

# SAP 3D Visual Enterprise View Integration Framework in SAP GUI 1.0



# Typographic Conventions

Type Style	Description
<i>Example</i>	Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Textual cross-references to other documents.
<b>Example</b>	Emphasized words or expressions.
EXAMPLE	Technical names of system objects. These include report names, program names, transaction codes, table names, and key concepts of a programming language when they are surrounded by body text, for example, SELECT and INCLUDE.
Example	Output on the screen. This includes file and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.
<b>Example</b>	Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.
<b>&lt;Example&gt;</b>	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.
<b>EXAMPLE</b>	Keys on the keyboard, for example, <b>F2</b> or <b>ENTER</b> .

---

# Document History

Version	Date	Change
1.0	2014-02-20	Initial Document

---

# Table of Contents

<b>1</b>	<b>Introduction .....</b>	<b>5</b>
<b>2</b>	<b>Prerequisites.....</b>	<b>6</b>
2.1	Development Skills .....	6
2.2	System Requirements .....	6
2.3	Supported OS .....	6
2.4	Edit Windows Registry .....	7
<b>3</b>	<b>Integration Framework of SAP 3D Visual Enterprise Viewer in GUI Applications.....</b>	<b>8</b>
3.1	Components.....	8
3.1.1	GUI Control Framework.....	8
3.1.2	GUI Viewer .....	8
3.2	Architecture diagram .....	9
3.3	Implementation .....	10
3.3.1	Methods of CL_GUI_VIEWER_MODEL .....	10
3.3.2	Events of CL_GUI_VIEWER_MODEL .....	12
<b>4</b>	<b>Example - Sample Application.....</b>	<b>13</b>
4.1	File Loading Options.....	14
4.2	Metadata Functions.....	17
4.3	Conversations in File .....	19
4.4	Visualization Options.....	20
4.5	Adding or Removing Parts .....	21

---

# 1 Introduction

This document describes the basic system setup, and it guides developers to adopt integration framework of SAP 3D Visual Enterprise Viewer in GUI applications.

---

## 2 Prerequisites

The following sections contain a list of the prerequisites required before integration.

### 2.1 Development Skills

You have knowledge of ABAP OO.

### 2.2 System Requirements

The system requirements are the following:

- SAP 3D Visual Enterprise Viewer 7.1 or above Enhancement Package
- SAP GUI Version 7.2 and above
- The Viewer Integration framework is available in the following releases of ERP 6.0:
  - EHP5 SP10 and above
  - EHP6 SP07 and above
  - EHP7 SP01 and above

### 2.3 Supported OS

You have one of the following operating systems:

- Windows XP (sp3 32) and 64 bit
- Vista (32 and 64 bit)
- Windows 7 (32 and 64 bit)

## 2.4 Edit Windows Registry

You have registered the viewer **CLSID** in Windows registry with the following keys using a .reg file:



Windows Registry Editor Version 5.00

- [HKEY\_CLASSES\_ROOT\SAP.VEViewerForABAP]  
@="DV GAC Control"
- [HKEY\_CLASSES\_ROOT\SAP.VEViewerForABAP\CLSID]  
@="{B2058416-94A8-4966-9FAF-49E7D6E22F6D}"
- [HKEY\_CLASSES\_ROOT\SAP.VEViewerForABAP\CurVer]  
@="SAP.VEViewerForABAP.1"
- [HKEY\_CLASSES\_ROOT\SAP.VEViewerForABAP.1]  
@="DV GAC Control"
- [HKEY\_CLASSES\_ROOT\SAP.VEViewerForABAP.1\CLSID]  
@="{B2058416-94A8-4966-9FAF-49E7D6E22F6D}"

## 3 Integration Framework of SAP 3D Visual Enterprise Viewer in GUI Applications

### 3.1 Components

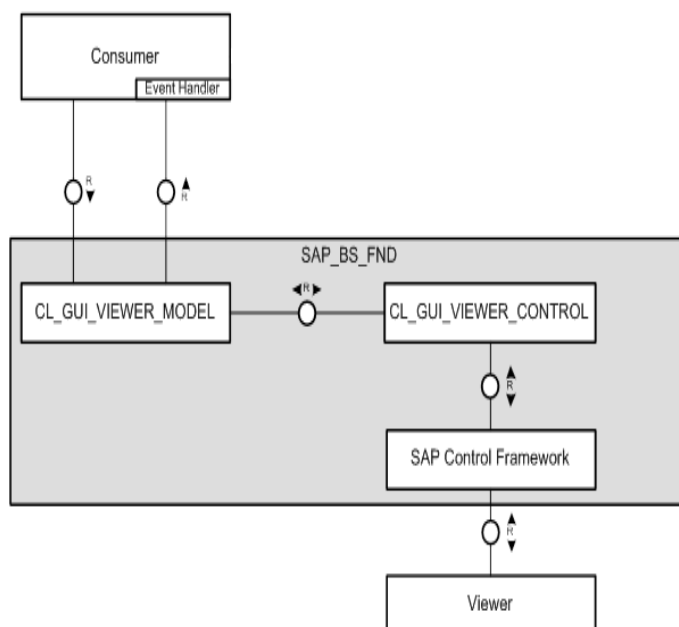
Source application: Application adopting SAP 3D Visual Enterprise Viewer in SAP GUI

#### 3.1.1 GUI Control Framework

This component is a standard SAP shipment which provides a framework for enabling and communicating through ActiveX Control.

