



Open CASCADE Technology
7.2.0

Inspector

August 30, 2017

Contents

1	Introduction	2
1.1	Overview	2
1.2	Getting started	2
2	Inspector	3
2.1	Overview	3
2.2	DFBrowser Plugin	3
2.2.1	Overview	4
2.2.2	Elements	4
2.2.3	Elements cooperation	8
2.2.4	TopoDS_Shape export	10
2.3	VInspector Plugin	10
2.3.1	Overview	10
2.3.2	Elements	11
2.3.3	Elements cooperation	12
2.4	ShapeView Plugin	13
2.4.1	Overview	13
2.4.2	Elements	14
2.4.3	Elements cooperation	14
3	Common controls	15
3.1	3D View	15
3.1.1	Overview	15
3.1.2	Elements	16
4	TInspectorEXE sample	17
5	Launch in DRAW Test Harness	19
6	Using in a custom application	20
7	Build procedure	21
8	Sources and packaging	22
9	Glossary	23
9.1	TDF_Attribute Simple types	23
9.2	TDF_Attribute List types	23
9.3	TDF_Attribute Array types	23
9.4	XDE tree node ID description	23

1 Introduction

This manual explains how to use Inspector.

1.1 Overview

Inspector is a Qt-based library that provides functionality to interactively inspect low-level content of the OCAF data model, OCCT viewer and Modelisation Data. This component is aimed to assist the developers of OCCT-based applications to debug the problematic situations that occur in their applications.

Inspector has a plugin-oriented architecture. The current release contains the following plugins:

Plugin	OCCT component	Root class of OCCT investigated component
DFBrowser	OCAF	TDocStd_Application
VInspector	Visualization	AIS_InteractiveContext
ShapeView	Modelisation Data	TopoDS_Shape

Each plugin implements logic of a corresponding OCCT component.

Each of the listed plugins is embedded in the common framework. The user is able to manage which plugins should be loaded by Inspector. Also he can extend number of plugins by implementing a new plugin.

1.2 Getting started

There are two launch modes:

1. Launch **TInspectorEXE** executable sample. For more details see [TInspectorEXE](#) section;
2. Launch DRAW, load plugin INSPECTOR, and use **tinspector** command. For more details see [Launch in DRAW Test Harness](#) section.

Note. If you have no Inspector library in your build directory, please make sure that OCCT is compiled with *BUILD_Inspector* option ON. For more details see [Build procedure](#).

2 Inspector

2.1 Overview

Inspector consists of the following components:

- **buttons** to activate the corresponding plugin;
- **view area** to visualize the plugin content.

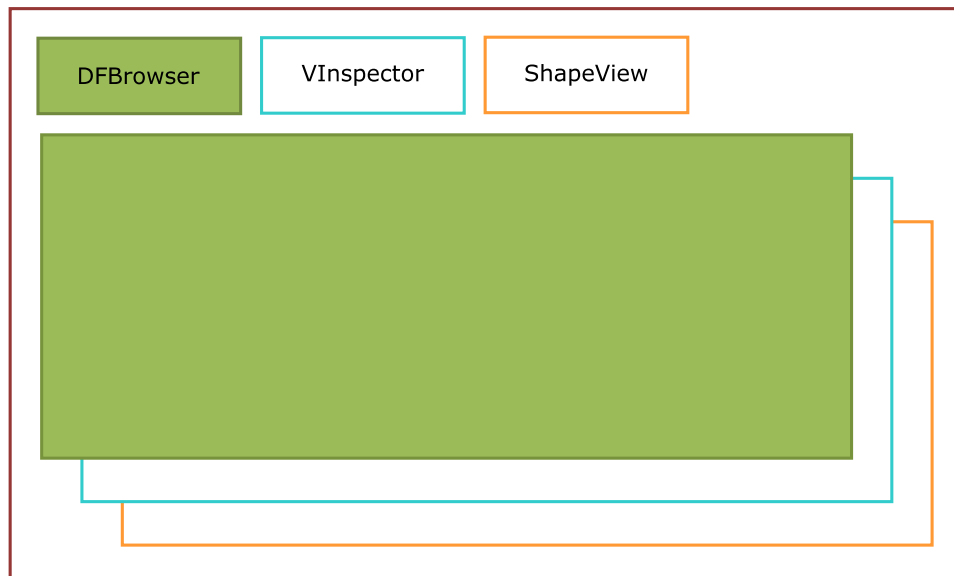


Figure 1: Plugins placement in Inspector

2.2 DFBrowser Plugin

2.2.1 Overview

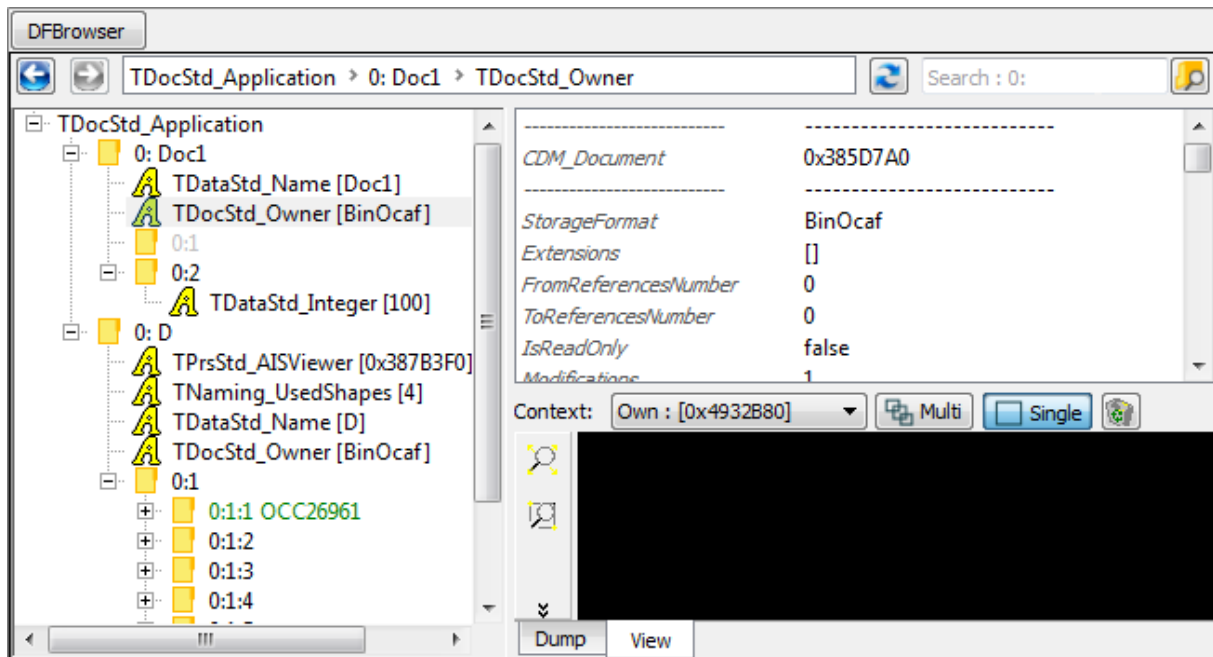


Figure 2: DFBrowser

This plugin visualizes content of TDocStd_Application in a tree view. It shows documents of the application, hierarchy of TDF_Labels, content of TDF_Attributes and interconnection between attributes (e.g. references). Additionally it has 3D view to visualize TopoDS_Shape elements stored in the document.

