE INTO DATA(component).  
    CLEAR parameters.  
    TRY.  
      ASSIGN COMPONENT component->name OF STRUCTURE <entity> TO FIELD-SYMBOL(<comp>).  
      IF sy-subrc = 0.  
        parameter-name = component->name.  
        parameter-kind = cl_abap_objectdescr=>returning.  
        GET REFERENCE OF <comp> INTO parameter-value.  
        INSERT parameter INTO TABLE parameters.  
        DATA(method_name) = |GET_{ component->name }|.   
        CALL METHOD me->(method_name)  
          PARAMETER-TABLE parameters.  
      ENDIF.  
    CATCH cx_sy_dyn_call_error.  
  ENDTRY.  
ENDLOOP.  
CATCH cx_root.  
ENDTRY.
```