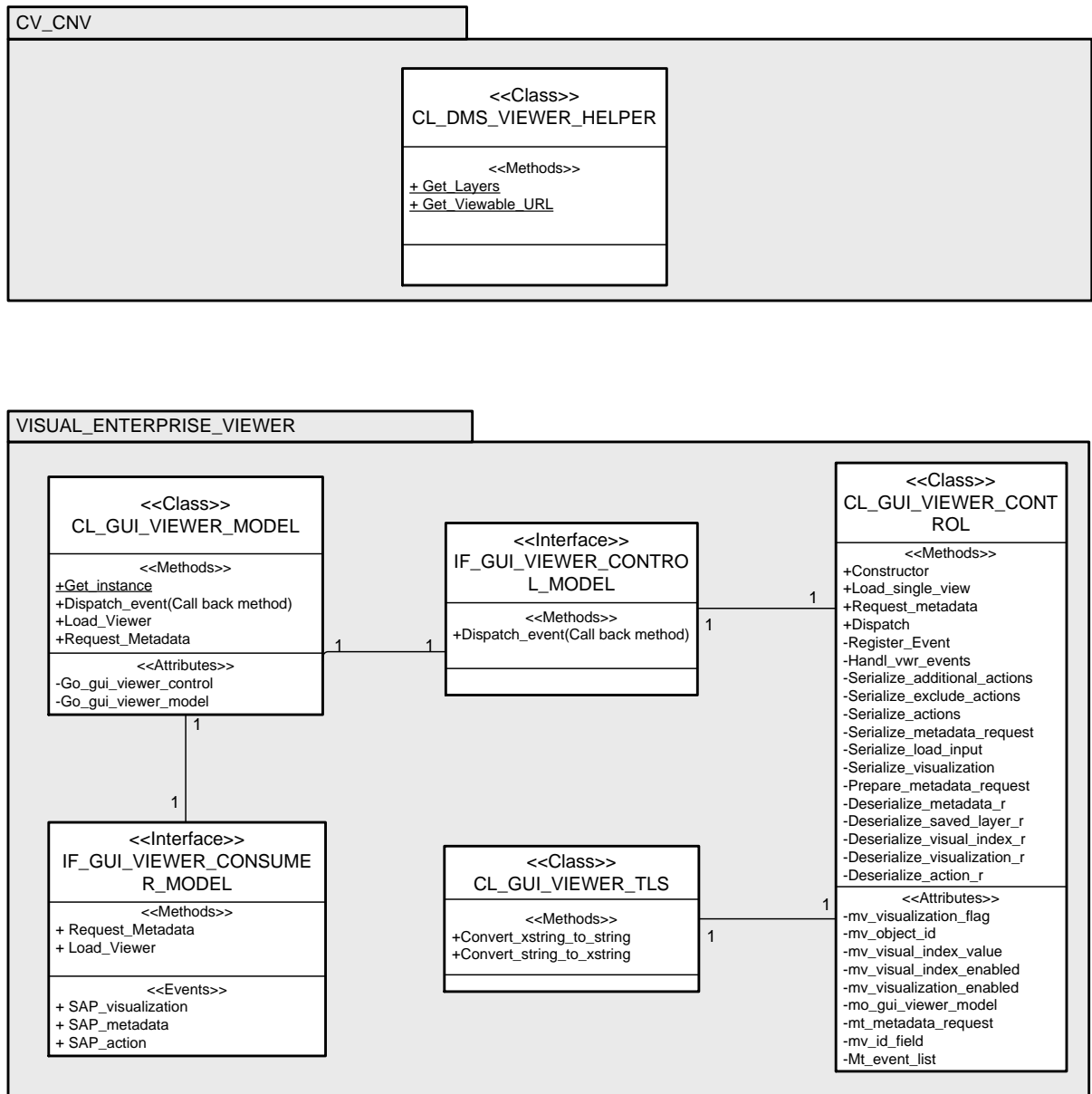
#### 3.1.2 GUI Viewer

This is a reusable component which communicates with SAP 3D Visual Enterprise Viewer through GUI Control Framework using GAC interface.





## 3.2 Architecture diagram



## 3.3 Implementation

Class `CL_GUI_VIEWER_MODEL` contains methods and events exposed from the viewer. This class is the touch point for consumer applications to integrate with SAP 3D Visual Enterprise Viewer. The class implements the consumer model interface `IF_GUI_VIEWER_CONSUMER_MODEL`.

### 3.3.1 Methods of `CL_GUI_VIEWER_MODEL`

The following table lists the various methods of the class `CL_GUI_VIEWER_MODEL`:

Method	Description
LOAD_VIEWER	<p>Loads the viewer with a monolithic viewable file in an application when called with appropriate parameters. Before calling this method, call <code>GET_INSTANCE ()</code>, which creates and returns a new instance or an existing instance of interface <code>IF_GUI_VIEWER_CONSUMER_MODEL</code>. You have to pass the reference of GUI container object where the viewer is to be embedded along with the file path of the viewable.</p> <p>Parameters of this method are as follows:</p> <ul style="list-style-type: none"><li>• <code>IT_LAYER_LIST</code> is used to load existing redlining</li><li>• <code>IT_ACTIONS</code> is used to create new application specific functions on the viewer</li><li>• <code>IT_EXCLUDE_ACTIONS</code> is used to exclude standard actions of the viewer</li></ul>
LOAD_ASSEMBLY	<p>Loads the viewer with a document assembly or multiple originals connected through transformation matrices.</p> <p>Parameters of this method are as follows:</p> <ul style="list-style-type: none"><li>• <code>IS_ROOT</code> holds the root reference of the assembly,</li><li>• <code>IT_SUBPARTS</code> is used to specify the part originals forming the assembly. The following are the key fields for this table:<ul style="list-style-type: none"><li>◦ <code>POSID</code> is a key that has to be generated and passed for the root and subparts information. This <code>POSID</code> is used to uniquely identify each part in the viewer.</li></ul></li></ul> <p><b>Note</b></p> <p>It is important to mention the <code>PARENTPOSIDs</code> (<code>POSID</code> of the parent sub assembly or assembly) till the sub-assembly in the document structure has been added to Viewer scene tree in the parameter <code>IT_SUBPARTS</code>,</p> <ul style="list-style-type: none"><li>◦ <code>URI</code> specifies the file path of the part file.</li><li>◦ <code>IT_TRAFO</code> is used to pass the transformation matrices of the root and the subpart, which specifies the coordinates for the geometrical positioning of parts in the viewer.</li></ul>