2.2.2 Elements

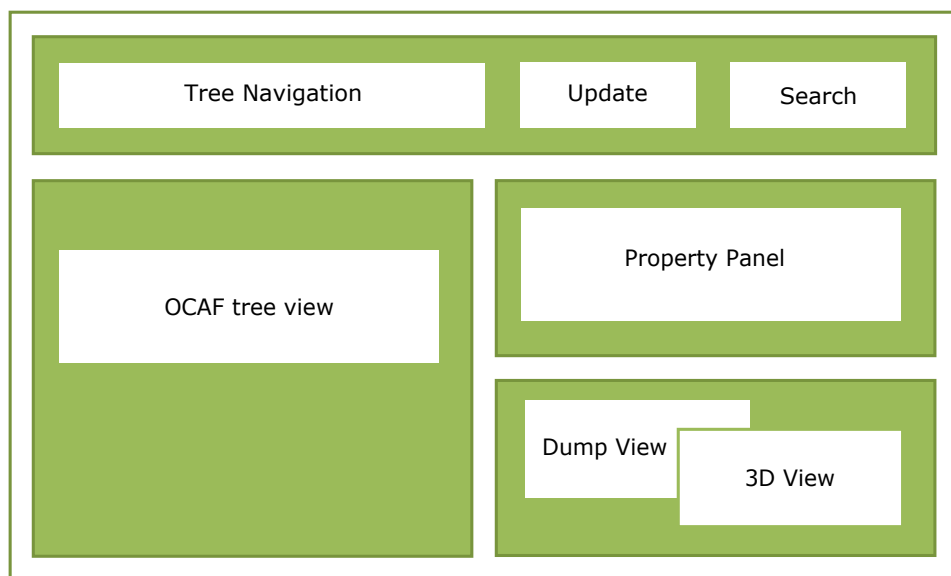


Figure 3: DFBrowser Elements

OCAF tree view

Each OCAF element has own tree view item:

Type	Tree item	Text	Description
TDocStd_Application	Application	TDocStd_Application	It is the root of tree view. Children are documents.
TDocStd_Document	Document	entry : name	It is a child of Application item. Children are Labels and Attributes items. Text view is an entry of the root label and the value of TDataStd_↔ Name attribute for the label if it exists.
TDF_Label	Label	entry : name	It is a child of a Document or another Label item. Children and text view are the same as for Document item.
TDF_Attribute	Attribute	attribute type [additional information]	It is a child of a Label. It has no children. Text view is the attribute type (DynamicType()->Name()) of TD↔ F_Attribute) and additional information (a combination of attribute values)

Additional information of TDF_Attributes:

Type	Text
TDocStd_Owner	[storage format]
TDataStd_AsciiString, TDataStd_Name, TDataStd_Real, other Simple types	[value]
TDataStd_BooleanList, TDataStd_ExtStringList, other List types	[value_1 ... value_n]
TDataStd_BooleanArray, TDataStd_ByteArray, other Array types	[value_1 ... value_n]
TDataStd_TreeNode	[tree node ID ==> Father()->Label()] (if it has father) or [tree node ID <== First()->Label()] (if it has NO father)
TDataStd_TreeNode(XDE)	[XDE tree node ID ==> Father()->Label()] (if it has father), [XDE tree Node ID <== label_1, ..., label_n] (if it has NO father)
TNaming_NamedShape	[shape type : evolution]
TNaming_UsedShapes	[map extent]

Custom color of items:

OCAF element Type	Color
TDF_Label	dark green , if the label has TDataStd_Name attribute, light grey if the label is empty (has no attributes on all levels of hierarchy), black otherwise
TNaming_NamedShape	dark gray for TopAbs_FORWARD orientation of TopoDS_Shape, gray for TopAbs_REVERSED orientation of TopoDS_Shape, black for other orientation

Context popup menu:

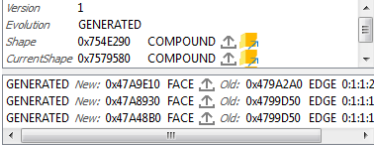
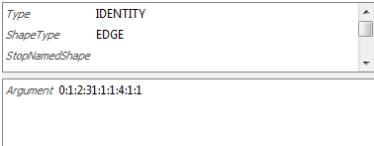
Action	Functionality
Expand	Expands the next two levels under the selected item
Expand All	Expands the whole tree of the selected item
Collapse All	Collapses the whole tree of the selected item

Property Panel

Property panel is used to display content of Label or Attribute tree view items. This control is used for content of Label or Attribute tree view items or Search result view. Information is usually shown in one or several tables.

TDF_Attribute has the following content in Property Panel:

Type	Description	Content
TDF_Label	a table of [entry or attribute name, value]	
TDocStd_Owner, Simple types, List types	a table of [method name, value]	
TDataStd_BooleanArray, TDataStd_ByteArray, other Array types	2 controls: * a table of [array bound, value], * table of [method name, value]	
TDataStd_TreeNode	2 controls: * a table of [Tree ID, value] (visible only if Tree ID() != ID()), * a tree view of tree nodes starting from Root() of the tree node. The current tree node has dark blue text.	
TDataStd_NamedData	tab bar of attribute elements, each tab has a table of [name, value]	
TNaming_UsedShapes	a table of all the shapes handled by the framework	

Type	Description	Content
TNaming_NamedShape	2 controls: * a table of [method name, value] including CurrentShape/OriginalShape methods result of TNaming_Tools, * an evolution table. Tables contain buttons for TopoDS_Shape export .	
TNaming_Naming	2 controls: * a table of TNaming_Name values, * a table of [method name, value]	

Dump view

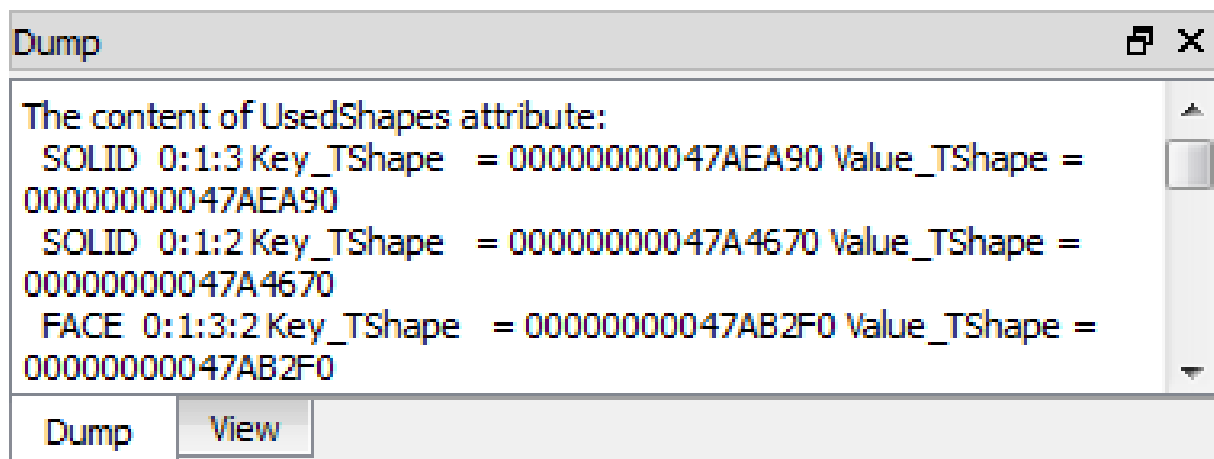


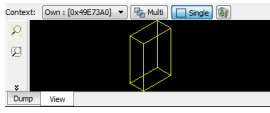
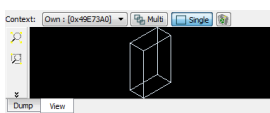
Figure 4: Dump of TDF_Attribute

Dump view shows result of **TDF_Attribute::Dump()** or **TDF_Label::Dump()** of selected tree view item.