Method	Description
SET_METADATA	Tags any kind of business data to the RH original. The consumer must specify the <code>POSID</code> , which uniquely identifies a part in an assembly loaded using the <code>LOAD_ASSEMBLY</code> method and the metadata which has to be tagged to this part. With the <code>LOAD_SINGLE</code> method, the <code>POSID</code> is the same for the monolithic file. This metadata is temporarily set for the session, and there is no possibility to save metadata information from SAP GUI.
REQUEST_METADATA	Places a request to the viewer to retrieve the business data or metadata embedded in the original. It does not provide metadata information, but only places a request to SAP 3D Visual Enterprise Viewer for metadata. It is the event <code>SAP_METADATA</code> that provides the retrieved information.  Metadata can be embedded in the original in the following ways: <ul style="list-style-type: none"> <li>• The DMS Document which contains the original has undergone a conversion process (CAD to RH) in SAP 3D Visual Enterprise Generator. The DMS takes care of sending the business data maintained in customizing to the VEG server so that the original is enriched with business data.</li> <li>• SAP 3D Visual Enterprise Author has been used to edit the original and add metadata information.</li> <li>• <code>SET_METADATA</code> method has been used to temporarily embed metadata onto the original in the SAP session.</li> </ul>
REMOVE_PART	Removes a part from the scene by passing <code>POSID</code> key. This does not alter the RH original that is loaded in the viewer. The part is also removed from the scene tree of the viewer.
ADD_ITEMS	Adds or replaces a part or parts in the viewable by passing the part file information and transformation matrices. Ensure that you have loaded the root instance and/or other parts and their transformation matrices using the <code>LOAD_ASSEMBLY</code> method.
SET_VISUALIZATION	Highlights or hides a part or parts in the viewable. <code>POSID</code> is used to uniquely identify a part in an assembly ( <code>LOAD_ASSEMBLY</code> ) and <code>IV_IDFIELD</code> is used to identify a part in the loaded monolith ( <code>LOAD_VIEWER_SINGLE</code> ).
SAVE_CHANGED_LAYER	Saves the changed markup layers in the original.
REQUEST_STEP_PROPERTIES	Places a request to the viewer to retrieve the work procedure steps. It does not provide the work procedure information. The event <code>SAP_STEP_PROPERTIES</code> provides the retrieved information.

## 3.3.2 Events of CL\_GUI\_VIEWER\_MODEL

Various events from SAP 3D Visual Enterprise Viewer are exposed in this class. Event handler methods are not implemented by the framework. The consumer application has to implement the suitable event handler methods. The events are described in the following table.

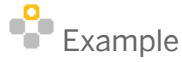
Event	Description
SAP_VISUALISATION	Informs that the part has been selected in SAP 3D Visual Enterprise Viewer
SAP_METADATA	Provides metadata information. This event gets fired after a request has been placed with the SAP 3D Visual Enterprise Viewer to provide metadata information through <code>GET_METADATA</code> method
SAP_ACTION	Informs the consumer that the source application event has been fired
VWR_MESSAGE	Reports error or success messages from SAP 3D Visual Enterprise Viewer to the application
SAP_LOAD_DONE	Informs that SAP 3D Visual Enterprise Viewer has successfully loaded or not
SAP_SET_LAYER_CHANGE	Informs that the layers are changed in SAP 3D Visual Enterprise Viewer
SAP_SAVE_LAYER_DONE	Informs that the layers are saved in SAP 3D Visual Enterprise Viewer
SAP_STEP_PROPERTIES	Provides step properties or work procedure information from SAP 3D Visual Enterprise Viewer
SAP_REMOVE	Informs that the part is removed from the loaded file in SAP 3D Visual Enterprise Viewer using the method <code>REMOVE_PART</code>



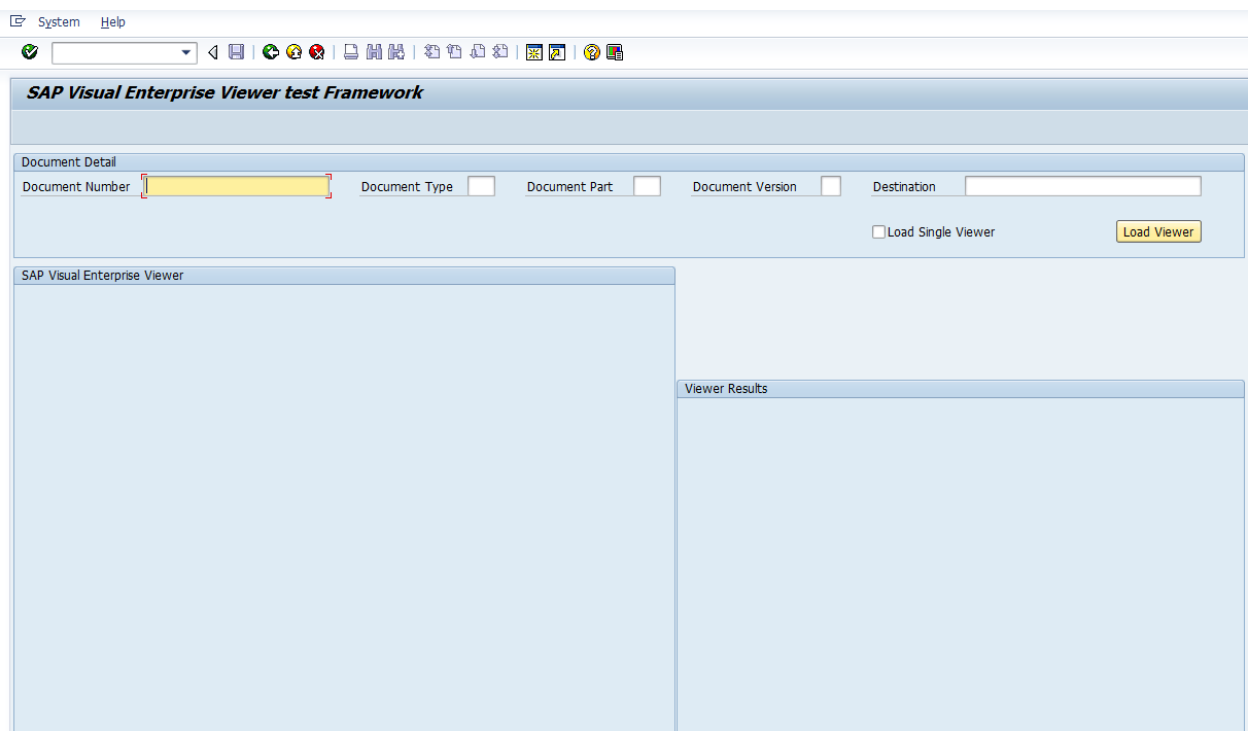
### Caution

Interfaces of the framework should not be re-implemented by the consumer.

## 4 Example - Sample Application



The following shows a sample program.



## 4.1 File Loading Options

As a first step we will load the viewer with the original on the screen. DMS Document details containing the originals should be entered on the screen to load the viewer in the screen. The checkbox *Load Single Viewer* allows the user to load the document using the method `LOAD_SINGLE` or `LOAD_ASSEMBLY`.

Additional parameters of load viewer

**New Action**

☐ Request Metadata ☐ Select Item

**Exclude Action**

☐ Delete-Markups ☐ Display-Grid ☐ Animate Options-Play ☐ Full Screen

☐ Camera View-Perspective ☐ Display-Cross Section ☐ Animate Options-End

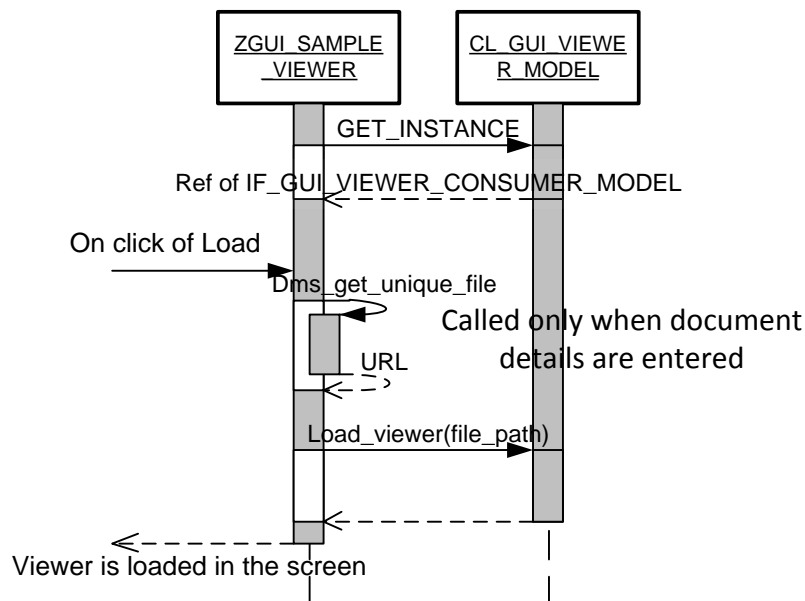
☐ Camera View-Orthographic ☐ Display-Markups ☐ Preferences-Roll Over Toolbar

**Additional Options**

☐ Display Existing Redlining ☐ Activate Visualization ☐ Activate Visual Index instead of Visualization

☐ Deactivate Redlining

Additional parameters to the load method (`LOAD_SINGLE` or `LOAD_ASSEMBLY`) can be passed in this pop up. New context menu actions can be created or existing viewer actions can be excluded. These parameters are optional. The viewer loads in the screen container area whose object reference is specified to the load method. Once the viewer is loaded, the event `SAP_LOAD_DONE` is triggered to inform whether the viewer loaded successfully.