3D view

3D View visualizes TopoDS_Shape elements of OCAF attribute via AIS facilities.

DFBrowser creates two kinds presentations depending on the selection place:

Kind	Source object	Visualization properties	View
Main presentation	Tree view item: TPrsStd_AIS↔ Presentation, TNaming_NamedShape, TNaming_Naming	Color: a default color for shape type of the current TopoDS_Shape	
Additional presentation	References in Property panel	Color: white	

Tree Navigation

Tree Navigation shows a path to the item selected in the tree view. The path is a sequence of label entries and attribute type name. Each element in the path is selectable - the user can click on it to select the corresponding tree view item.

Navigation control has buttons to go to the previous and the next selected tree view items.

Update Button

Update button synchronizes content of tree view to the current content of OCAF document that could be modified outside.

Search

The user can search OCAF element by typing:

- TDF_Label entry,
- TDF_Attribute name,
- TDataStd_Name and TDataStd_Comment attributes value.

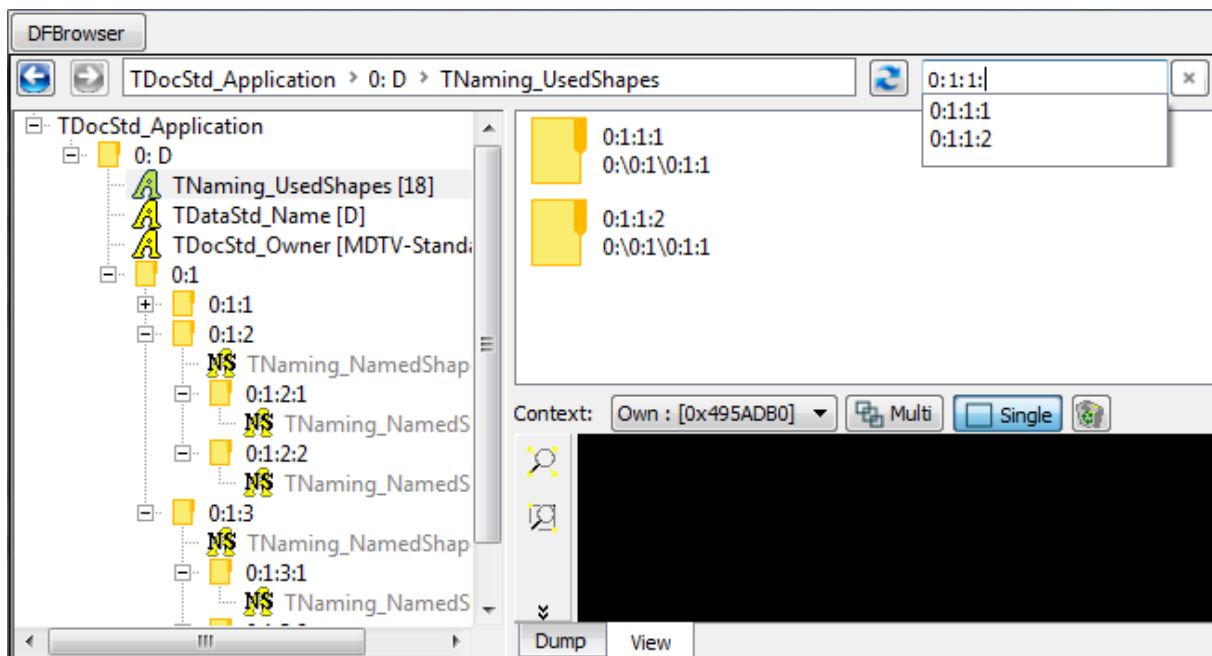


Figure 5: Search

As soon as the user confirms the typed criteria, the Property panel is filled by all satisfied values. The user can click a value to highlight the corresponding tree view item. By double click the item will be selected.

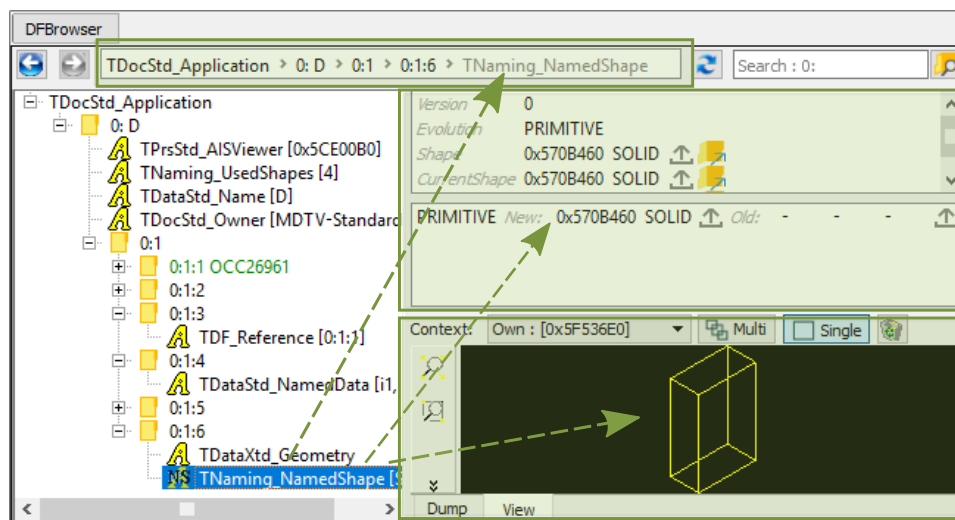
2.2.3 Elements cooperation

Tree item selection

Selection of tree view item updates content of the following controls:

- Navigation line
- Property Panel
- 3D View (if it is possible to create an interactive presentation)

- Dump View



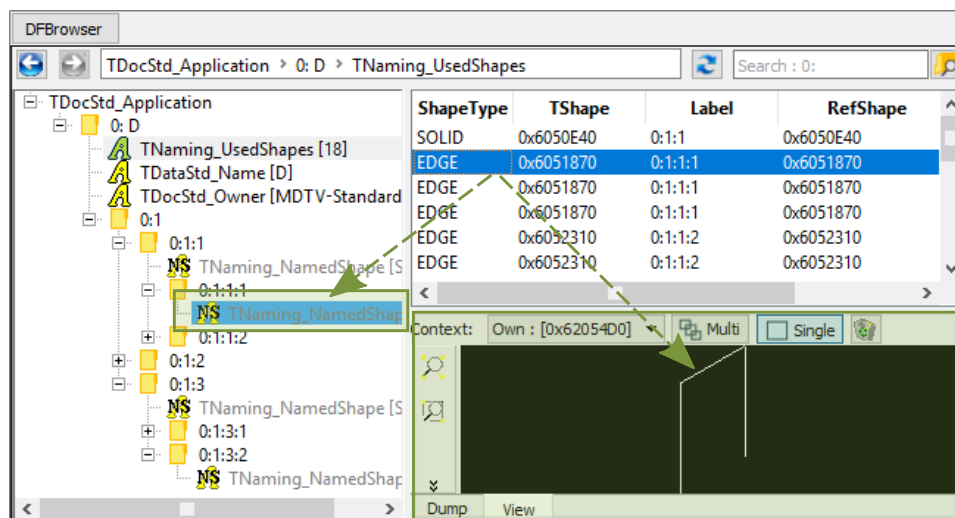
Property Panel item selection

If property panel shows content of TDF_Label:

- selection of the table row highlights the corresponding item in tree view,
- double click on the table row selects this item in tree view.