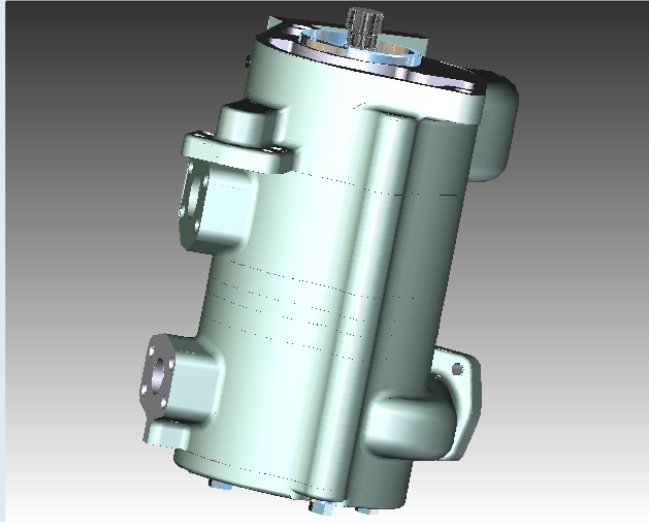
Sequence diagram on load

### SAP Visual Enterprise Viewer test Framework

#### Document Detail

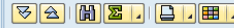
Document Number	<input type="text" value="OIL_PUMP"/>	Document Type	<input type="text" value="DRW"/>	Document Part	<input type="text" value="000"/>	Document Version	<input type="text" value="00"/>	Destination	<input type="text"/>
<input checked="" type="checkbox"/> Load Single Viewer <input type="button" value="Load Viewer"/>									

#### SAP Visual Enterprise Viewer



#### Viewer Actions

#### Viewer Results



#### Received Information

#### Viewer Messages

Sample code for `load_viewer`:

Pass the following parameters:

1. **File path**
2. **GUI container reference**
3. **Additional parameters in the pop up**
4. **Layer information**

```

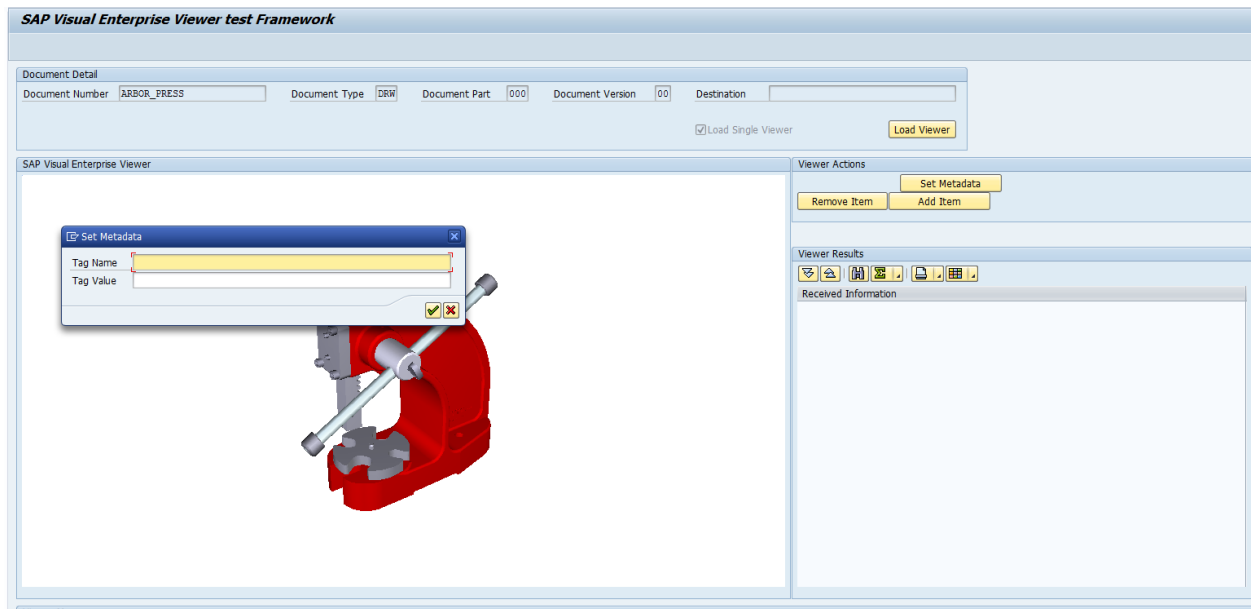
85 IF display_layer = abap_true.
86   CALL FUNCTION 'ZGET_DMS_LAYERS'
87     DESTINATION rfc_dest
88     EXPORTING
89       iv_fileid = gs_dir_file-file_id
90       dokar      = gr_dms_doc_key-dokar
91       doknr      = gr_dms_doc_key-doknr
92       dokvr      = gr_dms_doc_key-dokvr
93       doktl      = gr_dms_doc_key-doktl
94     IMPORTING
95       et_layers = lt_layer.
96 ENDIF.
97 * Assign parameter to local variable
98 lv_url      = gs_dir_file-url.
99 lv_file_name = gs_dir_file-file_name.
100
101 * Load Viewer with viewable
102 CALL METHOD gr_vwr_model->load_viewer
103   EXPORTING
104     io_parent      = gr_vwrcontainer
105     iv_filepath    = lv_url
106     iv_name        = lv_file_name
107     iv_id_field    = gv_id_field_stage
108     iv_visualization = activate_visualization
109     iv_visual_index = activate_visual_index
110     it_layer_list  = lt_layer
111     iv_create_layer = create_layer
112     iv_deactivate_redlining = deactivate_redlining
113     iv_display_all_markup = display_layer
114     it_actions     = gt_action
115     it_exclude_action = gt_exclude
116   IMPORTING
117     et_message     = gt_message.
118 PERFORM handle_message USING gt_message.
119 ENDFORM. " LOAD SINGLE VIEWER

```



## 4.2 Metadata Functions

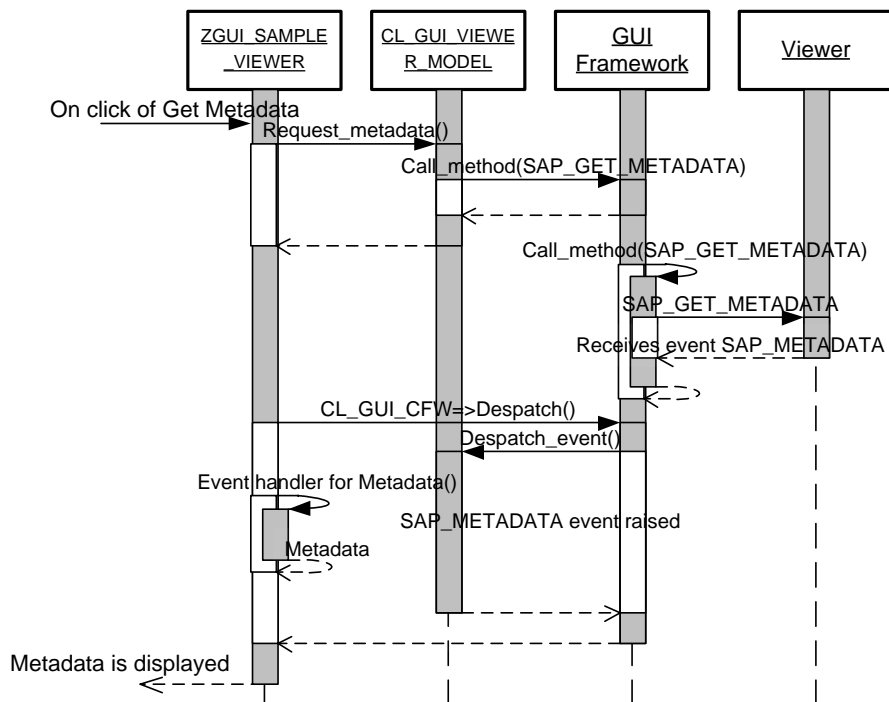
The user can choose to Set Metadata using the SET METADATA function. Attribute name and values have to be passed to temporarily embed the metadata on the file.



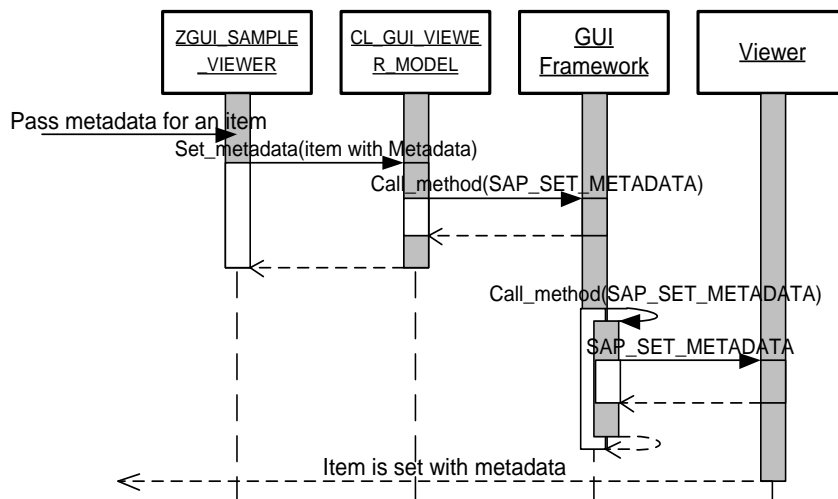
```
12 FORM set_metadata .
13   CALL SCREEN 1003 STARTING AT 10 10.
14
15   CALL METHOD gr_vwr_model->set_metadata
16     EXPORTING
17       it_metadata = gt_metadata
18     IMPORTING
19       et_message = gt_message.
20
21   PERFORM handle_message USING gt_message.
22
23 ENDFORM.          " SET METADATA
```

Metadata on the file can be retrieved by the Get Metadata function and is displayed on the screen in an XML format. The user has to choose a part in the file, for which the metadata has to be retrieved. The viewer event SAP\_METADATA sends the metadata information to the application.

```
20 WHEN '&GETMETADATA'.
21   CALL METHOD gr_vwr_model->request_metadata
22     IMPORTING
23       et_message = gt_message.
```