If property panel shows content of TDF_Attribute that has reference to another attribute, selection of this reference:

- highlights the referenced item in TreeView,
- displays additional presentation in 3D view if it can be created.



Attributes having references:

Type	Reference	Additional presentation
TDF_Reference	TDF_Label	
TDataStd_ReferenceArray, TDataStd_ReferenceList, TNaming_Naming	one or several TDF_Label in a container	

Type	Reference	Additional presentation
TDataStd_TreeNode	TDF_Label	
TNaming_NamedShape	TDF_Label in Evolution table	selected TopoDS_Shapes in property panel tables
TNaming_UsedShapes	one or several TNaming_NamedShape	TopoDS_Shapes of selected T_Naming_NamedShape

2.2.4 TopoDS_Shape export

Property panel of TNaming_NamedShape attribute has controls to export TopoDS_Shape to:

- BREP. The save file dialog is started to enter the result file name,
- [ShapeView](#) plugin. Dialog about exporting element to ShapeView is shown with a possibility to activate this plugin immediately.

2.3 VInspector Plugin

2.3.1 Overview

VInspector									
<div><div>Update</div><div>Select Presentations</div><div>Select Owners</div></div>									
Name	Size	Pointer	Shape type	Selection	Base Sensitive	Sensitivity	SubElement	Deviation/Deflectiton/Update/Priority	
▼ AIS_InteractiveContext	0			1			17 / 16	0,001	
> AIS_Shape	1	0x5C8FCE0	SOLID					0.001 / 0.96 / 1	
▼ AIS_Shape	3	0x5C0EE60	SOLID	1				0.001 / 0.4 / 1	
> SelectMgr_Selection	6		SHAPE	Deactivated		2		None / None	
▼ SelectMgr_Selection	8		VERTEX	Activated : 1		12		None / None	
> SelectMgr_SensitiveEntity		0x5C15C80	VERTEX	true	StdSelect_BRepOwner	12	1	8	
> SelectMgr_SensitiveEntity		0x5C2CC80	VERTEX	true	StdSelect_BRepOwner	12	1	8	
> SelectMgr_SensitiveEntity		0x5C2CE70	VERTEX	true	StdSelect_BRepOwner	12	1	8	
> SelectMgr_SensitiveEntity		0x5C24E30	VERTEX	true	StdSelect_BRepOwner	12	1	8	
> SelectMgr_SensitiveEntity		0x5C25020	VERTEX	true	StdSelect_BRepOwner	12	1	8	
> SelectMgr_SensitiveEntity		0x5C34A40	VERTEX	true	StdSelect_BRepOwner	12	1	8	
> SelectMgr_SensitiveEntity		0x5C34C30	VERTEX	true	StdSelect_BRepOwner	12	1	8	
Name	Size	Pointer	Shape type	AIS Name		Selected/Highlighted			
History	0								

Figure 6: VInspector

It visualizes interactive objects displayed in AIS_InteractiveContext in a tree view with computed selection components for each presentation. It shows the selected elements in the context and allows to select these elements.

2.3.2 Elements

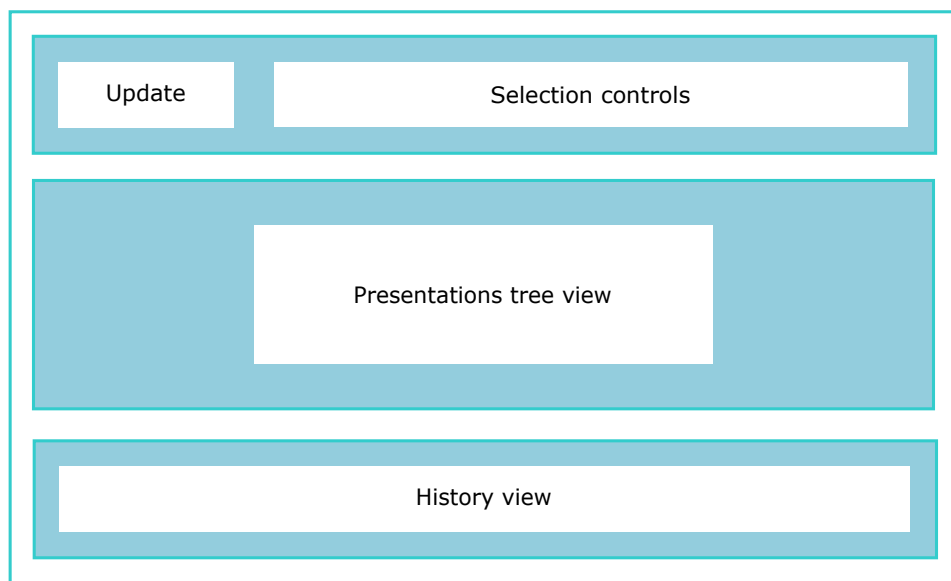


Figure 7: VInspector Elements

Presentations tree view

It shows presentations and selection computed of them. Also, the view has columns with information about state of visualization elements.

VInspector tree items.

Type	Description
AIS_InteractiveContext	It is the root of tree view. Children are interactive objects obtained by <i>DisplayedObjects</i> and <i>ErasedObjects</i> methods.
AIS_InteractiveObject	It is a child of AIS_InteractiveContext item. Children are SelectMgr_Selection obtained by iteration on <i>CurrentSelection</i>
SelectMgr_Selection	It is a child of AIS_InteractiveObject. Children are SelectMgr_SensitiveEntity obtaining by iteration on <i>Sensitive</i>
SelectMgr_SensitiveEntity	It is a child of SelectMgr_Selection. Children are SelectMgr_SensitiveEntity obtaining by iteration on <i>OwnerId</i>
SelectBasics_EntityOwner	It is a child SelectMgr_SensitiveEntity. It has no children.

Custom color of tree view items:

OCAF element Type	Column	What	Color
AIS_InteractiveObject	0	Text	dark gray , it is in <i>ErasedObjects</i> list of AIS_InteractiveContext, black otherwise
AIS_InteractiveObject, SelectMgr_SensitiveEntity, SelectBasics_EntityOwner	1	Background	dark blue , if there is a selected owner under the item, black otherwise
SelectMgr_Selection, SelectMgr_SensitiveEntity, SelectBasics_EntityOwner	all	Text	dark gray , if <i>SelectionState</i> of SelectMgr_Selection is not <i>SelectMgr_SOS_Activated</i> , black otherwise

Context popup menu in tree view:

Action	Item	Functionality
Export to ShapeView	AIS_InteractiveObject	Exports TopoDS_Shape of AIS_Interactive presentation to ShapeView plugin. It should be AIS_Shape presentation and ShapeView plugin should be registered in Inspector Dialog about exporting element to ShapeView is shown with a possibility to activate this plugin immediately.
Show	AIS_InteractiveObject	<i>Display</i> presentation in AIS_InteractiveContext
Hide	AIS_InteractiveObject	<i>Erase</i> presentation from AIS_InteractiveContext

Update

It synchronizes content of the plugin to the current state of AIS_InteractiveContext. It updates the presence of items and the current selection for the items.