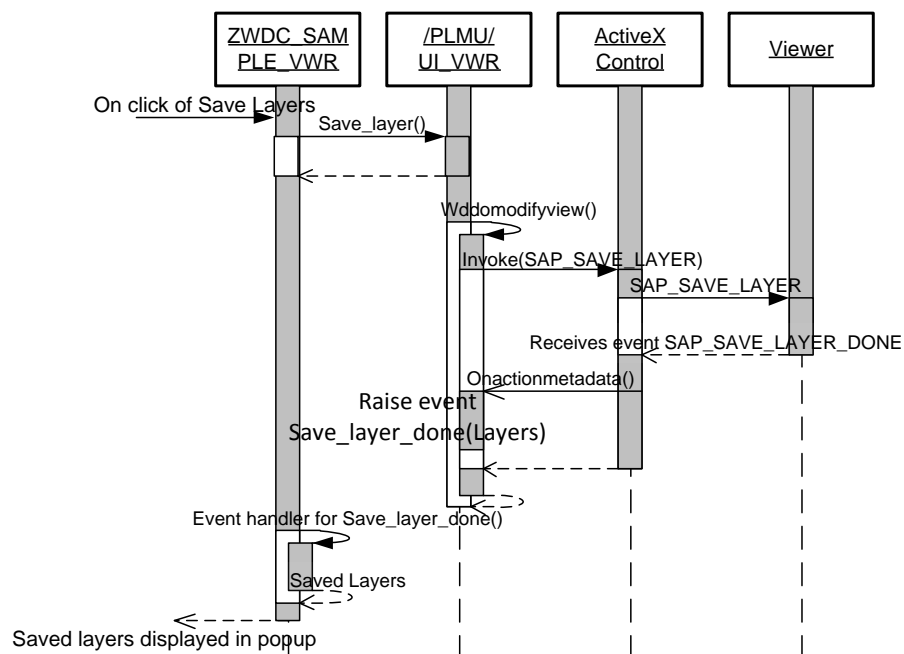
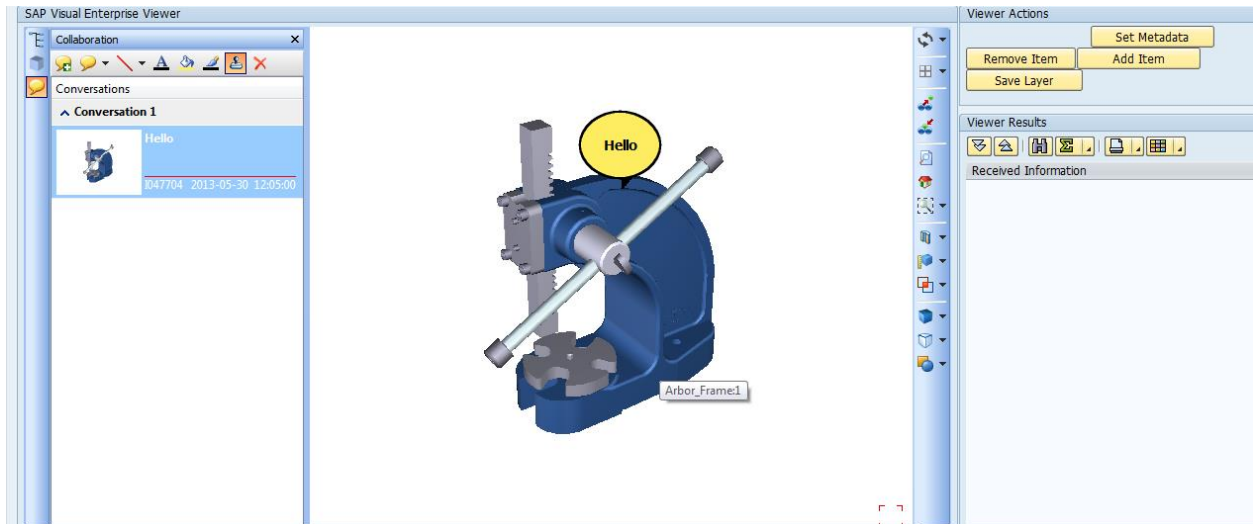
Sequence diagram GET\_METADATA



Sequence diagram SET\_METADATA

## 4.3 Conversations in File

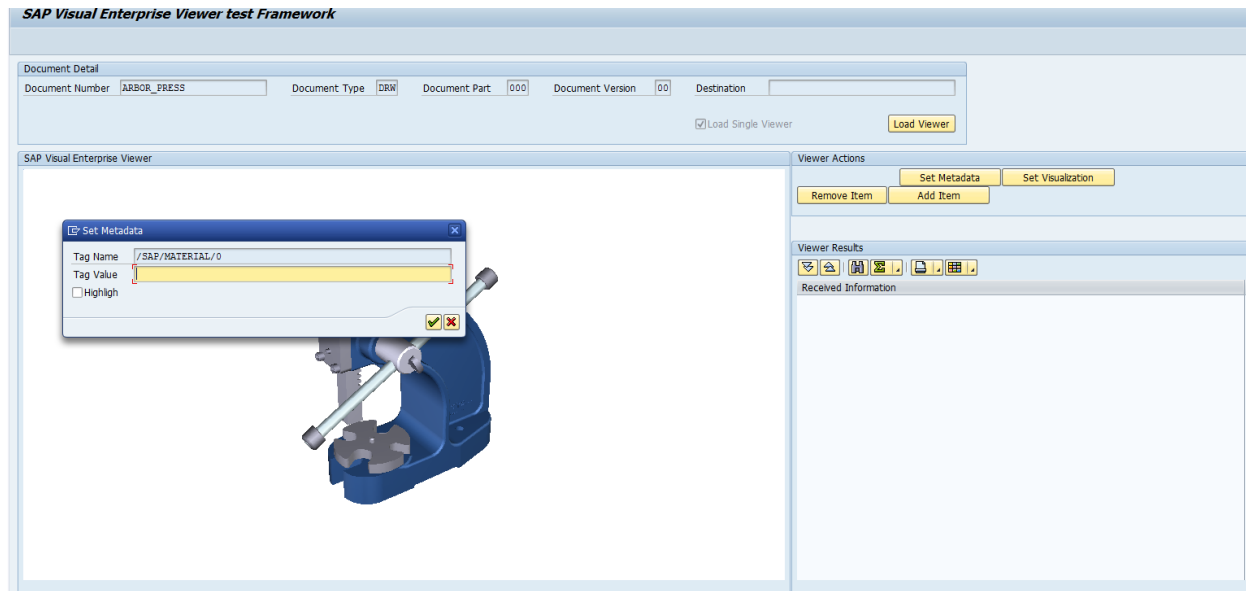
Conversations or comments can be created in the Viewer and saved to a backend store like KPRO using the [Save Layer](#) option. When a conversation is created or modified in the viewer, the event `SAP_SET_LAYER_CHANGE` informs that a layer has been modified or created with the viewer. The event `SAP_SAVE_LAYER_DONE` gives the layer information from the viewer to the consumer. Conversations can be deactivated through the file loading options.



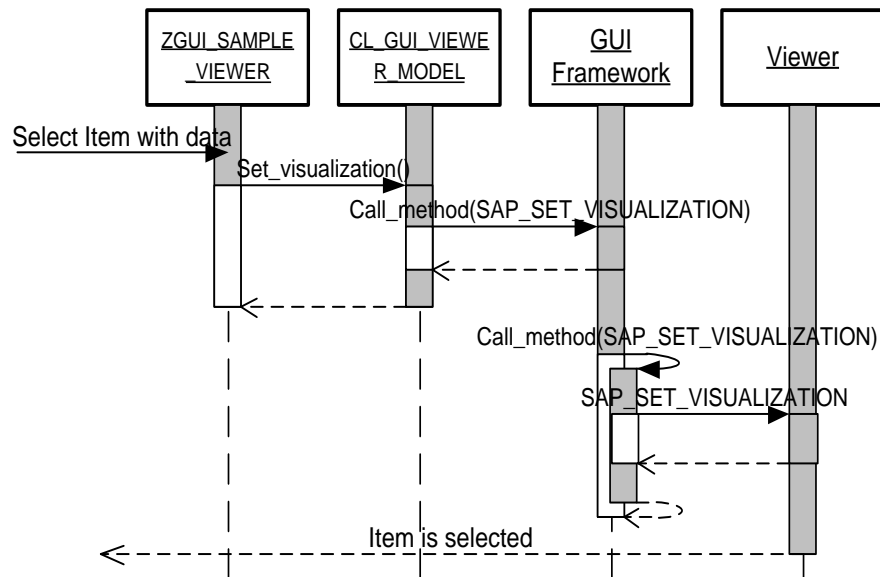
Sequence diagram SAVE\_LAYER

## 4.4 Visualization Options

The user can highlight parts or hide and show parts using `Set Visualization` by passing appropriate Metadata values in selection. While loading the file in the *Additional Options* pop-up, remember to *Activate Visualization* and pass the Metadata tag for selection. Related event `SAP_VISUALIZATION` informs whether part has been selected.



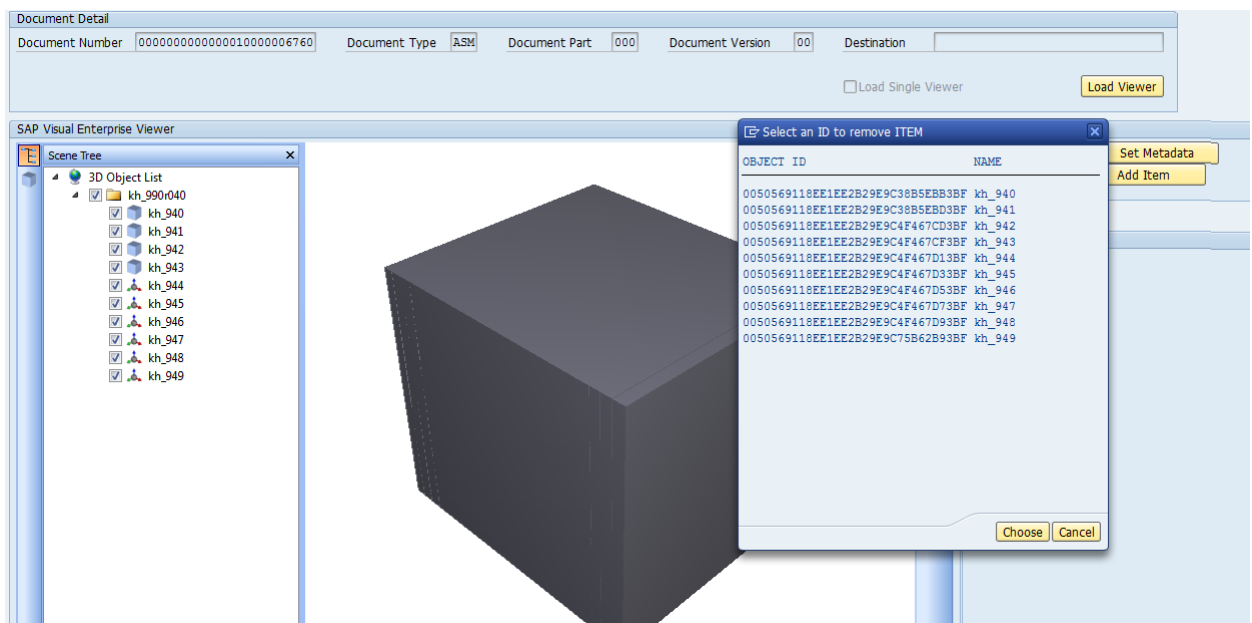
```
129 FORM set_visualization .
130     CALL SCREEN 1003 STARTING AT 10 10.
131
132     CALL METHOD gr_vwr_model->set_visualization
133     EXPORTING
134         it_visualization = gt_visualization
135     IMPORTING
136         et_message      = gt_message.
137
138     PERFORM handle_message USING gt_message.
139
140 ENDFORM.                                " SET VISUALIZATION
```



Sequence diagram SET\_VISUALIZATION

## 4.5 Adding or Removing Parts

Parts can be added or removed from the viewable using [Add Item](#) / [Remove Item](#). This works only when the viewable has been loaded using the `LOAD ASSEMBLY` method, and the parts which constitute the assembly have linked transformation matrices.



---

## Note

Viewer event handler methods must be implemented by the source application and are not provided by the framework.





[www.sap.com/contactsap](http://www.sap.com/contactsap)

**Material Number**

© 2014 SAP AG or an SAP affiliate company. All rights reserved.  
No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

National product specifications may vary.

These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries. Please see [www.sap.com/corporate-en/legal/copyright/index.epx#trademark](http://www.sap.com/corporate-en/legal/copyright/index.epx#trademark) for additional trademark information and notices.