Selection controls

Selection controls switch on/off the possibility to set selection in the context from VInspector plugin.

Action	Tree view item	Functionality
Select Presentations	AIS_InteractiveObject	Calls <i>AddOrRemoveSelected</i> of interactive object for the selected item
Select Owners	SelectMgr_EntityOwner or SelectMgr_SensitiveEntity	Calls <i>AddOrRemoveSelected</i> of SelectMgr_EntityOwner for the selected item

Please note, that the initial selection in context will be cleared. If the button is toggled, the button selection is active. Only one button may be toggled at the moment.

History view

At present the History view is under implementation and may be used only in a custom application where Inspector is loaded.

To fill this view, VInspectorAPI_CallBack should be redefined in the application and send signals about some actions applied to context. After, the call back should be given as parameter in the plugin. If done, new items will be created in the history view for each action.

2.3.3 Elements cooperation

VInspector marks current selected presentations in AIS_InteractiveContext with blue background in tree items. Use "Update" button to synchronize VInspector selected items state to the context.

It is also possible to perform selection in context using "Selection controls" VInspector. However, it should be performed carefully as it clears the current selection in AIS_InteractiveContext.

Selection change:

From	To	Action	Result
AIS_InteractiveContext	VInspector	perform selection in AIS_InteractiveContext	Click "Update" button in VInspector and check "Selection" column: AIS_InteractiveContext item has amount of selected objects, some of AIS_InteractiveContext Object have filled value if it selection happens for this presentation or entity owner of it
VInspector	AIS_InteractiveContext	activate one of Selection controls and select one or several elements in tree view	The objects become selected in AIS_InteractiveContext

2.4 ShapeView Plugin

2.4.1 Overview

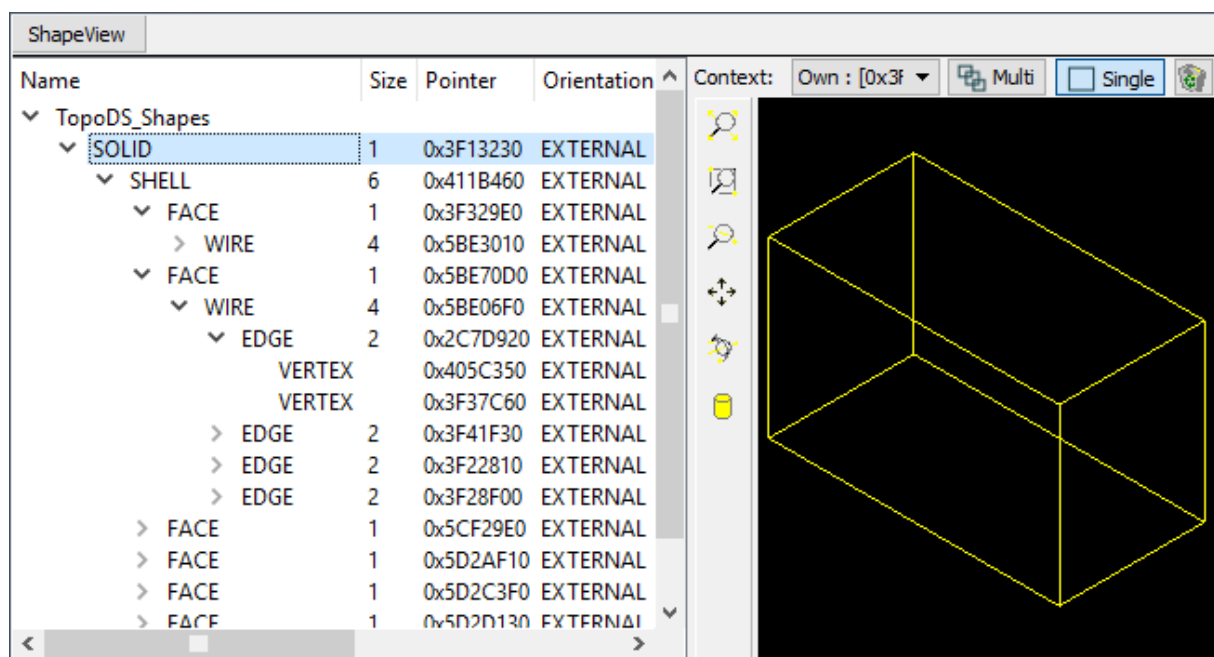


Figure 8: ShapeView

This plugin visualizes content of TopoDS_Shape in a tree view.

2.4.2 Elements

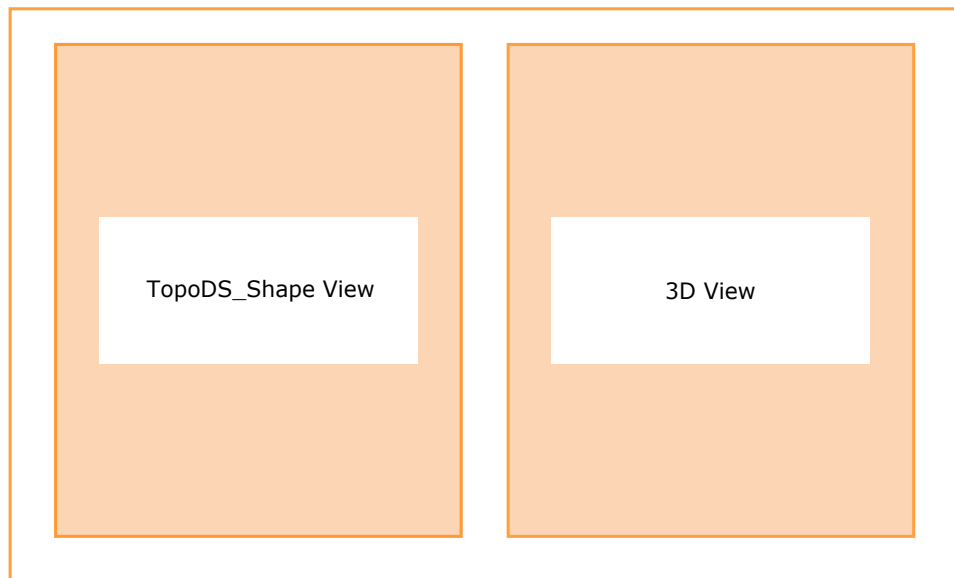


Figure 9: ShapeView Elements

TopoDS_Shape View

Elements of the view are TopoDS_Shape objects. This shape is exploded into sub-shapes using TopoDS_Iterator of the TopoDS_Shape. Child sub-shapes are presented in the view as children of the initial shape. Iterating recursively by all shapes we obtain a tree view of items shown in the ShapeView.

Columns of the View show some information about TopoDS_Shape of the item. The most informative column is the last column of TopoDS_Vertex and TopoDS_Edge shape types.

For TopoDS_Vertex it contains the point coordinates,

for TopoDS_Edge it contains the first and the last point coordinates, the edge length and some other parameters.

Context popup menu in tree view:

Action	Functionality
Load BREP file	Opens selected file and appends the result TopoDS_Shape into tree view
Remove all shape items	Clears tree view
BREP view	Shows text view with BREP content of the selected item. It creates BREP file in temporary directory of the plugin.
Close All BREP views	Closes all opened text views
BREP directory	Displays folder where temporary BREP files have been stored.

2.4.3 Elements cooperation

Selection of one or several items in TopoDS_Shape View creates AIS_Shape presentation for it and displays it in the 3D View.

3 Common controls

3.1 3D View

3.1.1 Overview

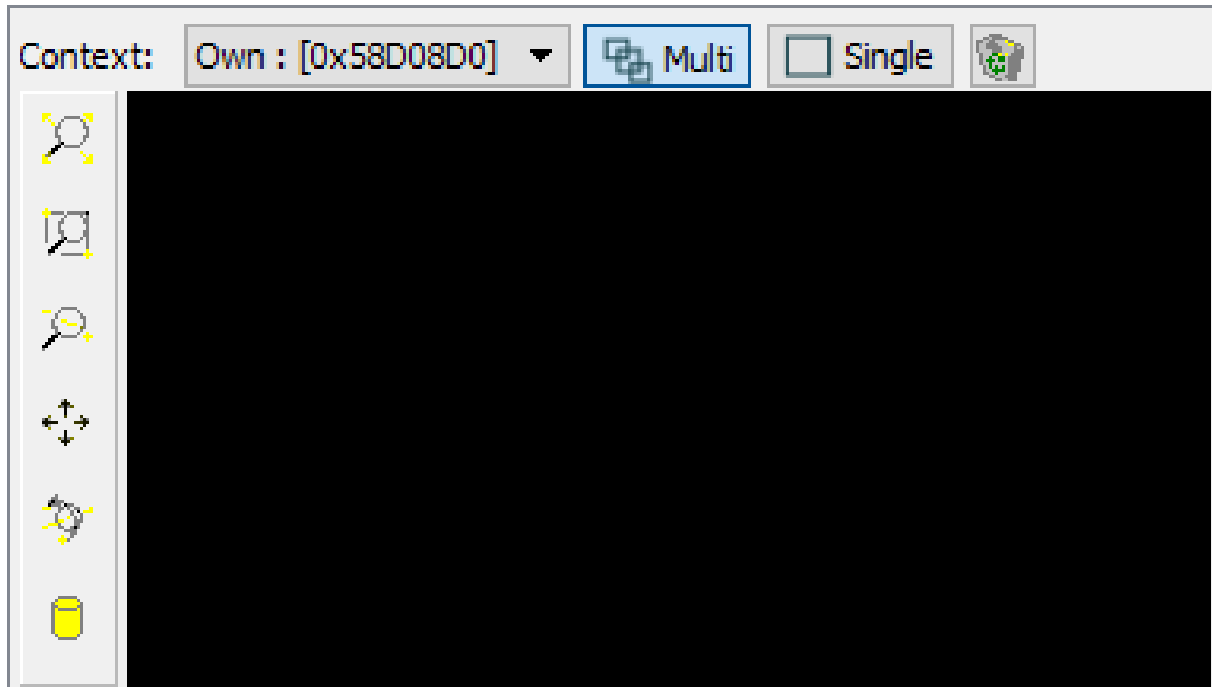


Figure 10: 3D View

Control for OCCT 3D viewer. It creates visualization view components with possibilities to perform some user actions for the view.

3.1.2 Elements

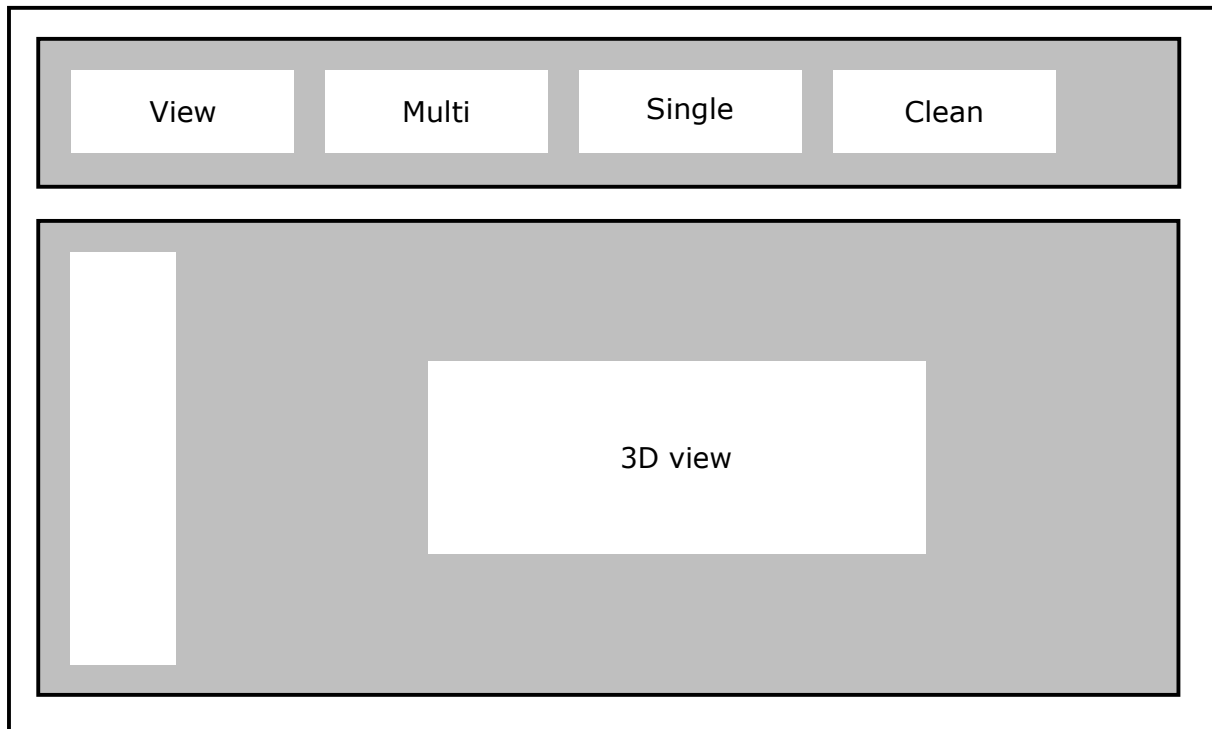


Figure 11: 3DView Elements

3D View contains:

Element	Functionality
3D view	V3d viewer with mouse events processing
Context	choice of another context that should be used in the plugin. It is possible to use the next contexts: Own - context of this view, External - context come in parameters which initializes plugin, None - do not perform visualization at all
Multi/Single	Buttons defined what to do with the previous displayed objects: Multi displays new presentations in additional to already displayed, Single removes all previous displayed presentations
Clean	Removes all displayed presentations
Fit All, Fit Area, Zoom, Pan, Rotation	Scene manipulation actions
Display Mode	Sets <i>AIS_Shading</i> or <i>AIS_WireFrame</i> display mode for all presentations

4 TInspectorEXE sample

Inspector functionality can be tried using this sample.

Use *inspector.bat* script file placed in binary directory of OCCT to launch it.

This script accepts the names of plugin's DLL that should be loaded. By default it loads all described above plugins.

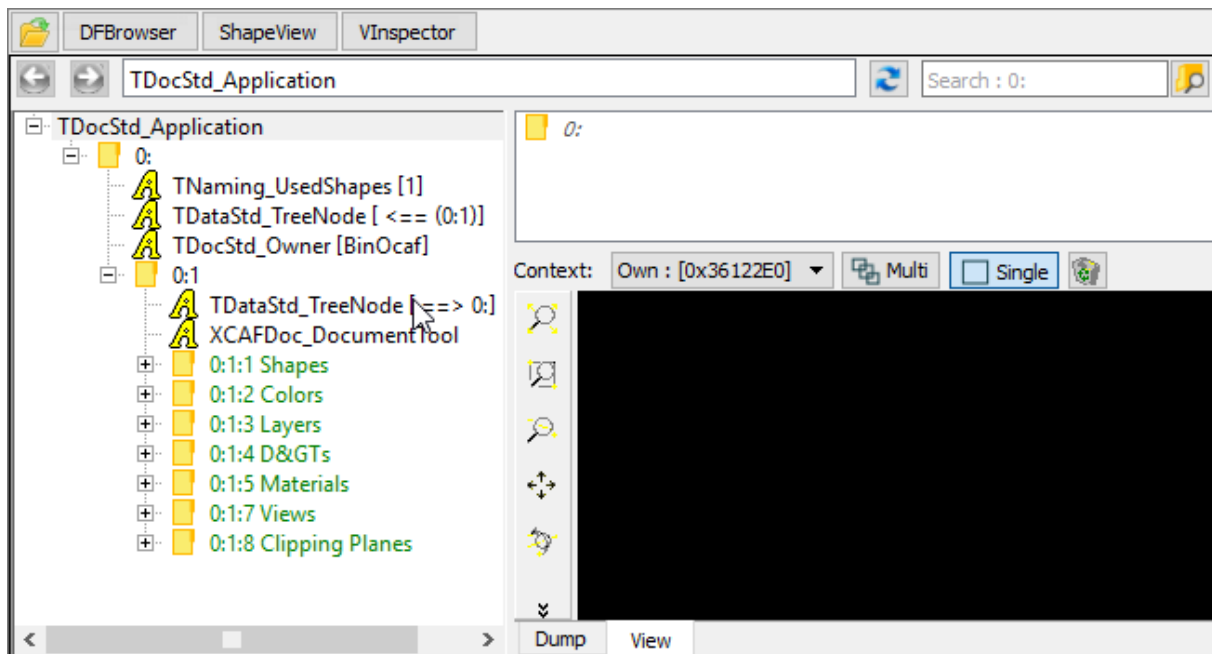
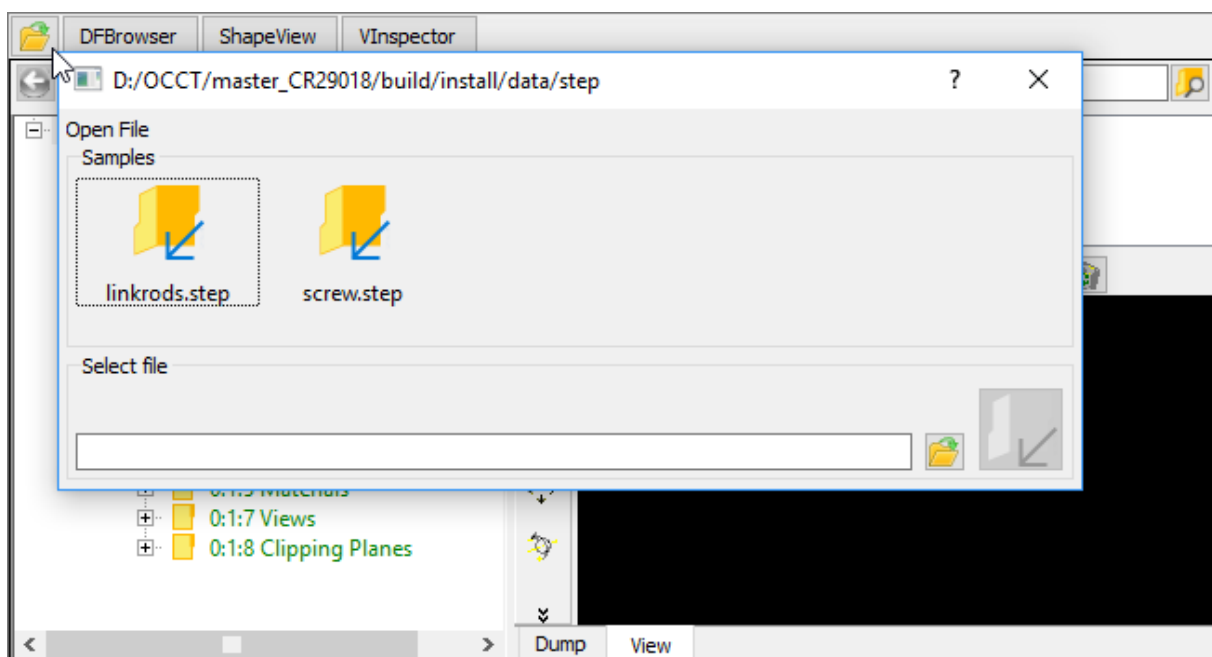


Figure 12: TStandaloneEXE

Click on the Open button shows the dialog to select a file. The user is able to select one of the sample files or load own one.



Depending on the active plugin, the following files should be selected in the dialog: OCAF document or STEP files

for DFBrowser and BREP files for VInspector and ShapeView plugins.

It is possible to click the file name in the proposed directory, enter it manually or using Browser button. The last Loading icon becomes enabled if file name is correct.

By default TInspectorEXE opens the next files for plugins:

Plugin DLL library name	Files
TKDFBrowser	step/screw.step
TKVInspector	occ/hammer.brep
TKShapeView	occ/face1.brep, occ/face2.brep

These files are found relatively *CSF_OCCTDataPath*.

Source code of TInspectorEXE is a good sample for [Using Inspector in a custom application](#).

5 Launch in DRAW Test Harness

TKToolsDraw plugin is created to provide DRAW commands for Qt tools. Use INSPECTOR parameter of pload command to download commands of this library. It contains tinspector command to start Inspector under DRAW. See more detailed description of the tinspector command.

The simple code to start Inspector with all plugins loaded:

```
pload INSPECTOR  
tinspector
```

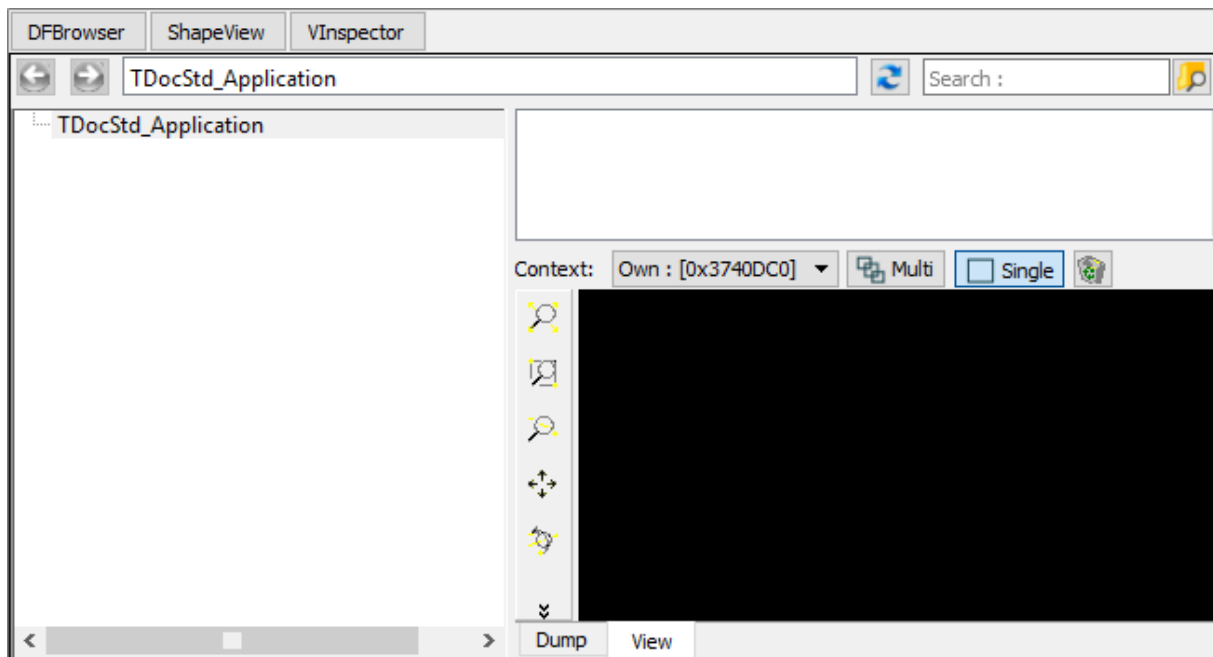


Figure 13: tinspector

Result of this command is the next:

- all available Plugins are presented in the Inspector. These are [DFBrowser](#), [VInspector](#) and [ShapeView](#).
- DFBrowser is an active plugin
- tree of OCAF is empty.

After, we should create objects in DRAW and update tinspector.

6 Using in a custom application

To use Inspector in an application, the next steps should be done:

- Set dependencies to OCCT and Qt in application (Header and Link)
- Create an instance of TInspector_Communicator.
- Register plugins of interest in the communicator by DLL library name
- Initialize communicator with objects that will be investigated
- Set visible true for communicator

C++ code is similar:

```
#include <inspector/TInspector_Communicator.hxx>

static TInspector_Communicator* MyTCommunicator;

void CreateInspector()
{
    NCollection_List<Handle(Standard_Transient)> aParameters;
    //... append parameters in the list

    if (!MyTCommunicator)
    {
        MyTCommunicator = new TInspector_Communicator();

        MyTCommunicator->RegisterPlugin ("TKDFBrowser");
        MyTCommunicator->RegisterPlugin ("TKVInspector");
        MyTCommunicator->RegisterPlugin ("TKShapeView");

        MyTCommunicator->Init (aParameters);
        MyTCommunicator->Activate ("TKDFBrowser");
    }
    MyTCommunicator->SetVisible (true);
}
```

Plugin	to be initialized by
TKDFBrowser	TDocStd_Application
TKVInspector	AIS_InteractiveContext
TKShapeView	TopoDS_TShape

7 Build procedure

By default the Inspector compilation is off. To compile it, set the **BUILD_Inspector** flag to "ON". See Configuration process.

When this option is switched On, MS Visual Studio project has an additional tree of folders:

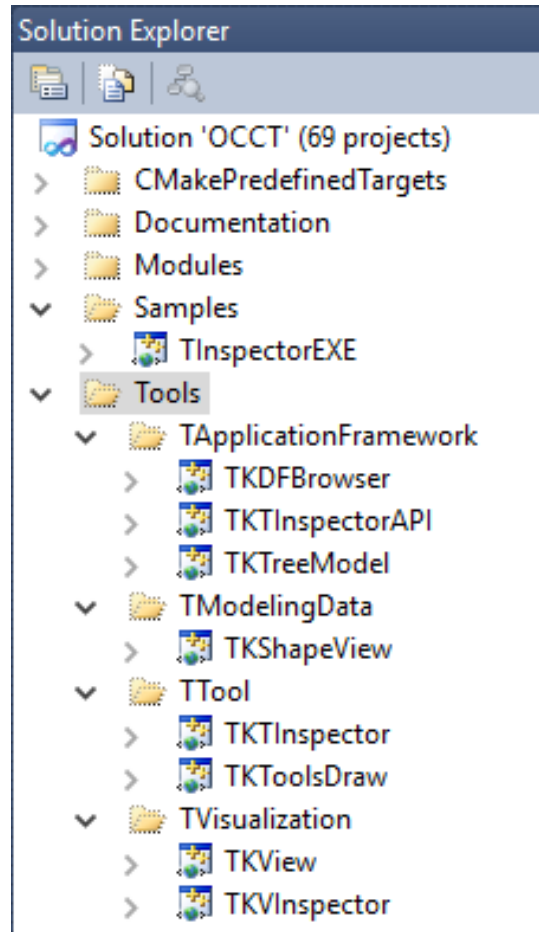


Figure 14: Inspector packages in MS Visual Studio

8 Sources and packaging

OCCT sources are extended by the /tools directory.

Distribution of packages participated in plugins:

Sources packages	Plugin
DFBrowser, DFBrowserPane, DFBrowserPaneXDE, TKDFBrowser	DFBrowser
VInspector, TKVInspector	VInspector
ShapeView, TKShapeView	ShapeView

Other packages:

Sources packages	Used in
TInspectorAPI, TKInspectorAPI	Interface for connection to plugin.
TreeModel, TKTreeView	Items-oriented model to simplify work with GUI tree control.
View, TKView	3D View component
TInspector, TKTInspector	Inspector window where plugins are placed
ToolsDraw, TKToolsDraw	Plugin for DRAW to start Inspector

In MSVC studio the separate folder contains Inspector projects.

9 Glossary

- **Component** – OCCT part, e.g. OCAF, VISUALIZATION, MODELING and others.
- **Plugin** – library that is loaded in some executable/library. Here, the plugins are:
 - DFBrowser,
 - ShapeView,
 - VInspector.

9.1 TDF_Attribute Simple types

Types where the content is a single value

Type	Kind of value
TDataStd_AsciiString	TDataStd_AsciiString
TDataStd_Comment	TCollection_ExtendedString
TDataStd_Integer	Standard_Integer
TDataStd_Name	TCollection_ExtendedString
TDataStd_Real	Standard_Real
TDF_Reference	TDF_Label
TDF_TagSource	Standard_Integer

9.2 TDF_Attribute List types

Type	Kind of value (container of)
TDataStd_BooleanList	Standard_Boolean
TDataStd_ExtStringList	TCollection_ExtendedString
TDataStd_IntegerList	Standard_Integer
TDataStd_RealList	Standard_Real
TDataStd_ReferenceList	TDF_Label

9.3 TDF_Attribute Array types

Type	Kind of value (container of)
TDataStd_BooleanArray	Standard_Boolean
TDataStd_ByteArray	Standard_Byte
TDataStd_ExtStringArray	TCollection_ExtendedString
TDataStd_IntegerArray	Standard_Integer
TDataStd_RealArray	Standard_Real
TDataStd_ReferenceArray	TDF_Label

9.4 XDE tree node ID description

GUID	Text
XCAFDoc::ShapeRefGUID()	Shape Instance Link
XCAFDoc::ColorRefGUID (XCAFDoc_ColorGen)	Generic Color Link
XCAFDoc::ColorRefGUID (XCAFDoc_ColorSurf)	Surface Color Link

GUID	Text
XCAFDoc::ColorRefGUID (XCAFDoc_ColorCurv)	Curve Color Link
XCAFDoc::DimTolRefGUID()	DGT Link
XCAFDoc::DatumRefGUID()	Datum Link
XCAFDoc::MaterialRefGUID()	Material Link